NOVA IMS Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

The impact of Microsoft Power Platform in streamlining end-to-end business solutions

Internship Report at Microsoft Portugal, Specialist Team Unit

Rodrigo Umbelino Barata Salgueiro

Internship Report presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

THE IMPACT OF MICROSOFT POWER PLATFORM IN STREAMLINING END-TO-END BUSINESS SOLUTIONS

by

Rodrigo Umbelino

Internship Report presented as partial requirement for obtaining the Master's degree in Data Science and Advanced Analytics

Co Advisor: Miguel de Castro Neto

Co Advisor*:* Manuel Dias

November 2020

DEDICATÓRIA

Para a minha mãe que tornou tudo isto possível.

AGRADECIMENTOS

Em primeiro lugar um agradecimento à minha mãe, avós e tio, pelo apoio incondicional que me deram ao longo destes anos, que foi absolutamente instrumental e necessário para o meu sucesso. Foi apenas e só devido ao seu esforço e sacrifício combinado que eu sou a pessoa que sou hoje e por isso estarei eternamente grato. Muito obrigado.

Aos professores Miguel de Castro Neto e Manuel Dias, por supervisionarem este trabalho e por todos os ensinamentos que me passaram. O vosso acompanhamento e conhecimento foram essenciais para este trabalho.

Para a minha equipa da Microsoft Portugal: obrigado por me receberem de braços abertos e por me terem ensinado tanto num espaço tão curto de tempo. É um privilégio poder trabalhar diariamente com um conjunto de pessoas tão talentosas.

Aos meus bons amigos e parceiros nesta jornada, Rita Franco e Vitor Manita, este mestrado não teria sido possível sem vocês, tornaram todo o percurso bem mais divertido. Resta agora celebrar (sem ser na sala 24) os frutos de todas as diretas e doses de cafeína que esta fase exigiu. Obrigado. Um agradecimento também ao Gaspar e ao Fartura, que tanto me ensinaram durante as inúmeras vezes que trabalhámos juntos durante a licenciatura.

A todos os meus restantes amigos, portugueses ou de além-mar, dos loucos da Loba sempre presentes até aos da Margem Sul, passando pela folia Torrense, pelos bairros de Alvalade até aos nascidos em Campolide: um bem-hajam. Um agradecimento pelas gargalhadas dadas até aqui e pelas que ainda estarão para vir.

Um agradecimento a três professores: à professora Fernanda Bento por me ensinar a perguntar e a aprender, ao professor Ivo Gonçalves por me ensinar a persistência académica com sentido de humor, à professora Carla Umbelino por me ensinar a sonhar e a viver.

ABSTRACT

Nowadays, there is a greater urgency for companies to innovate and digitally transform. Automation and digitalization of processes, coming up with new ways to connect with employees and customers, or investing in a robust infrastructure that can respond and extract insights from the ever-increasing amount of data that is being produced across all sectors of the enterprise: these are all topics that any company needs to have as a top priority if they want to keep up with their competitors and stay relevant in their own markets.

The present document describes the activities carried out during a period of 10 months (September 2019 – June 2020, correspondent to the fiscal year of 2020) at Microsoft's Corporation Portuguese Subsidiary, working as a full-time Technical Specialist. The technical specialist first received extensive technical and commercial training from a vast variety of resources (online resources, one-to-one shadowing, and on-site technical readiness) and after the ramp-up process, he proceeded to drive the business for Microsoft Portugal by conducting customer and partner meetings, and to play his part in the sales motion by providing deep technical expertise and compelling technology demonstrations.

The technical specialist oversaw a unified set of Microsoft technologies, called the Power Platform, that bundle together a set of tools such as Power BI for data analysis and visualization, Power Apps for line-of-business application development, Power Automate for workflow automation and RPA capabilities and Power Virtual Agents as an engine for creating and deploying intelligent chatbots. All these components are further enriched and integrated with the Office 365 and Dynamics 365 ecosystems, hundreds of data connectors, advanced database capabilities, artificial intelligence and machine learning models, and the extensibility from the resources available on the Azure Cloud.

KEYWORDS

Microsoft; Power Platform; Technical Specialist; Data Analysis; App Development; Process Automation; Artificial Intelligence; Chatbots; Cloud Computing; Machine Learning;

RESUMO

Atualmente, existe uma maior necessidade das empresas inovarem e reinventarem-se na sua jornada para a transformação digital. A automação e digitalização de processos, arranjar novas maneiras de conectar com trabalhadores e clientes, ou investir numa infraestrutura robusta que consiga dar resposta à necessidade de extração de informação de um volume de dados em aumento constante: isto são factores que qualquer empresa precisa de ter como prioridade se quiser manter-se a par com a sua competição e permanecer relevante nos seus próprios mercados.

O presente documento descreve as atividades levadas a cabo durante um período de 10 meses (setembro 2019 – junho de 2020, período correspondente ao ano fiscal de 2020) na subsidiária portuguesa da Microsoft Corporation, desempenhando o cargo de Especialista Técnico em regime de full-time. O especialista técnico primeiramente recebeu uma extensa formação técnica e comercial (recursos online, shadowing um-para-um e formação técnica presencial) e, após o processo de ramp-up, prosseguiu as suas funções de dinamizar o negócio para a Microsoft Portugal, através da concretização de reuniões com clientes e parceiros e desempenhando o seu papel no ciclo de venda, sendo a principal fonte de profundo conhecimento técnico e entregando demonstração tecnológicas apelativas.

O especialista técnico estava encarregue de um conjunto unificado de tecnologias Microsoft, chamado de Power Platform, que agrega um conjunto de ferramentas como o Power BI para análise e visualização de dados, Power Apps para desenvolvimento ágil aplicacional, Power Automate para automatização de processos e capacidades de automatização robótica e Power Virtual Agents como o motor para a criação de agentes virtuais inteligentes. Todos estes componentes são enriquecidos pela integração nativa com os universos Office e Dynamics 365, centenas de conectores para fontes de dados, bases de dados avançadas, inteligência artificial e modelos de aprendizagem automática, e a extensibilidade dos recursos existentes na Azure Cloud.

PALAVRAS-CHAVE

Microsoft; Power Platform; Especialista Técnico; Análise de dados; Desenvolvimento aplicacional; Automação de processos; Inteligência artificial; Chatbots; Computação cloud; Aprendizagem automática;

INDEX

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

AD	Active Directory
AI	Artificial Intelligence
ALM	Application Lifecycle Management
API	Application Programming Interface
BA	Business Applications
BI	Business Intelligence
BPMS	Business Process Management System
CDS	Common Data Service
CEO	Chief Executive Officer
CI/CD	Continuous Integration and Continuous Delivery
CRM	Customer Relationship Management
CSU	Customer Success Unit
CSV	Comma Separated Values
DAX	Data Analysis eXpressions
ERP	Enterprise Resource Planning
ETL	Extract, Transform, Load
FY	Fiscal Year
GAAP	Generally Accepted Accounting Principles
GUI	Graphical User Interface
HR	Human Resources
HTML	Hypertext Markup Language
IaaS	Infrastructure as a Service
IDE	Integrated Development Environments
iOS	Internet Operating System
IoT	Internet of Things
ISP	Internet Service Provider

IS	Information System
IT	Information Technology
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LoB	Line of Business
LT	Leadership Team
LUIS	Language Understanding Intelligent Service
ML	Machine Learning
M&O	Marketing & Operations
MW	Modern Workplace
NLP	Natural Language Processing
OCP	One Commercial Partner
OCR	Optical Character Recognition
OEM	Original Equipment Manufacturer
PaaS	Platform as a Service
PoC	Proof of Concept
POS	Point of Sale
Q	Quarter
Q&A	Question and Answer
RPA	Robotic Process Automation
SaaS	Software as a Service
SQL	Structured Query Language
STU	Specialist Team Unit
TS	Technical Specialist
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience

1. INTRODUCTION

The present document describes the work that was done during the fiscal year of 2020 at Microsoft Portugal, which is also a result of the partnership between Nova Information Management School and said entity. This work report is part of the requirements of completion of the Master Program in Data Science and Advanced Analytics and aims to smooth the student's transition between the academia world and the job market.

The full-time role of Technical Specialist is part of the Specialist Team Unit (STU) at Microsoft Portugal, which is responsible for all the pre-sales tasks on a more technical level (product/technological demonstrations, technical capabilities, product evangelizing, and so on). The STU is a crucial piece on the overall sales motions and is deeply interconnected with other teams in the organization. The huge plethora of products that are made available by Microsoft, as well as the vast number of clients with different characteristics that the company has, demands for an according segmentation across the organization.

Since the Technical Specialist role is by far the most technical one in the entire Sales team, there is a big effort in upskilling this team, not only from a technical perspective, but also from a commercial and industry point of view. These learning materials ranged from online upskilling produced and distributed by Microsoft, an in-person two-week event that happened in Munich, Germany, with the purpose of delivering deep technical and commercial training and mentoring and shadowing sessions with peers and manager. These set of initiatives allowed for an effective onboarding to the full-time role, and a gradual learning of a very wide set of technologies that needed to be apprehended.

Whilst this readiness phase was occurring, the technical specialist was also actively involved with several projects for internal use at Microsoft Portugal, that served as practical use-cases for the appliance of the technical knowledge that he was acquiring during that period (such use-cases will be described in further detail later in this report).

After the technical readiness was achieved, the Technical Specialist was fully integrated within the sales motions, maintaining deep technical meetings with customers and partners, custom building and developing tailored technology demonstrations and PoCs for strategic accounts, delivering technical workshops to engaged customers and representing Microsoft Portugal as a speaker in both internal and external events.

2. COMPANY AND ROLE DESCRIPTION – MICROSOFT CORPORATION

In this chapter, a more detailed description is made regarding the company in which this work report is inherent to – Microsoft Corporation. It will cover a brief history of the company, its mission, values and vision, and a more detailed view of the overall enterprise structure, to ensure that the scope of the different teams and roles are clear.

2.1. COMPANY PRESENTATION AND LOCATION

Microsoft Corporation is a publicly owned multinational headquartered in Redmond, Washington. It was founded in 1975 by Bill Gates and Paul Allen, in Albuquerque, New Mexico. Satya Nadella is the company's current CEO, replacing Steve Ballmer in 2014, leaving his previous role in the company as the Executive Vice-President of the Cloud and Enterprise Group. The company's core business model revolves around the entire lifecycle of manufacturing, developing, selling, and supporting a wide array of software and hardware solutions, at the enterprise and personal level.

Since its very beginning, Microsoft has produced some of the world's most well-known technological products, that include the various Windows operating systems, Azure Cloud services, the Modern Workplace Office productivity applications, and the Business Applications segment with the full suite of Dynamics CRM and ERP. In addition to these, the company has solidified its range of offerings with services like LinkedIn (the business-oriented social networking service), GitHub (a collaborative platform for developers) and the popular gaming system Xbox.

As of today, Microsoft operates at a worldwide scale, reaching more than 200 countries. The Portuguese subsidiary was created in 1990, driving the business on a national level ever since. It currently maintains two offices situated in Lisbon, one in Parque das Nações and another one in Amoreiras. Since the company has such a vast portfolio of technological solutions, as well as a worldwide geographical reach, it serves and operates in virtually every industry (Figure 1).

Figure 1 - Sample of industries in which Microsoft operates, source: https://www.microsoft.com/pt-pt/industry (2020)

Microsoft caters to a diverse pool of customer segments, including small to medium organizations, large global enterprises, private consumers, ISPs, governments and public sector institutions, OEMs, etc. As of July 2020, the company possessed an employment headcount of more than 150.000 employees, a trillion-dollar market capitalization, and presented a revenue of $38.0 billion (a 13% increase compared to the last fiscal year) and a net income of $11.2 billion (a 15% GAAP decrease compared to the last fiscal year) (Microsoft Corporation, 2020). These results surpassed Wall Street's expectations, even when the impact of the global COVID-19 pandemic is considered.

2.2. MISSION AND VALUES

2.2.1. Mission

Microsoft's mission is to leverage technology as the disruptive engine for positive change: "*Our mission is to empower every person and every organization on the planet to achieve more.*"

2.2.2. Values

Microsoft's values are divided into two categories: Company Values and Corporate Values.

Corporate Values:

- Respect
- Integrity
- Accountability

Company Values:

- Innovation
- Diversity and Inclusion
- Corporate Social Responsibility
- AI
- Trustworthy Computing

2.3. ENTERPRISE STRUCTURE

For a correct understanding of the context of the role and team itself, it is important to give a brief explanation of how Microsoft is structured. Since it would be impossible to describe the full organizational chart of a company as large as this one, it will be broken down into two pieces: firstly, a more high-level approach of the different areas that exist at a senior leadership level

(Microsoft worldwide). Secondly, a more granular explanation will be made that accurately reflects the structure of the Portuguese subsidiary for Microsoft.

As of June 2020, Microsoft Business Organization is divided into 2 main groups: the Engineering Groups (responsible for the engineering side of the different technological areas at Microsoft) and the Business Functions (the core areas that maintain and develop the business at the enterprise level).

Engineering Groups:

- Cloud + AI Group
- Experiences + Devices
- Technology + Research
- Core Services Engineering & Operations

Business Functions:

- Business Development Group
- Corporate, External and Legal Affairs
- Corporate Strategy
- Finance Group
- Global Sales, Marketing and Operations
- HR Group
- LinkedIn
- Marketing Group
- Worldwide Commercial Business

At the national level, the Portuguese subsidiary obeys the following structure:

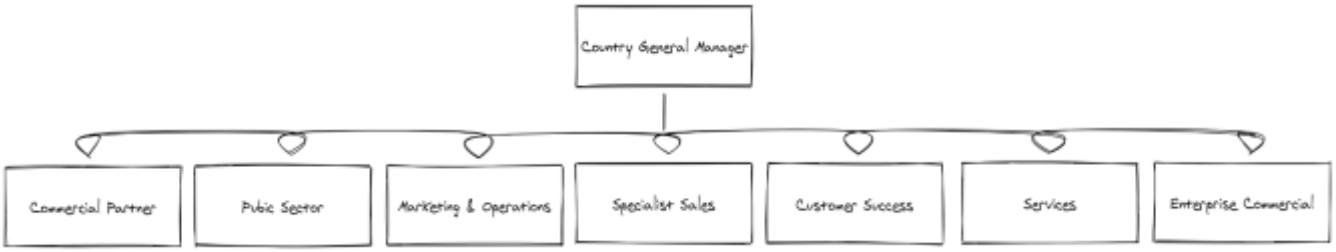

Figure 2 - Microsoft Portugal leadership team organizational chart

The Portuguese Leadership Team (LT) is leaded by the Country General Manager, and there is an assigned Lead for every business area. The One Commercial Partner (OCP) team manages the end-to-end execution of Microsoft's commercial partner programs, focusing on commercial partner ecosystem strategy, partner solution strategy, commercial partner investment governance, the global services partner, OCP field engagement and value delivery to Microsoft's unmanaged partners. The Public Sector team is responsible, as the name implies, for the business development amongst public sector clients, finding new revenue opportunities and empowering public sector customers to drive more value from technology. As for the M&O team, their main focus is to drive cross-company insight and execution excellence, bringing strategy and priorities to life, accelerating the pace of transformation and enabling Microsoft to deliver business impact at scale. Specialist Sales is the team where the Technical Specialist role is part of (among others), and its main goal is to drive Microsoft's new revenue growth and competitive market share across the business lifecycle, making use of deep technical and industry knowledge, as well as commercial and sales skills. The Customer Success Unit (CSU) is the team responsible for driving customer's technology adoption, and to guide them to get real business value from their IT investments, increase their satisfaction and help them achieve their short and long-term technology goals. The Services team applies their extensive technical knowledge to solve customer's business problems, by understanding their targets and goals and guiding them in their digital transformation journey. Last, but not least, the Enterprise Commercial team amplifies the seller's capabilities to have differentiated customer conversations, while shaping the company's agenda to develop industry specific solutions.

2.4. TEAM AND ROLE DESCRIPTION

Having a clearer perspective of how the company is structured at a national and international level, it is now possible to deep dive on the team in which the technical specialist operates (STU), as well as a more complete take on the role itself. In the Portuguese subsidiary, the STU is leaded by the Specialist Sales Lead and it is divided by different segments and solution areas: the Intelligent Cloud segment, which encapsulates all of the Azure cloud workload, and is driven by Azure specialists across two solution areas (Apps and Infrastructure and Data and Analytics); the MW & BA segment, which is driven by Modern Workplace and Business Applications specialists and focus on the whole Office 365, Security, Voice and Comms, the Surface line and the ERP/CRM businesses; last but not least, there is the Technical Sales segment which consists of technical specialists across all of the already mentioned solution areas.

Focusing on the role of the Power Platform Technical Specialist, it sits in the middle of the Modern Workplace and Business Application Solution Areas. The TS is engaged by the sellers to own and win customers' technical decisions regarding Microsoft technology and positions solution requirements to the customer for deployment and usage. The role seeks to proactively own technical decisions and spends a vast majority of its working hours in maintaining a deep theoretical and experiential technical knowledge, as well as understanding core area compete offerings technical capabilities, always keeping the customer problem in mind. Regarding the Power Platform stack, the TS must maintain expert level of knowledge in Power BI Desktop/Service/Mobile/Report Server/Gateway, CDS, Visio data visualizations, Power Apps,

Power Automate and Power Virtual Agents. In addition to this, the TS also must have profound mastery of tools for use case ideation across the Power Platform (workshops, hackathons, etc) as well as depth in solution development across Office 365 (Teams, SharePoint Online, Forms Pro, Microsoft Graph, etc), Dynamics 365 and Azure cloud.

3. THEORETICAL FRAMEWORK

In this section it will be presented the theoretical framework of the concepts and tasks that were addressed by the technical specialist during the fiscal year of 2019. This chapter focuses more on the current technological landscape and different data concepts, whilst in the next chapter the actual tools that were used are explained in depth from a technical perspective.

3.1. LOW-CODE: PLATFORMS, MOMENTUM AND OPPORTUNITIES

A low-code platform can be defined as "Platforms that enable rapid delivery of business applications with a minimum of hand-coding and minimal upfront investment in setup, training and development" (C. Richardson & Rymer, 2016). Low code brings a new perspective into the development world, not only by reducing costs and accelerating deployment, but also because it broadens the horizons to a new target audience of citizen developers. A citizen developer is usually a business-oriented power user, that has enough technology chops on mainstream technical tools to work together with IT teams and pro-development professionals to develop and deploy applications, integrations, automations, business intelligence/analytics or AI models.

The rise of low-code adoption is largely justified by the increasing demand for software by digital businesses, with 23% from a sample of global developers stating that there is active usage of low-code platforms in 2018 in their companies, and 22% admitting that they are planning on shifting into said technologies by 2019 (Rymer & Koplowitz, 2019). Moreover, analysts predict that the demand for mobile applications will grow at least five times faster than IT capacity to deliver (Moore, 2015) and that in the next five years 500 million apps will be created, equal to the number built over the past 40 years (Gens et al., 2019). Turning the prospects into the foreseeable future, the IT analyst giant Gartner predicts that by 2024, low-code development will be responsible for more than 65% of development activity (Vincent et al., 2019).

Figure 3 - The Forrester Wave™: Low-Code Development Platforms For AD&D Professionals, Q1 2019 (Rymer & Koplowitz, 2019)

Figure 4 - Gartner 2019 Magic Quadrant for Enterprise Low-Code Application Platforms (Vincent et al., 2019)

With this amount of traction in the low-code market, it is only natural that software vendors increasingly funnel investment into their own offerings in the low-code space. Analysts prospect four major leaders of the low-code universe in today's market: Microsoft, Salesforce, OutSystems and Mendix (Figure 3 and Figure 4). Each vendor has its own take on a low code solution, but the trend that we are seeing is that they are investing on said platforms and complementing the existing offers with more advanced and out-of-the-box features.

When it comes to the business intelligence space, more and more often chief data officers and analytics leads are seeking to deploy modern BI platforms that can leverage mobile and embedding capabilities, so that they can expand the impact of BI and advanced analytics in their organizations, countering the current 30% of all employees that are currently adopting BI at the enterprise level (Sallam & Howson, 2017).

Figure 6 - Gartner 2020 Magic Quadrant for Analytics and Business Intelligence Platforms (J. Richardson et al., 2020)

Figure 5 - The Forrester Wave™: Enterprise BI Platforms (Vendor-Managed), Q3 2019 (Evelson et al., 2019)

Because of the huge market opportunity in the analytics realm, the competition is fierce between the biggest software vendors, although Microsoft is positioned as a clear leader by the major tech analysts Forrester and Gartner (Figure 5 and Figure 6). As BI platforms are getting progressively more advanced and matured, the comparison between different tools is made based on differentiators like augmented BI and platform extensibility (Evelson et al., 2019). While modern analytics platforms evolved to tools that supported full analytic workflows (beginning in self-service data preparation, to visual rendering and consequently generating insights), long are the days when BI software was differentiated by their data visualization capabilities, in fact, these features are now the standard for business intelligence tools. Nowadays, enterprises are increasingly looking for platforms that have integrated support for enterprise reporting capabilities, as well as augmented analytics that use ML and AI features to assist business users in the whole lifecycle of data analysis - from data preparation to insight generation (J. Richardson et al., 2020).

Microsoft's offering, as far as low code is concerned, is the Power Platform. This technology was especially disruptive in the low-code realm because it addressed not only low-code application

development (Power Apps), but also low-code automation (Power Automate), business intelligence and data analytics (Power BI), chatbot creation (Power Virtual Agents) and AI and ML models, as well as native integration with the remaining Microsoft ecosystems.

3.2. Microsoft clouds: Office 365, Dynamics 365, and Azure

In order to understand the depth of a no-cliff development technology like the Power Platform, it is important to understand the different clouds that exist in the Microsoft universe that are the foundations for extensibility and integration of the platform. Microsoft has three main clouds amongst its product suite: Office 365 (the cross-productivity hub), Dynamics 365 (CRM/ERP product line) and Azure (the cloud computing platform).

3.2.1. Microsoft Office 365

Microsoft Office 365 is a secure subscription-based online workspace that includes several software applications designed for improving the collaborations between team members in any organization (Michalak, 2015). There are several subscription plans, all at different price ranges, but they all include (at least to some extent), the Microsoft Office 365 core apps and services: Word, Excel, PowerPoint, Outlook (business-grade email, calendar, and contacts), OneNote (file sharing and notes management), OneDrive (file storage) and Microsoft Teams (online videocalls and conferencing). Besides these core apps, there is a plethora of other functionalities associated with the Office 365 cloud (SharePoint, Yammer, Microsoft Forms, etc) that aim to elevate the productivity and collaboration experience in the work environment. Because the service is subscription based, all Office 365 licenses in business environments are updated automatically when there is a new version, at no extra cost for the end consumer.

Figure 7 - Office 365 core apps and services, source: https://www.microsoft.com/en-us/microsoft-365/products-apps-services (2020)

3.2.2. Microsoft Dynamics 365

Microsoft Dynamics 365 is a suite of business applications that allow you to combine different tools to create an infrastructure, a framework, and a core system within an organization, allowing companies to meet the demands of an ever-changing industry as well as customer expectations (Crichtley, 2018). Similarly to the other Microsoft clouds, Dynamics 365 allows for native integration with other Microsoft systems, making it possible to enhance applications by providing connectivity to Office 365 and synching the CRM/ERP solutions with Outlook, SharePoint, Teams, LinkedIn, etc. These business applications provide a holistic 360-degree view of customers and operations, by leveraging the existent interactions that are driven by Dynamics' intelligent services, companies can provide a very high-quality experience to the end-user. It aims to remove the complexity of traditional ERP/CRM systems by creating and moving modular business applications for customer relationship management and enterprise resource planning into a single, unified platform (Figure 8).

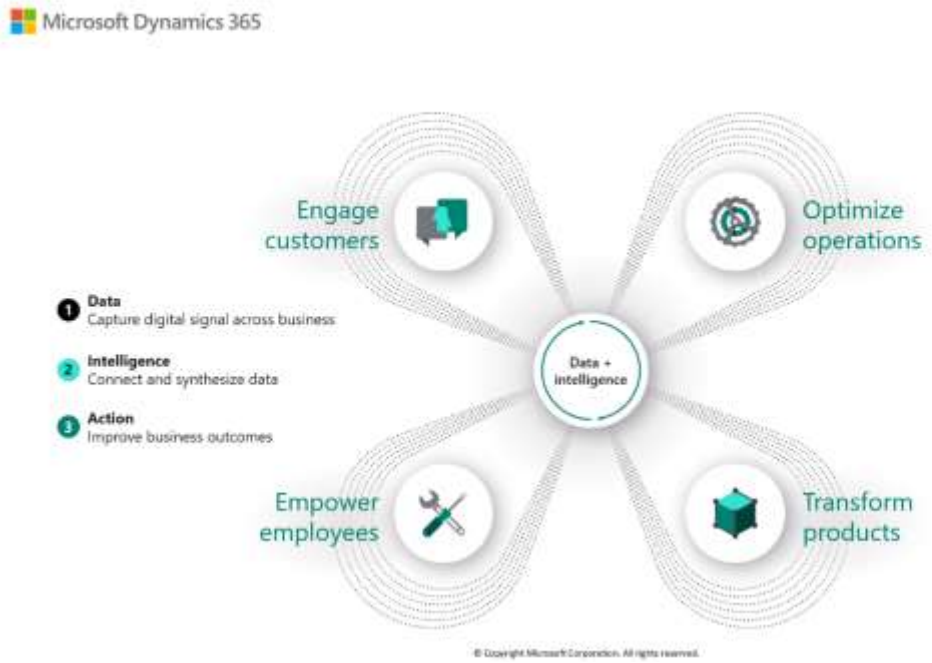

Figure 8 - Microsoft Dynamics 365 digital feedback loop, source: https://docs.microsoft.com/en-us/learn (2020)

Because of its modular structure, Dynamics 365 applications are divided into different sections depending on the systems that they derive from and use-cases that they address.

CRM/Model-driven applications:

- Dynamics 365 Sales is used for managing the relationship with customers at the enterprise level, to make data-driven decisions and to help closing the sales cycle faster. This module is used for the maintenance of a company's sales, leads and orders, creating marketing lists and campaigns and to manage accounts and contacts.
- Dynamics 365 Customer Service is all about maintaining the best possible customer relationship by providing with extensive features that help the company to manage the services that it provides to their customers.

- Dynamics 365 Field Service uses a mixture of scheduling algorithms, automation technology, to deliver onsite support to customer locations. Mobility is a crucial aspect of this module since it empowers mobile workers to successfully fix issues while being onsite with their customers.
- Dynamics 365 Marketing is a module focused on marketing automation that works integrated with the Sales module, and already has out-of-the-box embedded visualizations for BI. Amongst its features, it is possible to share information across sales and marketing teams, create customized graphical email messages, and more, all with the goal of turning prospects into business relationships.

ERP/Finance and Operations applications:

- Dynamics 365 Commerce allows for personalized customer engagements in order to increase brand loyalty, reduce costs by optimizing operations and supply chain efficiencies, and increase revenue by improving the productivity of the company's employees. At the core of this module, there is an omnichannel solution that aims to unify the different POSs of the company (back-office, in-store, call center, etc).
- Dynamics 365 Finance helps companies to automate and improve their overall financial operations, by providing features like real time performance monitoring, outcome prediction, and data-driven decisions. The Finance module reduces operational expenses, decreases financial complexity and risk, and provides the necessary data for making financial decisions with the help of AI.
- Dynamics 365 Human Resources aims to automate various staffing and recruitment processes, including the administration of benefits, performance reviews, training, employee retention, and so on. It is the module that provides the HR staff with management framework.
- Dynamics 365 Supply Chain Management strives to transform and improve the company's supply chain operations by using AI and IoT technology across the several stages of the supply chain. The goal of this module is to modernize warehouse management, maximize operational efficiency and the life of the company's assets.

Business Central application:

- Dynamics 365 Business Central is the one-stop business management solution for small to mid-sized companies. It is a module that allows the management of the company's processes including finance, manufacturing, sales, services, and more, all in one unified solution.

In addition to these predefined modules, all the Dynamics 365 realm can be further extended using the components of the Power Platform, for extra customization, or integrated with ad-hoc services from the Azure cloud.

3.2.3. Microsoft Azure

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage,

applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." (Mell & Grance, 2011)

One of the main advantages of betting on cloud computing, is the granularity with which companies can choose to adopt the technology. It is extremely cost-effective with most cloud providers (e.g., Microsoft, Amazon Web Services, Google, etc) providing a pay-as-you-go or consumption-based pricing model. In addition to this, the level of security, reliability, flexibility, elasticity, and scalability are characteristics that are perceived by business owners as the main advantages of cloud computing, allowing them to spend more time on what they need to spend, and let the underlying infrastructure details be self-managed.

Microsoft Azure is Microsoft's cloud computing platform that provides an endless variety of services without the need for purchasing or provisioning your own hardware. Azure enables the rapid development of solutions and provides the resources needed to accomplish tasks that are not suitable for on-prem environments (Collier & Shahan, 2015). The goal of using Azure as a cloud computing platform is to improve the efficiency of running a business, no matter the size of said business.

There are three deployment methods for the Azure cloud: public, private and hybrid. The public cloud is the most common deployment model, the customer does not have to manage any local hardware, because everything runs on the cloud provider's hardware. The private cloud is when a customer creates a cloud environment in their own datacenter and allow the users in the organization to access its computing resources (the customer remains responsible for the maintenance and costs of the hardware/software that they decide to provide). Lastly, the hybrid cloud allows a mixture of both previously mentioned clouds, giving customers the flexibility of running their applications in the most appropriate location depending on the use-case (use-cases in which it is impossible to put certain data in the public cloud for legal reasons, for example).

Besides the different deployment methods, the Azure cloud can also be classified in three categories: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The category that the customer chooses to pursue largely depends on the use-case that needs to be addressed, as well as the existent development practices in the company.

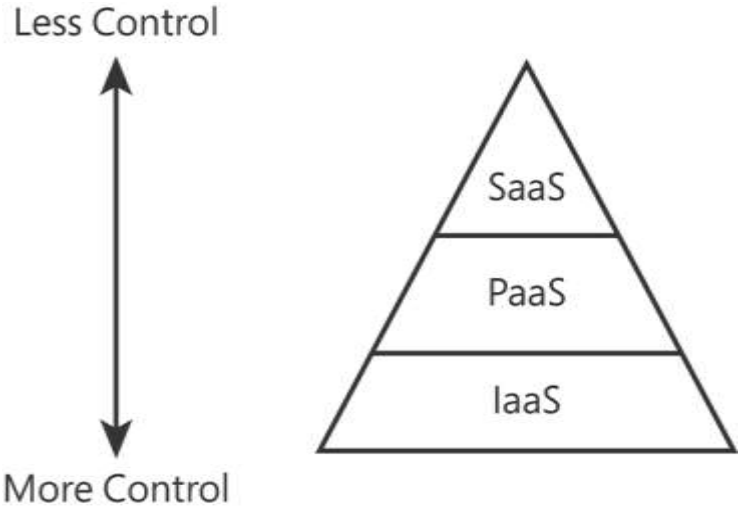

Figure 9 - Cloud categories pyramid (Cheshire, 2020)

Infrastructure refers to the hardware that an application uses, and IaaS refers to the virtualized infrastructure offered by the cloud provider. It is the deployment method with the most responsibility allocated to the customer, IaaS means that you control the operating system install and installation of other services, but also means that the customer is responsible for patches with security updates, troubleshooting, etc. The cloud provider is only responsible for providing the virtual machine, and the customer benefits from the underlying infrastructure in the area of fault tolerance and disaster recovery (Cheshire, 2020).

In a PaaS environment, besides the infrastructure, the cloud provider also provides the operating system, preinstalled software for connection to database and network systems (middleware) and several other features for building and managing complex cloud applications. PaaS services offer the flexibility of controlling the application but offload management and control of the underlying systems to the cloud provider, making PaaS located right in the middle of the cloud pyramid (Figure 9) (Cheshire, 2020).

Lastly, at the top of the cloud pyramid (Figure 9) there is the SaaS offering, where the cloud provider controls everything. It is basically software provided by a cloud provider that is installed on infrastructure completely controlled by the hosting provider, and the customer benefits from using software written and maintained by a third-party, as well as benefitting from allowing the cloud provider to maintain and configuring the application (Cheshire, 2020).

Because these different categories follow a layered structure, the levels of abstraction which are present in each category fluctuate quite a bit in terms of the depth of control that the end-customer will have over the resources. For each category, the level of control is less while the level of abstraction is higher, with different resources being managed by the customer or the service provider at each stage (Figure 10).

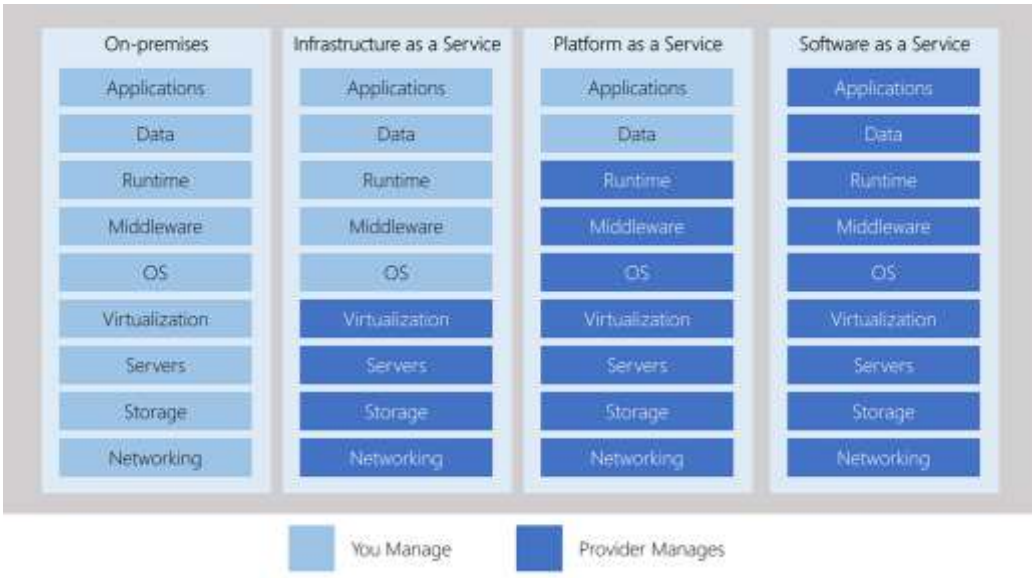

Figure 10 - Management responsibilities across cloud categories, source: https://docs.microsoft.com/en-us/learn (2020)

Azure brings a massive global infrastructure and countless services and features. The most used categories include: Compute, Networking, Storage, Mobile, Databases, Web, Internet of Things,

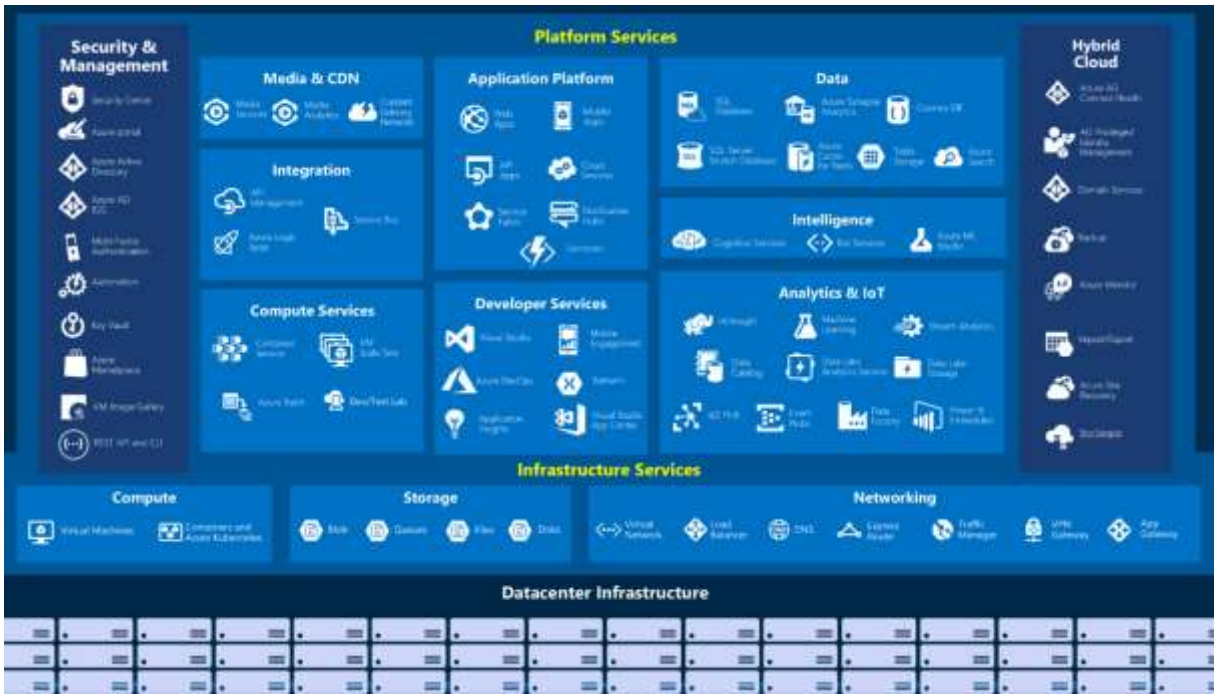

Figure 11 - High-level view of the services and features that the Azure cloud provides, source: https://docs.microsoft.com/en-us/learn (2020)

Big Data, Artificial Intelligence and DevOps. These are just a sample of the categories that are available on the Azure cloud, with a vast array of services inside each category. In the figure above (Figure 11) it is possible to get a high-level view of the variety and number of features that Azure provides.

3.3. DATA ANALYSIS AND VISUALIZATION

Data analysis is one of the most sought-after technical skills of the modern world. This is due to a variety of different reasons, but the technological advances allied with an increasing growth on the amount of data that is being generated caused a simultaneous increase in computation power, and a decrease of costs for cloud computing. This was the perfect scenery for companies to start analyzing the vast amount of data that was being stored 20 years ago, which caused a paradigm shift for companies that have decided to embed data analytics into their core business (Govindan et al., 2020).

With the rise of data analytics, a formal definition of data analysis was defined as “the process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusion, and supporting decision-making” (Xia & Gong, 2014). This is an agnostic definition of data analysis, independent of the vast universe of tools and methods that one can apply to perform such analysis. And like any other problem, following a semi-structured and defined process (Figure 12) will ensure that all the issues are addressed and that the appropriate steps are taken:

1. Problem definition and planning: the scope of the problem and the goals of the analysis should be defined and planned from the get-go, as well as the subject experts that will constitute the team that will be responsible for the analysis.
2. Data preparation: perhaps the lengthiest step of all, and one of the most important ones, it includes the collection, characterization, cleaning, and transformation of data into the appropriate forms to be processed.
3. Analysis: With the outputs from the two previous steps, the relevant data mining/analysis techniques should be selected and applied, with the relevant optimizations to wield the best results.
4. Deployment: The results of the analysis should be deployed in order to address the problem scope established in the first step. (Myatt & Johnson, 2014)

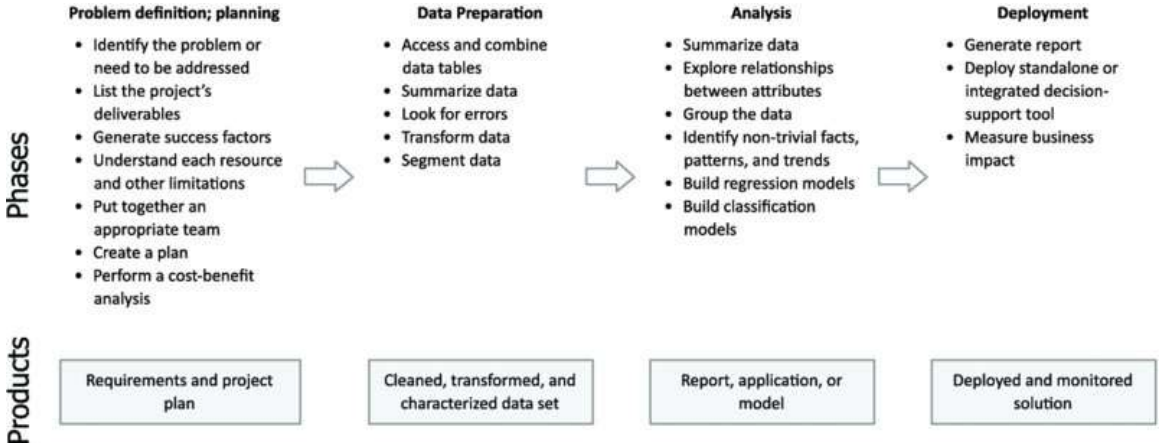

Figure 12 - Data analysis process description (Myatt & Johnson, 2014)

Of course, that this broad view of the data analysis process encapsulates a multitude of different techniques, tools, and concepts. During the analysis step one could fit the entire realm of data mining in it, as well as the whole concept of data visualization in the last stage of deployment of the solution or even communication of results. Data mining can be defined as "the process of discovering interesting patterns and knowledge from large amounts of data. The data sources can include databases, data warehouses, the Web, other information repositories, or data that are streamed into the system dynamically" (Han et al., 2012).

If we further dissect the concept, a pattern can be classically described as a parsimonious description of a subset of data points, and a model is simply a statistical description of the entire data set. With the advances in this area, we were able to extract increasingly complex information from data, which on the other hand caused new problems to emerge: the complexity of said structure obliges the user to understand the process and rational behind the final product, as well as the challenge of having an effective visualization process that accurately conveys the nature of the final models (which can be a very complex task to achieve) (Fayyad et al., 2002).

A visualization can be defined, at its core, as the communication of information using graphical representations (Ward et al., 2010). And it has been a very powerful tool since the dawn of times, with cave paintings used thousands of years ago as the first way of storing image data,

and complex visualizations being produced nowadays using programming languages and respective libraries (Python's Seaborn/Matplotlib or R's ggplot2/Plotly) or by robust BI software like Power BI, Sisense or Qualtrics.

Data visualization is, at the end, the front-line tool that will be used to convey all the work performed during the analysis and data mining stages, to the decision-makers and business users. It is also why it is such a crucial step, an incorrect or bad visualization can cause misinterpretation of information, which is a recurrent problem even in the most reputable and well-known institutions. And this recurrent problem is mainly caused by increasing technology advances, and the massive democratization of tools like Microsoft Excel, which allowed for virtually anyone to feed some data into a graphing application and produce a visualization. This remarkable situation opens exciting precedents for new advances in data visualization, but also opens the door to a lot of poor practices in the field (Knaflic, 2015).

3.4. Machine learning and AI

The natural evolution of data analysis/mining was without a doubt the concept of machine learning and artificial intelligence. The first concept of machine learning was first developed in the 1950s and the 1960s, and the main goal at the time was to create computer programs that would behave differently when introduced to different datasets (Mitchell, 1998).

Nowadays, machine learning refers to the components of artificial intelligence systems that learn from data to perform various tasks, and include the data, model, and processes for training, testing and validation. On the other hand, AI refers to computer systems that are able to perform tasks that are usually native to the human nature, like speech recognition, face detection or image classification (Burns et al., 2019).

Since machine learning consists of prediction algorithms that are dependent on the data that is used, the whole concept is intrinsically tied to the field of data analysis and statistics. Because it is such a general concept at its genesis, there is a variety of different subareas in which one could use machine learning to tackle different problems: text or document classification, natural language processing (NLP), speech processing tasks, computer vision applications, computational biology, fraud detection, recommendation systems, self-driving cars, and the list goes on and on. If formulated correctly, the majority of prediction problems that can be found in the real world can be tackled with machine learning, when the learning problem is casted correctly (Mohri et al., 2018).

There are three main types of machine learning: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning is all about learning the model from labeled training data that allow us to predict unseen or future data, meaning that for a specific set of training inputs, the desired output signals are already known, and the model is trained based on that experience. In reinforcement learning, an agent is developed that improves its performance based on interactions with the existing environment, using a reward signal, learning a series of actions that maximizes this reward signal via an exploratory trial-and-error approach or deliberative planning. Lastly, and contrasting with supervised and reinforcement learning, unsupervised learning deals with unlabeled data, meaning that the whole concept revolves

around exploring the structure of the data to extract meaningful information without the *apriori* knowledge of a outcome variable or reward function (Raschka & Mirjalili, 2019).

Amongst the most common machine learning tasks (as well as the ones that were studied extensively), is it worth mentioning:

- Classification tasks: the problem of assigning a category to each item (e.g., image/document/text classification, OCR, speech recognition, etc).
- Regression tasks: the problem of predicting a real value for each item (e.g., stock value prediction, forecast of economic variations, etc).
- Ranking tasks: the problem of learning to order items according to some criterion (e.g., web search, search engine page order, shopping cart recommendation, etc).
- Clustering tasks: the problem of partitioning a set of items into homogeneous subsets (e.g., social network analysis, large dataset analysis, etc).
- Dimensionality reduction tasks: the problem of transforming an initial representation of items into a lower-dimensional representation while preserving some properties of the initial subset (e.g., preprocessing images for computer visions tasks) (Mohri et al., 2018).

Fortunately, with the rise of the digital era and the ever-increasing advances in technology and reduction in computing costs, we are being able to consistently improve the generation of predictions for unlabeled data, as well as the production of robust and efficient algorithms that address such predictions, which is the main goal of machine learning and AI all along.

3.5. LINE-OF-BUSINESS APPLICATIONS

It was the mass production techniques and mass marketing of the mid-twentieth century that dramatically increased the product availability for end-consumers. With larger product availability came a shift in the way that the purchasing process had to be made, for businesses to keep up with supply and demand. This was the genesis of business applications, the need for solutions that could effectively link back-office and the company's touch points, the birthplace of traditional CRM/ERP technologies (I. J. Chen & Popovich, 2003).

Today's organizations need to have the ability to deliver vital business information to its workforce, and the technological landscape of today needs this delivery to be available from anywhere in the world, at any given second. It is this demand for mobility that pushed companies into implementing mobile business applications for the support of mission-critical processes, on par with the evolution of handheld devices (with increasingly more powerful portable devices in the hands of any layman) and with a revamp on the integration of web-based systems with back-end systems (L. Chen & Nath, 2004).

With all these advantages, it is only natural that the interest in the usage of mobile business applications in the context of an enterprise environment is increasing (Varshney & Vetter, 2002). Organizations who bet on the transition (for the correct use-cases) to mobile line-of-business applications with the intent of streamlining business processes may benefit from improved productivity, lower operational costs, increased customer satisfaction and better decision-making (Varshney et al., 2004).

The use of such technology in the corporate environment can be broken down into several silos: the technology characteristics, the task characteristics, the usage of the technology and the impact of such usage. Starting with the technology characteristics, LoB mobile applications distinguish from earlier ISs by their functionality (combining traditional communication systems with data processing), their portability (their ease of use independent of time and place), and their system performance and user support (how the business application itself interacts with hardware and software specs). When it comes to task characteristics, it is the context in which the technology is used and by whom, which includes the structure of the task itself (the type of task that the user usually performs), the frequency (how often said task is performed), the mobility of the task (the location of the user that is performing the task), and the need for emergency handling (the time frame in which a specific task needs to be performed following a triggering event). These factors plus the usage rate of the technology are the best metrics to infer the impact of the presence of mobile LoB applications in the enterprise context (Gebauer & Shaw, 2014).

IT vendors also evolved side-by-side with the product. Most major players like Microsoft, Salesforce, OutSystems evolved their range of offerings in the business applications space from simple .NET products, to mature CRM/ERP solutions, and more recently with low-code frameworks that can answer the ever-increasing demand for these business applications (stationary or mobile).

3.6. Process automation and RPA

A process can be defined as "a description and ordering of work activities across time and space that is designed to yield specific products or services while ensuring the organization's overall objectives and providing a conceptual basis for the integration and coordination of distributed resources, tasks, and individuals". These processes can be mapped into steps and actions for manual execution, or they can be mapped into workflows (which are process descriptions) for automatic execution by a workflow management system (Cichocki et al., 2012).

Process automation has been around since the 1900s, when the first concepts of automation were applied in assembly lines, specifically in the mechanical and electrical industries, which was a huge drive for mass production and had a dramatic impact on productivity (Christie, 2012). In more recent years, process automation has increased exponentially in its importance, having gain traction in all of the major industries – and with technological advances allowing for instrumentation systems to control complex processes, with increased reliability, safety, and advanced maintenance strategies (Jämsä-Jounela, 2007).

Although it is usually related to enterprise grade/end-to-end business processes, process automation can also be applied to more technical use cases of integration/orchestration, meaning that it is a very permissive technology that can be applied to a variety of different areas. Process automation can be divided into the following categories (although the boundaries are fuzzy between them and often overlap): human task management (the workflow engine controls the sequence of steps, but humans do the actual work), straight-through processing (fully automated and integrated processes that use APIs to transfer data between systems),

orchestration (used when IT components are coordinated using a workflow engine), and challenges related to distributed systems (directly tackling the issues revolving around the inherent unreliability of networks) (Ruecker, 2020).

Amidst the vast offering available around process automation, there is also robotic process automation (RPA). Unlike traditional workflow technology, with RPA the information system remains unchanged, since RPA itself is an umbrella term for tools that operate on the user interface of other computer systems in the way a human would do (Aalst et al., 2018). At its essence, RPA can be defined as "tools that perform 'if, then, else' statements on structured data, typically using a combination of user interface (UI) interactions, or by connecting to APIs to drive client servers, mainframes or HTML code. An RPA tool operates by mapping a process in the RPA tool language for the software 'robot' to follow, with runtime allocated to execute the script by a control dashboard." (Tornbohm & Dunie, 2017).

RPA tools have seen a spike in demand, which is justified by the fact that organizations are looking for ways to cut costs and to quickly link their in-house/legacy applications together, which is perceived by organizations as a clear path for a higher Return of Investment (RoI) (Aalst et al., 2018). However, it is important to establish the differences between RPA tools and traditional workflow technologies, since they are fundamentally different and address different use cases as well: RPA sits on top of existing systems and access these platforms through the presentation layer, that is why these tools do not require programming skills for software interface configuration; in addition to this, RPA does not create a new application nor it stores transactional data, it does not require a data model/database like most BPMS systems (Aguirre & Rodriguez, 2017).

Despite all the advances in process automation technology and RPA tools, at the end of the day the desired business results can only be achieved if the execution of the processes is successful (Scheer et al., 2004). That is why we are seeing so many flavors of offerings from different vendors across the market, with players more focused on classical process automation like Bizagi or Nintex, others dedicated to pure RPA tools like UiPath, Blue Prism or Automation Anywhere, and even vendors that are combining both offerings into a single technology, as it is the case with Microsoft Power Platform's Power Automate (Tornbohm & Dunie, 2017).

3.7. NLP AND CHATBOTS

Natural language processing (NLP) is a subfield crossover between linguistics and AI, and can be defined as "the function of software or hardware components in a computer system which analyze or synthetize written language." – with the epithet "natural" being used to distinct formal languages (programming languages, mathematical notations, etc) from human speech (Jackson & Moulinier, 2002).

Because of its usual huge quantity of data, classical approaches to the processing of natural language texts can be divided into the following sections (keeping in mind that this preprocessing of textual data is absolutely crucial for the whole effectiveness of the NLP context, which is entirely dependent on the quality of the data): the surface text data, the tokenization

phase, lexical analysis, syntactic analysis, semantic analysis, pragmatic analysis and finally the speaker's intended meaning (Indurkhya & Damerau, 2010).

Amongst the countless different data processing techniques where textual data is concerned, it is important to briefly describe each of the main methods. Tokenization refers to the processing step of dividing the input text into units (tokens), individualizing each word/punctuation mark as its own element. Capitalization aims to standardize all the strings from the dataset, usually converting all text data to lowercase for simplicity sake. Stop words and its removal is essential in NLP tasks, since these are function words that can be ignored in keyword-oriented information retrieval, without impacting the accuracy of the final result, and downgrading the complexity level of the dataset, which is always desired. Lastly, stemming and lemmatization are also applied for retrieving the simplified form of morphological analysis of a given word, usually stripping off any affixes, preserving the core stem of the word. There are several more techniques for text preprocessing, but these are the core ones that are present in any NLP toolkit (Manning & Schutze, 1999).

Due to its vast range of techniques and scope of work, there are a lot of different applications in the field of NLP: information retrieval, information extraction (recognition, tagging and extraction into a structured representation, utilized in question-answering applications, visualizing and data mining), question-answering (system provides a list of potentially relevant answers in response to a user's query), summarization (implementations that can reduce the size of text, while maintaining a coherent discourse), machine translation and dialogue systems (applications that can maintain a conversational context based on the input exchange with an end-user, e.g. chatbots) (Liddy, 2001).

And among these different applications, chatbots are without a doubt one of the most popular ones, since they marked a shift in the way that individuals interact with data and cloud services (Brandtzaeg & Følstad, 2017). This rise of popularity in chatbots was caused by the advances in the fields of machine learning and AI, with tech giants making like Microsoft, Google and Facebook placing their bets on the technology – Microsoft CEO Satya Nadella even stated on the 2016 edition of the developer event Microsoft Build "Bots are the new apps" (Cava, 2016).

Chatbots are agents that serve as user interfaces for natural language processing capabilities, operating under the umbrella of a data/service provider (Dale, 2016). This technology is overwhelmingly present and targeted at mobile messaging platforms (Følstad & Brandtzaeg, 2017), with messaging titans like Facebook Messenger and Microsoft Teams popularizing the technology into more come-of-age messaging platforms like Slack or Viber.

However, the pervasive presence of chatbots is not limited to messaging platforms, they can also be used as a complement to existent services like websites or online forms. In addition to this, they pose an interesting and promising alternative to traditional customer service or any kind of marketing purposes, since they represent a mean for direct customer engagement through text-messaging, without the need for the development and maintenance of targeted webpages or applications (Xu et al., 2017).

4. TOOLS AND TECHNOLOGY

In this chapter, we will make a deep dive in all the different tools and technologies that are part of the Power Platform. All the different components will be described in detail in terms of architecture and functionality, as well as the respective synergies that exist between this low-code platform and all the pro-development extensibilities that are made present by the underlying cloud infrastructure.

4.1. POWER BI

Microsoft Power BI addresses self-productivity and corporate needs for on-demand analytics and business intelligence by comprising a set of software services, connectors and tools that bundle together a variety of data sources into immersive and interactive insights. It serves the citizen developer by providing a way of quickly generating insights from a simple data source like an Excel workbook in an easy manner, but also serves as an enterprise-grade and robust BI tool, by enabling real-time analytics and connection to cloud-based or on-prem data warehouses. In addition to this, pro-developers can also find in Power BI an easy way of making custom development, being it by building ad-hoc custom visualizations or by connecting to complex data systems. Whatever the audience is, the goal of Power BI is to allow everyone to do more with their data, ranging from a single individual to whole global corporations, with the aim of facilitating the decision-making process.

Power BI branches out into three different elements: Power BI Desktop which is a Microsoft Windows desktop application; Power BI Service, an online SaaS solution; and Power BI Mobile which is a set of native mobile BI apps available for Windows, iOS, and Android. Collectively, they bring the necessary features for anyone (no matter their role) to effectively create, share and consume dashboards and insights.

Figure 13 - Power BI Desktop

Figure 14 - Power BI Service

Figure 15 - Power BI Mobile

These three elements are the core pillars of a common business intelligence pipeline: the ideation and creation of the report (from data fetching and preprocessing to visual layout) in Power BI Desktop, publish the report to the Power BI Service and lastly share the content with other people, allowing them to consume insights via the service or on a mobile application.

Power BI is composed of several core building blocks, that when put together and expanded, can create complex and more advanced reports. The first of these building blocks are visualizations (or visuals), which provide a visual representation of data. These visualizations can take many shapes and forms, since there is a huge variety of them inside Power BI, which also provides with the possibility of creating custom developed visuals and is always releasing new out-of-the-box visualizations. These can be a simple matrix, charts, color-coded treemaps or more complex animated visualizations. They vary from simple artifacts to very complex ones, but they all share the same goal, to accurately and easily present data in way that provides relevant information and context to the end-consumer.

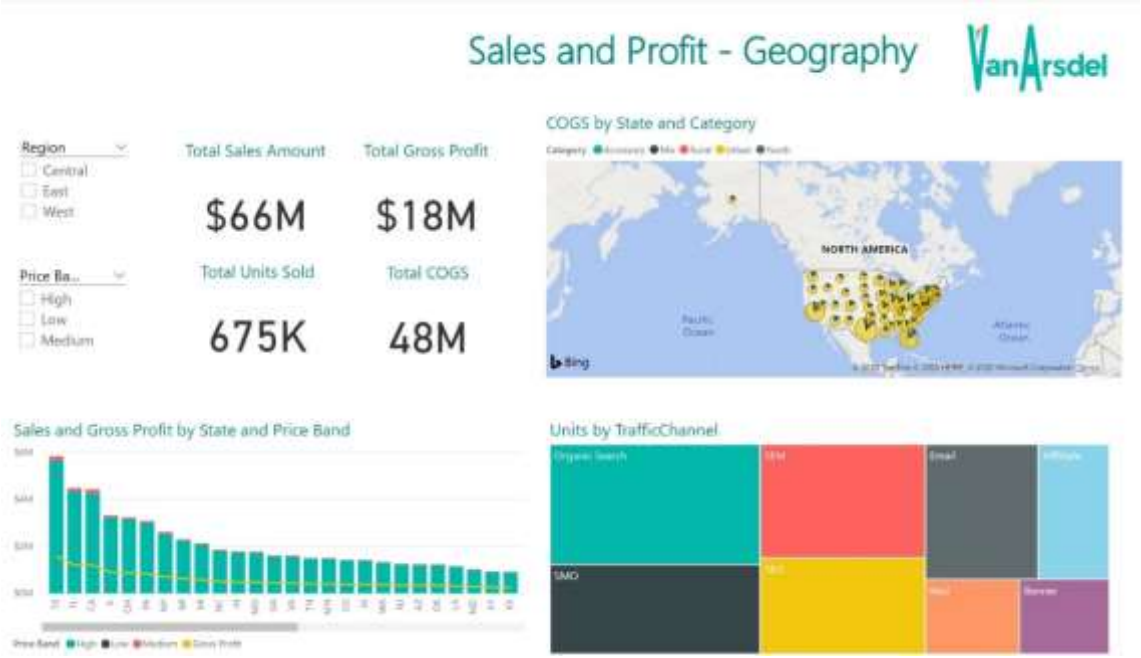

Figure 16 – Different types of Power BI visualizations

One of the other Power BI building blocks are datasets, which function as the collection of data which Power BI uses for creating visualizations. Datasets can be based on a single data source, or a multitude of different ones, providing the flexibility of filtering and working with all the data from the company, no matter where it resides. These functionalities make it easy to combine data and by leveraging the vast array of data connectors you can easily create for example a dataset from three database entities (being it Excel, Azure SQL or Oracle, just to name a few), a service like Salesforce, Twitter or MailChimp and even combining all of these with data that resides in an independent website through the web scraping tool. This unique combination creates a single dataset, albeit its origin derives from a variety of different sources.

In addition to these, another crucial building block of Power BI are reports. Reports are collections of visualizations that are fueled by the data that was configured and preprocessed in the dataset, and they can be bundled together in one or multiple pages. Essentially, it is the

perfect container for a collection of visualizations that can be arranged and customized with extremely deep granularity, to give organizations the necessary freedom to discover the way their data best tells its story.

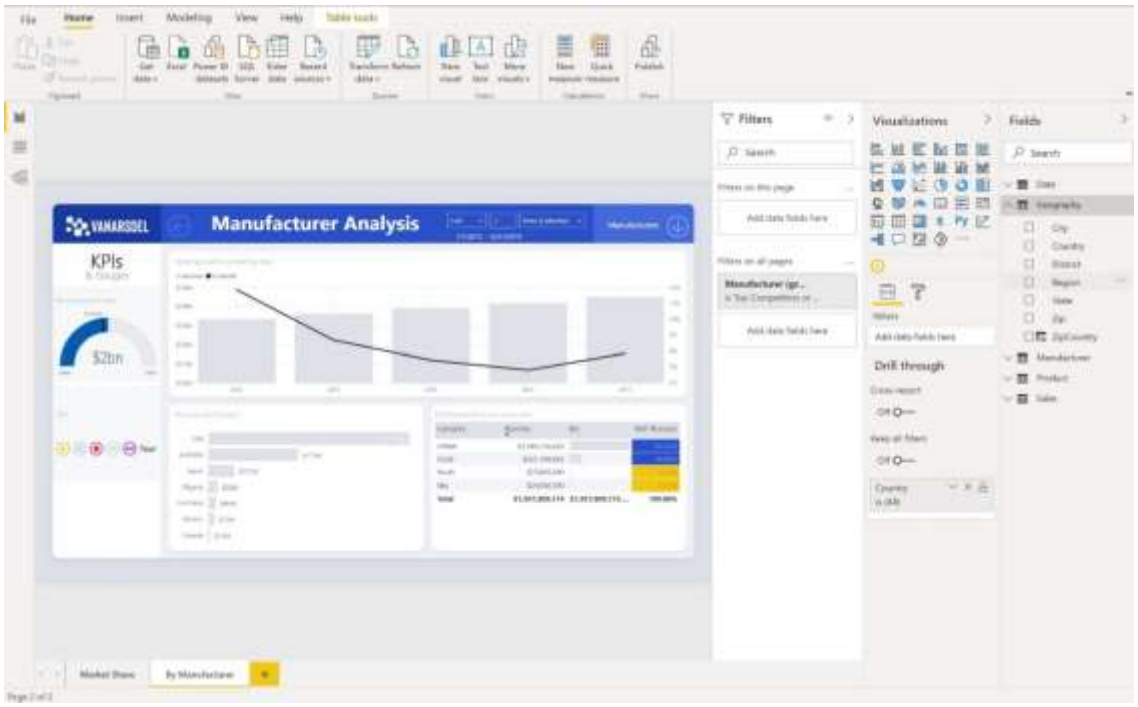

Figure 17 - Power BI report in Power BI Desktop

Lastly, dashboards and tiles are the final Power BI building blocks. A dashboard is a collection of visualizations from a finished report that is presented in a single page, that can be easily shared with other users and groups. Tiles are the placeholders for visualizations in reports and dashboards, one tile holds one individual visual. These can be moved around and resized by the report/dashboard makers, to ensure that the desired layout is achieved.

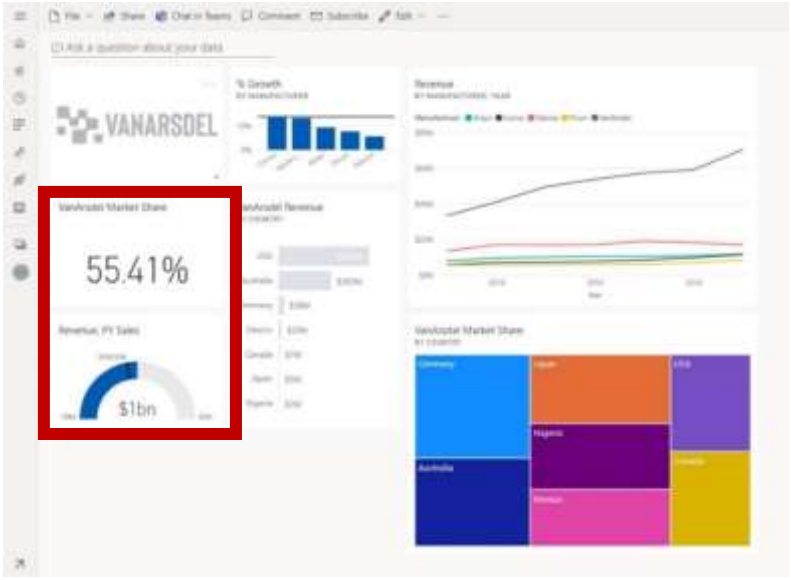

Figure 18 - Power BI dashboard with one of the tiles highlighted in red

Power BI Desktop is the tool where all these building blocks come together. BI development begins by bringing the necessary organizational data into Power BI Desktop, leveraging the dozens of data connectors to cloud services and other generic sources (Figure 19). Depending on the used data source, a user might be prompted to sign in and authenticate the request or access it anonymously if the data is public and does not require user authentication.

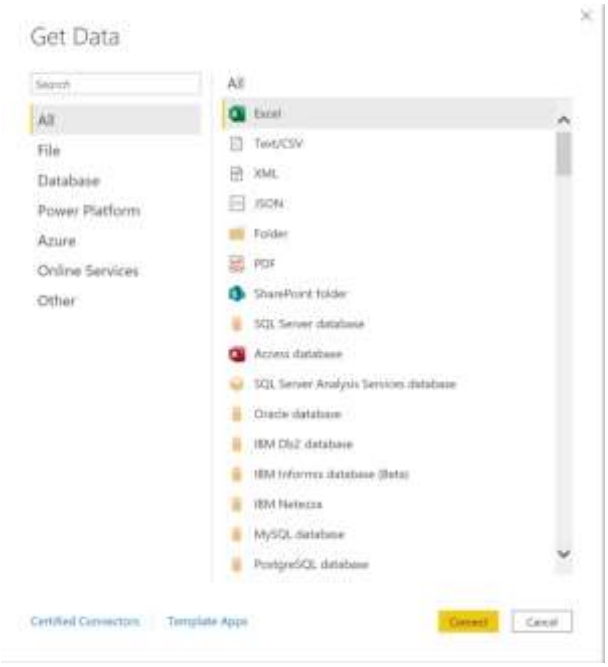

Figure 19 - Power BI data sources sample

It is when the data is brought to Power BI Desktop via a data connection that we can use the capabilities of Power Query. Power Query is an ETL engine for data preparation and transformation, allowing users to reshape through a graphical interface the data that is coming from different data sources. Power Query can be found not only in Power BI, but also in many other services like Microsoft Excel, the Common Data Service and Azure Data Lake Storage for example.

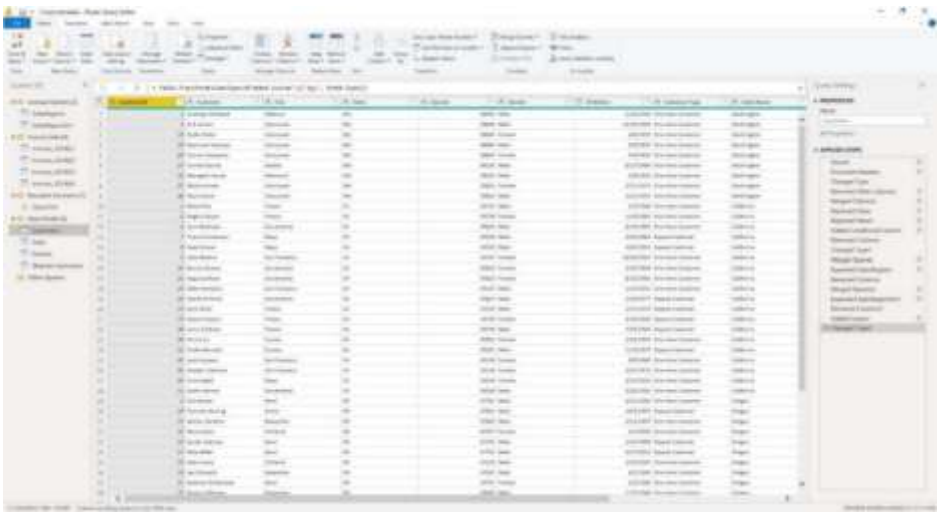

Figure 20 - Power Query Editor in Power BI Desktop

Power Query's UI (Figure 20) is constituted by a set of different menus, buttons, and ribbons in a user-friendly way, making hundreds of different transformations and classic data operations and preprocessing techniques easy. The interface also includes a display that previews the underlying data and the respective transformations steps, and these capabilities are independent from any existent limitations in the inherent data sources. These data transformations can be as simple or as complex as they need to be, and the engine automatically creates the required Power Query M formula language in the backend when a new transformation step is created. This M scripting language functions as the abstraction layer between the data transformations and the query itself. This brings the flexibility of addressing more complex use-cases of data preprocessing, where perhaps the Power Query Editor UI is not sufficient for a given transformation – through the Power Query's Advanced Editor (Figure 21), it is possible to access the full M script of the query and fine-tune with extreme granularity any necessary functions or transformations.

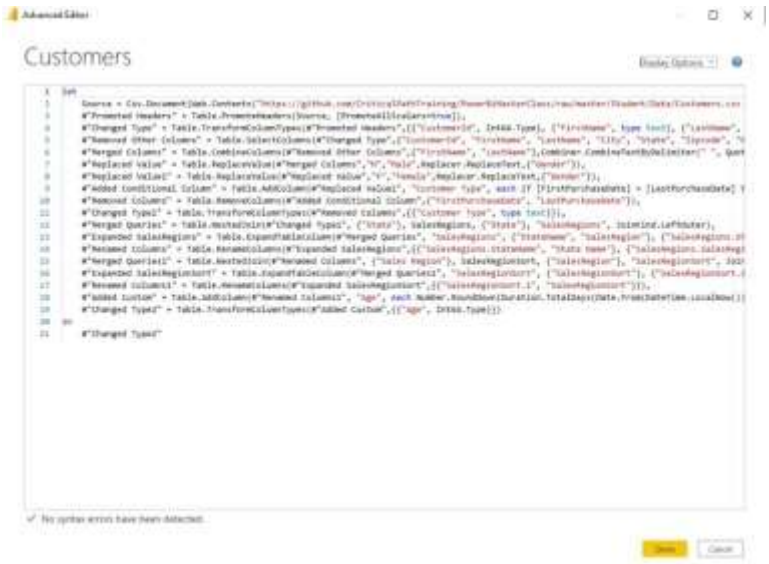

Figure 21 - Power Query's Advanced Editor with a sample M query script

The common Power BI Desktop workflow can be divided into three areas: data modelling, data analysis and data visualization. Data modelling is by far the most important phase of any BI implementation, since a good data model allows for more effective maintenance and report creation, more performant data exploration, and more accurate insights for the end user. A data model in Power BI (Figure 22) can be as simple or as complex as it needs to be, but at the end of the day it is a varying number of tables that are coming from multiple data sources (already preprocessed through Power Query if they need be) and with relationships between them. One can create various kinds of relationships depending on its cardinality, making use of any primary and foreign keys that the data model has.

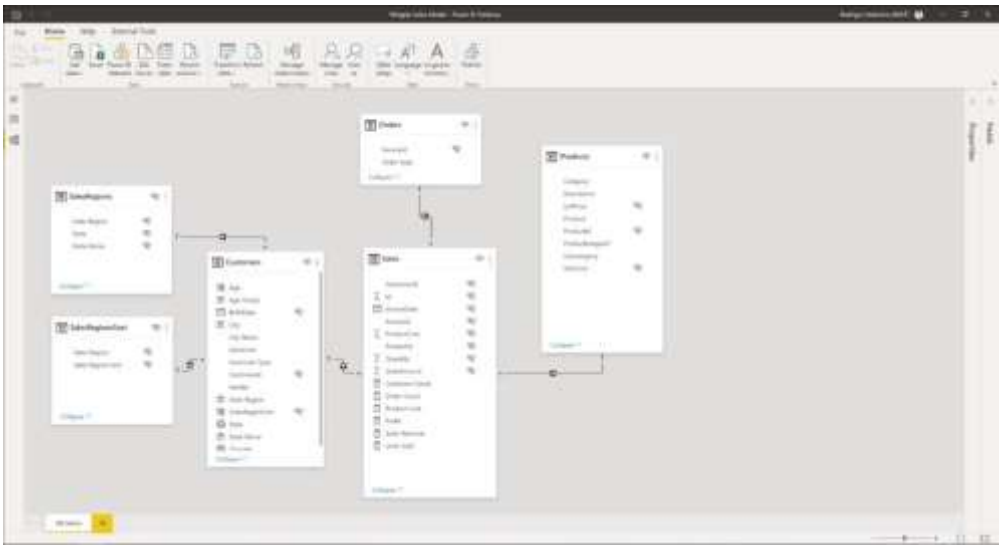

Figure 22 - Example of a data model in the Model pane of Power BI Desktop

When it comes to designing a data model, it is considered a best practice amongst Power BI analysts to try to follow a star schema. A star schema is a popular method of schema design, which is built for high usability and performance, and assumes that every table in a dataset is either defined as a dimension table or a fact table. Fact tables often include observational data values like sales orders or any other kind of transactional data, whereas dimension tables contain the details regarding the data in the fact table, like customers, locations, products, etc. In report building, a star schema optimizes performance since we can leverage dimension tables that are connected to fact tables through any type of relationship, to filter and group the data in the fact tables using measures and visualizations.

It is in the advanced analytics capabilities of Power BI that DAX really comes into play. DAX is a formulaic language that is used across several software, including Power BI, Azure Analysis Services and Power Pivot in Excel. It provides the flexibility in its structure to perform advanced statistical operations, advanced calculations, and queries on data present in tabular data models (like the ones in Power BI datasets). The primary use cases for DAX include the creation of custom measures, calculated columns and tables, and more advanced scenarios like row-level security.

Measures are formulas that are calculated dynamically depending on the context that they are executed, that is why they are mainly used in a reporting context, since its output depends on the data model combinations and filters that are applied in the client's side (in this case, the actual Power BI report). This way, the measure evaluates each context at runtime and fetches the relevant data to each cell with the formula applied. These measures (Figure 23) can be as simple as a sum of all sales amount in a data column or they can be more complex in order to tackle more demanding reporting requirements. On the other hand, calculated columns and tables produce computed objects that are generated by DAX and are then stored in-memory with the remaining data model. While calculated columns are created by validating a DAX formula against every row of a given table, calculated tables derive from all or some tables of the data model and, upon creation, behave has any other table in the model, supporting any kind of necessary formatting and relationships with other tables.

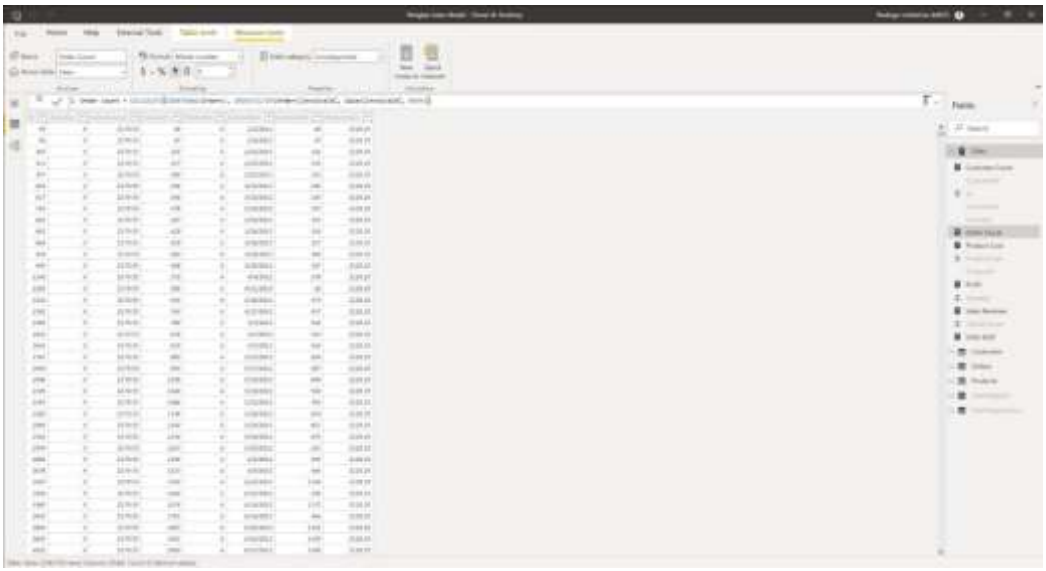

Figure 23 - Example of a measure written in DAX in the Data pane of Power BI Desktop

By combining the power of DAX and the out of the box features of Power BI visuals, it is possible to democratize advanced analytics capabilities to business users who might not have an extensively technical background. Statistical analysis, clustering techniques, grouping, and binning data, outlier identification and time-series analysis were tasks that were reserved strictly to data analysts and are now made available to proficient citizen developers and made easier to BI professionals.

After the data is modelled and transformed, it is time to visualize it. Data visualizations is what differentiates a spreadsheet from effective comprehension and retention from the end user, using Power BI visuals. They are an essential aspect of any reporting activity, because it is the final abstraction layer between the raw data that data analysts prepare and the insights that the report audience will consume to make informed business decisions.

Power BI comes with plenty of native visuals, and in addition to these, it is possible to bring even more visualizations created and distributed by third party entities (Figure 24). Using all these visuals, it is possible to further format and customize the visuals in countless different ways, this makes sure that the analyst has a wide range of functionality when it comes to report building. The native visuals range from traditional bar, column, pie charts to mapping visuals and gauge for KPIs or even complex 3D visualizations. One can also directly embed R script and Python visuals in the report, by leveraging popular libraries for data visualization directly in the Report pane of Power BI Desktop. Power Apps are also one of the native visuals within Power BI, allowing for data entry and an applicational interface directly inside the report, showing a tight integration between all the components of the Power Platform.

Figure 24 - An "Aquarium" custom visual in the Report pane of Power BI Desktop

Power BI is also very tightly integrated with a variety of AI features. To make for a clearer explanation, let's divide the AI capabilities of Power BI into two different areas: AI from a data layer perspective and AI from the visualizations' perspective. Together, these AI powered analytics provide a seamless experience for end users, data analysts and data scientists alike.

Firstly, at the data layer, one can use Power BI in conjunction with Azure ML, AutoML and Azure Cognitive Services. Nowadays, there are plenty of organizations that use Machine Learning models to have better insights and predictions about their business, and with the tight integration that is provided by Power BI and Azure ML, the users who have permission to do so can leverage automatically these Azure Machine Learning models, giving them the ability to visualize and invoke insights from these models, embedding it in reports and dashboards, providing business users who need them with powerful AI analytics. These Azure ML models are usually built by data scientists from the ground up, to address a specific prediction or forecast, are then made available to the business analysts to consume in Power BI.

When it comes to Automated ML, business users can train, validate, and invoke ML models directly in the Power BI service. This follows a wizard-like procedure where analysts can specify the data to train their model (Figure 25) (AutoML supports classification, regression and forecasting tasks), and the service will automatically extract the most relevant features, the

Figure 26 - Power BI AutoML

Figure 25 - Power BI AutoML model explainability report

most efficient algorithm is automatically selected, and it is then fine-tuned and validated. After the model is trained, Power BI automatically generates an explainability report (Figure 26) which includes a confusion matrix, AUC percentage and visualizations, detailed recall, precision and key influencers metrics, model quality over the training iterations and if the selected model uses ensemble learning, it also shows the weight of each constituent in the ensemble model.

Lastly, at the data layer, Azure Cognitive Services are a set of pre-built AI algorithms that can leveraged out of the box to address a specific business scenario. As of today, there are four Cognitive Services that can be used within the data preparation stage in Power BI: Sentiment Analysis (detecting positive or negative sentiment from textual data), Key Phrase Extraction (evaluating unstructured text data and returning the main points and key list of phrases), Language Detection (provides a language identifier from text data) and Image Tagging (provides textual identifier tags that are recognized through computer vision from image data).

At the visualization layer, there are several visuals with AI capabilities that can unlock advanced analytics scenarios, even for less technical Power BI users. The most prominent features include Q&A, Quick Insights and Forecast, R and Python visualizations, Decomposition Tree and Key Influencers. The Q&A visual (Figure 27) enables report consumers to explore data using natural language capabilities and receiving answers in the form of graphs and charts. By simply typing a question in the Q&A visual, it will provide results that iterate solely on the underlying data in the Power BI model, also enabling the user to pin the outputted visualization directly into a report or dashboard.

Quick Insights comprise a series of AI functionalities that allow users to further explore the data in the report. These can be run before the report building process, against the whole dataset, to generate quick insights on the data before going deeper into the visualization phase but can also be used at the dashboard tile level to gather scoped insights. These functionalities include finding where a distribution is different, explaining the increase or decrease of a specific value in a visual (Figure 28) as well as automatic forecasting on visuals that use date/time fields.

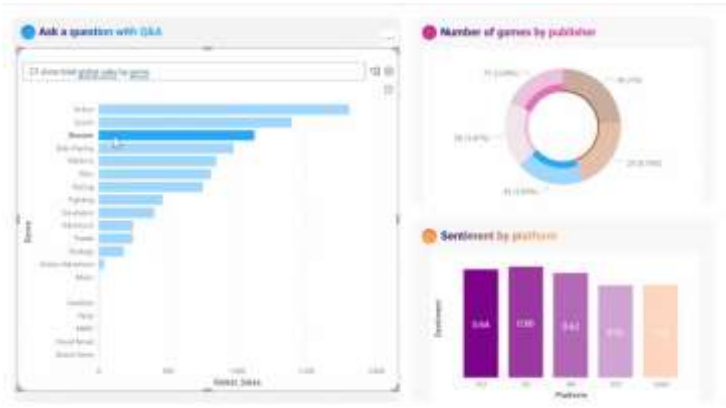

Figure 27 - Q&A visual with a user-entered question and an outputted visualization

Figure 28 - Quick Insights explaining the percentage decrease in a report's visualization

Power BI also provides close integration from a visualization perspective with other scripting languages like Python or R. This means that a data analyst can run Python or R scripts directly inside Power BI Desktop, and import the produced datasets into the Power BI Desktop data model for further analysis. The Python and R visualizations (Figure 29 and 30 respectively) are then rendered in the Report pane, and they behave with the same level of interaction as any other Power BI visual – anytime a report consumer slices and interacts with the data, the scripts are automatically rerun to reflect the changes in the visualizations.

Figure 29 - Python visualization in Power BI Desktop

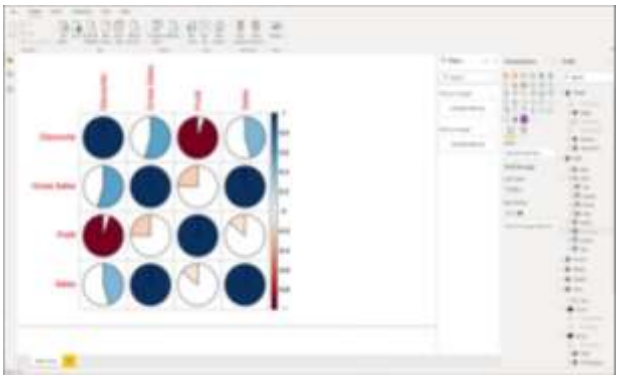

Figure 30 - R Visualization in Power BI Desktop

If the use case asks for data visualization across multiple dimensions, the Decomposition Tree (Figure 31) is the best suited visual in Power BI. It presents granularity and detail without the need to drill-through the visual, presenting immediate insights with interactive filtering, automatically aggregating data and allowing to explore all the dimensions in any given order. Because of the AI capabilities inherent to the visual, the report consumer also has the option to ask the visual to find the next dimension to explore next based on specific criteria. These features make the Decomposition Tree a particularly effective visualization when it comes to conducting root cause analysis or other ad hoc exploration scenarios.

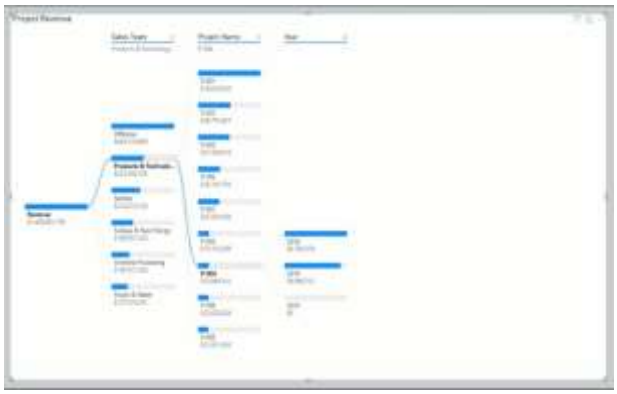

Figure 32 - The Decomposition Tree visual

Figure 31 - The Key Influencers visual

Lastly, the Key Influencers visual (Figure 32) automatically analyzes the data by ranking the important factors based on a metric and presents them as the key influencers. This visual is particularly helpful when the end-user is trying to understand the factors that drive a specific metric that they are trying to analyze, as well as contrasting the relative importance between these factors.

4.2. POWER APPS

Microsoft Power Apps is an aggregation of different applications and services which make use of a rich set of data connectors and the Common Data Service (explained in further detail in chapter 4.5) to build custom business applications that target enterprise needs. This provides organizations a way of quickly developing LoB applications independently of where their data resides (being it Microsoft owned systems like SQL databases in Azure or third-party services like Zendesk or Amazon Redshift).

Contrary to the Power BI building blocks, the authoring experience in Power Apps is completely web-based. It consists of three core tools: the Power Apps Home Page (Figure 35) (where users can decide what type of app they want to build, use any pre-built templates for accelerating development, etc), the Power Apps Studio (Figure 34) (the actual authoring canvas for the apps, connect to any kind of database, building the actual UI and arranging the business logic through Excel-like formulas) and Power Apps Mobile (Figure 33) (which is a mobile application available for Windows, iOS and Android used to run the actual Power Apps.

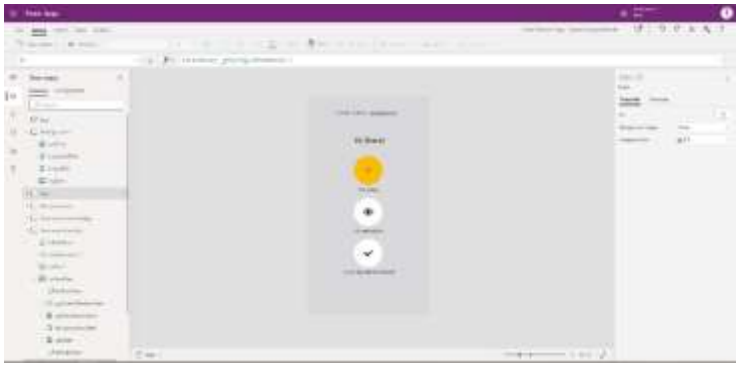

Figure 34 - Power Apps Studio

Figure 33 - Power Apps Mobile

Figure 35 - Power Apps Home Page

Power Apps can be consumed in virtually any type of mobile device but can also be consumed in the browser or in embedded scenarios. Some of these embedded scenarios include embedding the app in a SharePoint site, directly in a tab in a Microsoft Teams channel or even by embedding it in a Power BI report to allow report consumers to insert data directly from the report. These seamless integration across Microsoft technologies brings a lot of flexibility when tackling use-cases from customers and provide a very rich experience for end-users.

There are three types of Power Apps that one can create, and each type is suitable for different scenarios. Arguably the most used ones are Canvas Apps (Figure 38), these are the ones that the

app maker designs with pixel-perfect granularity regarding the app's UX/UI, has complete control over the different screens, components and controls, formulaic business logic that is implemented in the Power App as well as combining data from a multitude of data sources through the built-in data connectors in the same application. The scenarios covered by Canvas apps are virtually limitless since the developer has complete control on the functionality and outcome of the application, making this type of Power App extremely suited for any kind of line-of-business scenarios or frontline use-cases.

In addition to Canvas apps, there are Model-Driven apps (Figure 37). Model-Driven applications differ from their counterparts by using Common Data Service entities as the primary building blocks, and by having an automatically generated responsive UI across devices, out of the box offline capabilities and fewer options for the developer when it comes to customization. These types of Power Apps start from the data model and build upon the business data, logic and processes that reside in the Common Data Service, with the final interface being composed of model forms, configurated views and other components. Because Model-Driven applications function almost like mini-CRM/ERP instances derived directly from Dynamics 365, they are usually used to tackle more backend scenarios of data entry and information management.

Lastly, the final type of application is Power Apps Portals (Figure 36). While the other types of Power Apps described above serve LoB scenarios, delivering solutions that interact with business data to users inside the organization, Portals extend access to Common Data Service entities to external and internal audiences alike. Users can develop Power Apps Portals using a hybrid configuration panel that brings some of the granularity of the Studio for Canvas App, and the no-code components that are built upon Common Data Service artifacts like forms or views. The final product can be described as a website/portal that can be used by external users, that can leverage the most common OAuth authentication methods like Google, Azure AD, Facebook, or LinkedIn, providing an easy way for external audiences to interact with a company's business data hosted in CDS.

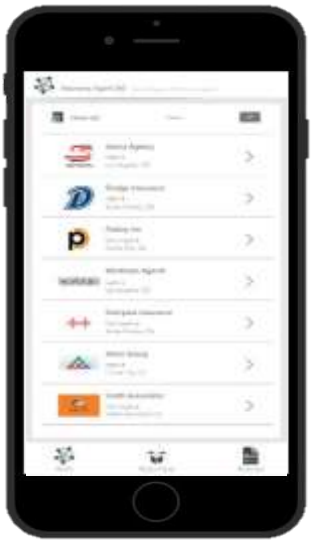

Figure 38 - Canvas app consumed in a mobile device

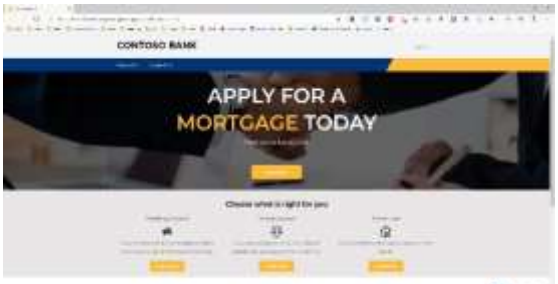

Figure 36 - Power Apps Portal

Figure 37 - Model-Driven app

The scope of this thesis will focus primarily on the Canvas type for Power Apps, since Model-Driven applications and Power Apps Portals overlap with the technical expertise and duties of the Dynamics 365 TSs.

When designing business applications through Power Apps, much like Power BI reports, there is a design process that needs to be followed to ensure that the final solution meets the expectations of the consumers. This process differs from organization to organization, but usually consists of correctly understanding the business requirements (understanding what is the problem that this business application will need to solve, making sure that all company policies are respected), knowing which data sources will need to be used, pre-requisites regarding theming or overall UX/UI and the business logic that needs to be implemented, being it at the application level or built in the underlying schema of the data model.

At the time of writing, Power Apps (and the remaining Power Platform components) can take advantage of more than four hundred data connectors to all types of cloud and OnPrem services, Microsoft native as well as from third party entities. If for some chance there is not an out of the box connector that can interact with the data from a particular system, more seasoned developers can create custom connectors based on the service's API to publish a connector to said system, that can be then used by the remaining citizen developers. This facilitates the average citizen developer in producing business solutions by allowing them to connect directly to where their business data resides, without having to make complex integrations every time there is a need for a business application. This combined with the simple layout of the Power Apps Studio that resembles common Office 365 applications like Power Point and Excel, makes it a very powerful tool with a low barrier to entry for business users in any organization.

The Power Apps Studio can be divided into four main areas: the left section pane (Figure 39 [1]) can be alternated between the Tree View (a hierarchical collapsible display of all the screens, controls, components and media artifacts that are present in the app), the Data Sources View (in which one can see what data sources are connected to the Power App and connect to additional ones if there is that necessity), and the Media View (which shows all the media artifacts in the application, being it images, video or audio data). The middle section of the Studio is the actual canvas for the application and is the section that takes up the most real estate (understandably so), in which the controls and components that the developer uses are rendered at runtime, allowing for a very visual display of the application itself.

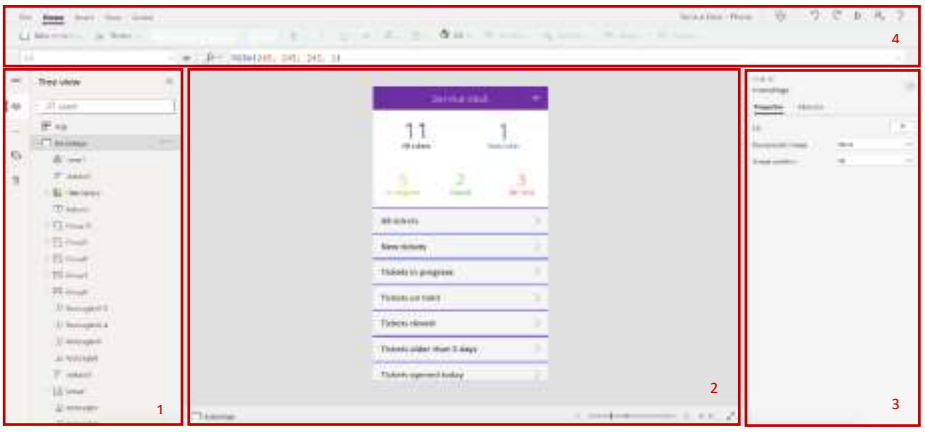

Figure 39 - [1] left pane/tree view, [2] canvas, [3] properties pane, [4] top section with formula bar and tabs

The Tree View works as a developer navigation shortcut between all the different screens and UI elements, with any interaction made in it being reflected in the actual canvas of the application. The canvas works very similarly to the workflow in a Power Point slide, users can simply drag and drop controls into the screen and resize and move them around the screen's real estate. These controls range from common labels, text inputs and buttons, controls that perform data shaping operations like forms, galleries, and data tables, to more advanced or specific controls at the end of the spectrum like Mixed Reality controls, AI Builder controls (explained in further detail in chapter 4.5), barcode scanners and more. Each type of control is tied to a specific set of properties inherent to said control, which are highlighted in the right pane of Power Apps Studio when a given control or screen is selected, allowing for the developer to quickly customize the properties of the control. Like it was said before, the properties change depending on the control, but these include formatting options, UX behaviors that can be configured depending on specific user interactions, navigation between screen and so on. These properties customizations are usually made easier and more granular using Power Apps own formulaic Excel-like language, which brings us to the last section of Power Apps Studio, the top area/header. This section contains a formula bar very similar to the one business users use in Excel, that allows developers to use Power Apps formulas and tie them to specific properties in each control and screen, giving the flexibility to shape the final UX and business data in the overall business application. These formulas range from variables declaration, data transformations, different techniques for data loading in the Power App, navigation between screens or conditional formatting of specific controls. Because Power Apps can be consumed in a variety of different devices, the developer can also leverage these formulas to access device-specific functionality, like for example accessing the current coordinates of an end-user through the GPS functionality of his/her mobile device while the individual is running the application. It is also in the top section of the Power Apps Studio that developers can add new screens and leverage existent pre-built templates of new screens for common application use-cases, like a calendar or a detail form, or specify a specific template/color palette to be used throughout development. The developer can also leverage the close integration between Power Apps and Power Automate (described in further detail in chapter 4.3), and bind a Power Automate flow that brings limitless automation possibilities to the application (for example the end-user clicking on a button in the Power App to submit a set of data that needs to be approved by his/her immediate manager before patching that data into the data source).

Power Apps, as with any other Power Platform component, allows to preview the result even during the development stage. This gives the developer the possibility of getting a sneak peak of the end-users' experience as he/she makes any change in the application, and live debug as new functionalities are being added to the Power App. Lastly, another very important feature of Power Apps Studio, is the App Checker: these set of features give feedback to the app maker regarding incorrect formulas in the Power App, performance recommendations based on changes that can be done to the app structure to improve processing times, and even an accessibility checker, to make sure that the Power App being developed is accessible to any kind of audience. These tips include missing accessible labels, color-blind friendly color palettes, just to name a few.

Once the Power App is developed, it is important to test it before publishing it out for end-user's consumption, it is a crucial part of any CI/CD pipeline. Testing ensures that the app quality is maintained and continuously validated at the end of the development phase, or even after there are any updates made to the app, identifying any problems in an earlier stage of the publishing process, and providing an easy and sustainable way of correcting and hot fixing any issues to enforce the reliability of the application. At the enterprise scale, it is not feasible to manually test every business application that is put into production, that is why the option of test automation brings a minimal effort approach to software testing.

Power Apps provides a Test Studio (Figure 40) for this effect. It is a tool built using the same low-code philosophy as the remaining platform, that provides developers with an automated way of writing and organizing tests for canvas apps, either by using Power Apps expressions or by using a built-in recorder that saves app interaction automatically, much like a small RPA-like technology. These features provide validation for the apps' features and developers can embed these tests into the app deployment process.

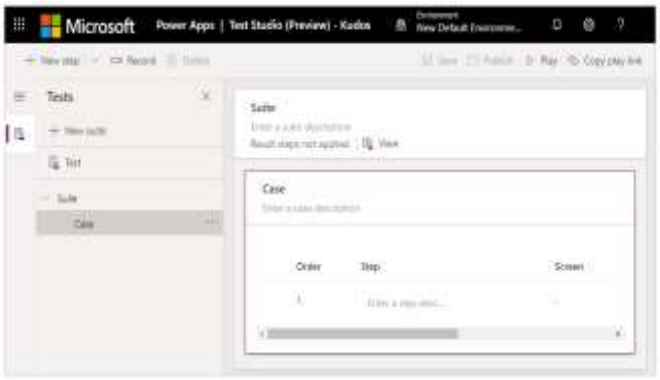

Figure 40 - Power Apps Test Studio

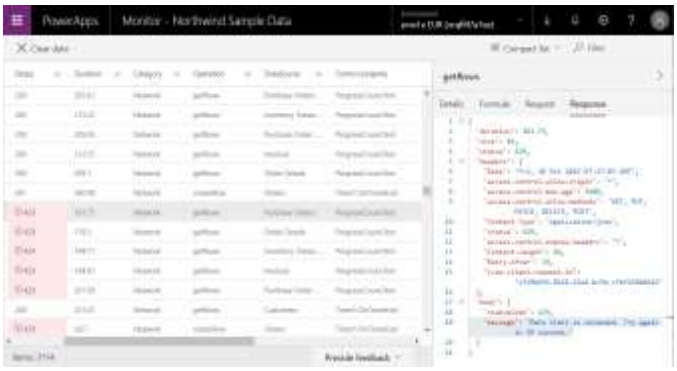

Figure 41 - Power Apps Monitor Tool

In addition to the Test Studio, Canvas apps can also leverage the Monitor tool (Figure 41). This tool provides the developer with the ability of analyzing a running log of all the actions that are performed within the app, to better understand the impact that each formula within the Power App has in the overall data flow. It details network calls with a tracer-like behavior, showing how long a given call took to take place, the amount of data that it fetched and even if there was any erroneous behavior within that call. This is also a tool that a more seasoned developer can use to further debug Power Apps and improve their performance.

Like every Power Platform component, Power Apps provides pro-development extensibility to cater to more sophisticated use-cases. This is where the Power Apps Component Framework comes into play, this framework allows developers with technical expertise to build custom and reusable code components that deliver specific functionality that is not made available by the out-of-the-box components. Developers can leverage existing client frameworks like React, AngularJS and TypeScript together with the Power Apps Command Line Interface, easily integrating with any preferable IDE like Visual Studio Code.

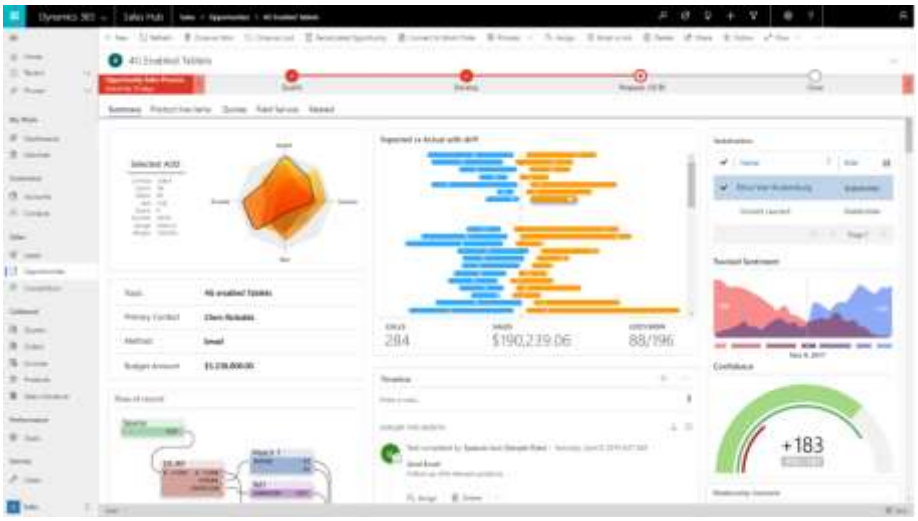

Figure 42 - Different Power Apps custom code components

Developers can leverage the Power Apps CLI to seamlessly create these code components, package them and deploy them as any other Power Platform artifacts, opening possibilities to incorporate said artifacts into the underlining ALM structure. These components can then be published as is in a standalone manner as individual pieces or they can be incorporated in component libraries, aggregating different components into the same library to be used by citizen developers.

4.3. POWER AUTOMATE

Microsoft Power Automate is the Power Platform component that provides self-service low code cloud automation to a huge variety of apps and services, as well as desktop automation with the built-in RPA capabilities. Power Automate also leverages the same set of hundreds of different data connectors that are made available out-of-the-box in the Power Platform, but it can also use custom connectors to specific services developed by IT to automate targeted use-cases.

Power Automate has three core building blocks that work together to provide an end-to-end authoring experience of workflows that power enterprise-grade automation. A web-based interface (Figure 43) for creating and designing cloud flows, that includes hundreds of different templates for different automation scenarios, the actual cloud flow designer, an approvals center for managing any automations that require an approval from the user's end, and several other functionalities that will be explained in further detail in the next chapters. In addition to this, there is also a desktop application appropriately named Power Automate Desktop (Figure 44), that serves as the authoring tool for desktop flows and encapsulates all the RPA capabilities in Power Automate. Lastly, there is the Power Automate Mobile App (Figure 45) available for Windows, iOS, and Android, which allows its users to access the approvals center, instantiate any of the templates available in the web-based tool as well as providing a simplified designer experience in the mobile device as well, receive any push notifications that might be configured in a given automation, amongst other functionalities.

Figure 43 - Flow Designer in Power Automate web

Figure 44 - Power Automate Desktop

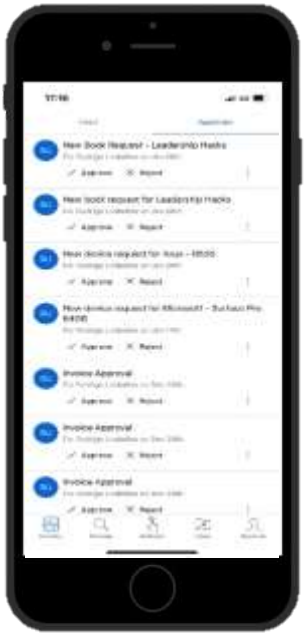

Figure 45 - Power Automate Mobile

There are five kinds of different flows that a user can leverage to automate a specific scenario:

- Automated cloud flows, these flows are triggered by a specific event like when an email from the user's manager is received, when in a new post is submitted in a given blog, when a Common Data Service entity is updated, etc.
- Instant cloud flows, these are flows that are manually triggered as needed to run a specific automation, like getting today's weather forecast for the user's current location, saving a specific Teams message to OneNote, selecting an item in a SharePoint list for manager's approval, etc.
- Scheduled cloud flows, are automations that run when and how often the developer chooses to, these can include copying items from an Oracle's database into a CSV file on a weekly basis, receiving a monthly email summary of new opportunities in the organization's CRM system, etc.
- Desktop flows, these are RPA flows that are developed using Power Automate Desktop and are suited for automating any system or legacy application that does not have a cloud or API point of entry for automation purposes. These can include web scraping scenarios, web page automation and desktop applications since these are developed at the GUI level by analyzing the user's interactions with the interface and keyboard.
- Business process flows, these are flows that are configured at the Common Data Service level, leveraging the relevant entities, and are used to guide users through the different steps of a predefined process. These business process flows are integrated and consumed in the context of model-driven applications.

To completely understand the depth of different automation scenarios covered by Power Automate, it is important to analyze the anatomy of a cloud flow. Cloud flows (no matter what type) are composed of a trigger that initiates the automation and a set of actions to be completed following that trigger event. These triggers and actions can be combined to address simple individual productivity automation scenarios or can be leveraged in mature enterprise-grade automation use cases. These actions and triggers can use any kind of built-in or custom data connectors from the Power Platform to perform operations and initiate automated events in the context of these services, leveraging outputted dynamic content from the data connectors in use, as well as a formula bar for entering expressions that perform operations (functions, operators, variables, etc) in the underlying workflow definition at runtime, using JSON that is validated in its structure by the Workflow Definition Language schema, also present in Azure Logic Apps. In addition to using the data connectors as different triggers and actions for the cloud flow, the user can also make use of inherent controls that are made available in the designer, such as conditions, parallel executing, loops, defining scopes, switch statements, different data operations like joins, defining variables and filtering arrays, and even direct integration with Power Virtual Agents and AI Builder (explained in greater detail in chapters 4.4 and 4.5 respectively).

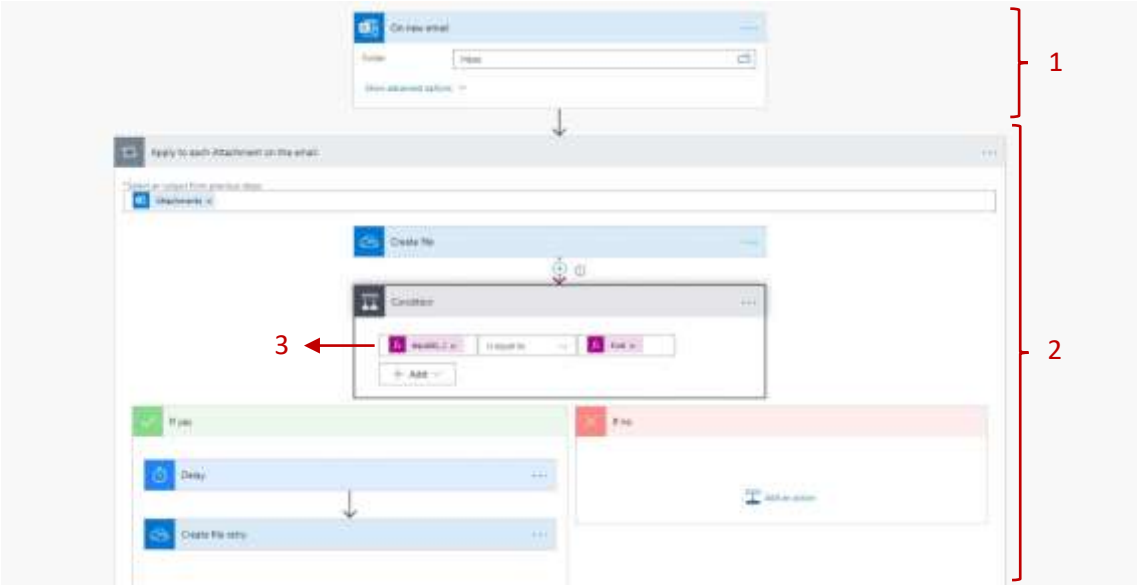

Figure 46 - The anatomy of a cloud flow [1] flow trigger, [2] different flow actions, [3] use of an expression to validate a condition

As with Power Apps, Power Automate also has a Flow Checker that guides the developer into ensuring that the flow's implementation meets the standards when it comes to performance considerations and possible reliability thresholds. It is always active, and it is automatically ran when a user saves a flow or it can also be ran in an ad hoc manner, alerting if the flow contains any warnings or errors in its structure. Power Automate also has a testing feature for flows, allowing the developer to manually trigger the automation, which is a functionality that proves to be extremely useful in debugging scenarios and when the developers need to make sure that the automation is working as expected before pushing the flow into the production phase.

There is native integration between the cloud flows from the web designer and the desktop flows from the Power Automate Desktop application. Although Power Automate Desktop is a tool that allows for very advanced RPA development, it burrows the same low code strategy from its predecessor by featuring a very similar drag and drop UI that the users are accustomed to with the web designer. There are hundreds of different actions that developers can use to automate using RPA: desktop/Windows applications, web applications, terminal emulators, tasks in Amazon Web Services, manage resources in the Azure cloud like resource groups and virtual machines, just to name a few. It basically provides endless automation scenarios since a user is only limited to the boundaries of his/her own GUI. Although the developer can manually drag and drop the different actions in Power Automate Desktop, configuring each step with the necessary characteristics that are demanded by that automation, the tool also features a web recorder and a desktop recorder. This means that whether the scenario calls for a web automation, a desktop automation, or even a mixture of both, developers can always leverage these recorders to record themselves doing the different steps in the workflow in a manual manner. The tool then analyzes the different components with which the user interacts, collecting inputs and outputs and keeping track of the different UI changes that derive from said interactions, and automatically bundles them together into a series of actions in Power Automate Desktop, which then are usually tweaked and optimized to convey the optimal automation for that scenario. Once a desktop flow is created and published, it also shows up on the web version of Power Automate. With this in mind, developers can blend together the two technologies to seamlessly incorporate cloud automation and robotic process automation, unlocking use cases that call for both technologies (for example, a cloud flow that triggers when a new invoice is submitted into the organization's expense platform, utilizes AI and OCR capabilities to extract the data from the PDF file that contains the invoice, submits said invoice data directly to the manager of the individual who initially uploaded the document through a push notification for approval purposes, and, if approved, it triggers a desktop flow that takes the approved invoice's data and submits it into the company's legacy invoice application).

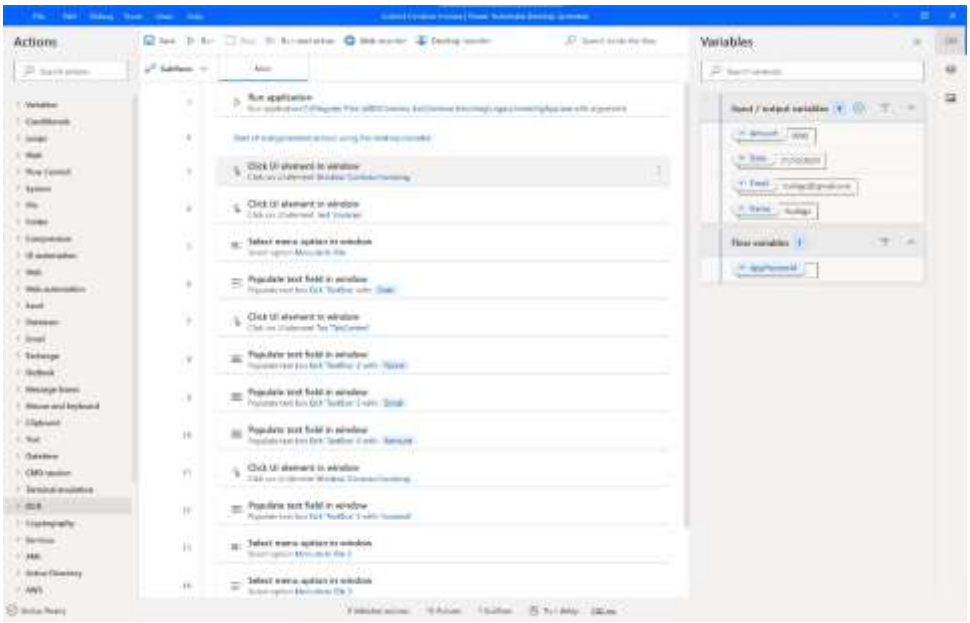

Figure 47 - A desktop flow in Power Automate Desktop

There are two kinds of running modes for desktop flows: attended automation and unattended automation. Attended automation is more suitable for productivity scenarios, since the target machine in which the desktop flow will run must have an authenticated active Windows user session. Because it is supervised RPA, the users must not interact with the device while the desktop flow is running, because doing so could cause to break the interaction between the desktop flow and the GUI. Whilst for unattended automation, it is best in context the of automations which do not require human supervision, since Power Automate automatically signs in on the target device and once the automation is completed, it simply signs out and reports its activity. This is a core feature of desktop flows from Power Automate, since arguably the majority of RPA use cases do not require human supervision and there are several drawbacks of having the processes running in attended mode when it comes to automation scenarios at the enterprise level. Although RPA is a very powerful technology, cloud automation should always be prioritized when it comes to implementing a workflow architecture. This is due to the fact that cloud automation via APIs is far more reliable and consistent than RPA, because there are less changes to the infrastructure of online services, while in RPA the workflow is entirely dependent on the UI of the several applications that the users interact with, which are usually very prone to alterations and changes in behavior.

4.4. POWER VIRTUAL AGENTS

Microsoft Power Virtual Agents is, at the time or writing, the most recent member of the Power Platform family. Much like many of the other components, Power Virtual Agents provides web-based, low code graphical interface for the development and deployment of chatbots. Traditional bot development solely relies on development teams with deep knowledge in NLP techniques and all the other prerequisites that conversational AI brings to the table, with all its complexities and code. Since Power Virtual Agents follows the same low code approach as the remaining platform, it decreased the gap between subject matter experts and the development teams, relieving the IT department of complex development and vastly improving the time to market of chatbot solutions, allowing human agents to prioritize and focus on the more complex issues. In addition to this, Power Virtual Agents provide very mature and advanced out of the box conversational AI capabilities because it is built upon Microsoft LUIS and the Azure Bot Framework, leveraging these two very popular and renown conversational intelligence products in a user-friendly SaaS offering.

Power Virtual Agents consists of a single building block, a web-based user interface, that encapsulates all the features that users need to create, test, and publish the bot to a variety of different channels. The main features of the Power Virtual Agents interface include topic configuration (where users can access and customize built-in system topics and custom user topics for their bot), entities (where users can access and customize prebuilt system entities and custom entities that used by the bot), out of the box analytics (bot performance, resolution rate of engagements, customer satisfactions, amongst several others), the publishing page (where users can publish a production-ready bot to a variety of different channels to be consumed by end users), the settings and management panels (allow to configure to which channels the bot is deployed to, configure and manage bot skills and authentication methods, configuring fallback

topics and define transfer to a live agent settings) and lastly a test bot panel (in which bot authors can test the bot topics in real-time as it is being developed to assure that everything is behaving as expected).

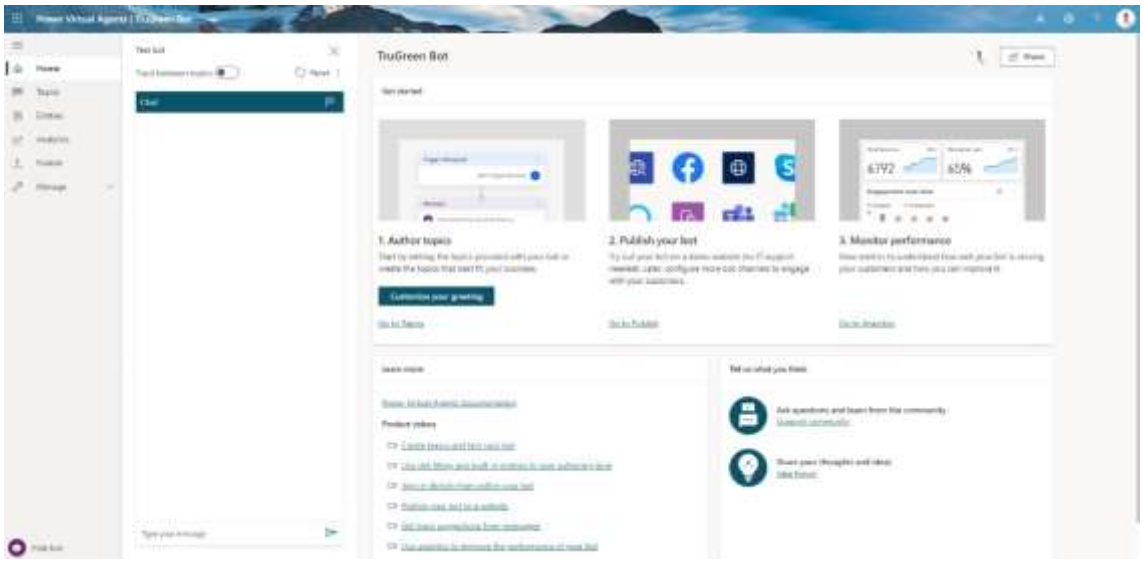

Figure 48 - Power Virtual Agents web interface

Topics are the foundational pieces of the chatbot structure. They basically represent conversation pieces defined by content that the end user will be able to have with the bot, and are made up of two components, trigger phrases and conversation nodes. Trigger phrases are keywords, sentences and questions that the bot developer configures *apriori* of bot deployment that will function has the trigger for that particular topic when an end user uses natural language that relates to that specific matter, this way the bot will automatically learn when to respond with that specific dialog, meaning that the more trigger phrases a topic has, the more likely will be the efficacy of the bot in using the topic. In most use cases, topics will be custom built to cater to the requirements of the use case, but there are several built-in topics that are widely used in bot development, such as greetings, goodbyes, escalation topics, confirming success/failure, switching the conversation to a different topic, or simply ending the conversation. Conversation nodes are the steps that the bot will follow after a given topic is fired by the trigger phrases. There are several types of conversation nodes that the developer can leverage to shape the interaction between the end user and the chatbot: asking a question, in which the developer can provide multiple choice options to the end user or by asking an open-ended question (whatever option is used, the developer can configure which information needs to be identified from the user's end and store it in variables to be used in the remaining context of the conversation – the developer can store one of the multiple choices that the user picks, can store the user's entire natural language response, or even use prebuilt or custom entities, which are basically information units that represent real-world artifacts, that can filter the user's response in extracting fragments like a zip code, an order number, an email and so on); adding a condition based on the comparison of specific variables to add a partition to the conversation, shaping the structure of the conversation and defining several outcomes based on user's interaction; calling an action, and this conversation node includes both the ability of authenticating the current user using one of the several OAuth mechanisms (a very often used

feature of LoB chatbots for accessing employee-owned organizational data) or calling a Power Automate flow (this feature alone broadens the horizons to limitless integration and automation scenarios, since developers can now leverage the full stack of Power Automate's technology to pass context-aware variables from the conversation directly to the flow, and returning the flow's outputs back into the conversation with the end user, if needed); simply showing a message without requiring extra input by the end user; redirecting the end user to another existent topic, shifting the interaction to a different conversation subject; and finally, the last conversation node is simply ending the conversation, which can be done by prompting an automatic survey to the end user to access customer satisfaction or redirecting the end user to a live human agent (which can access the complete transcript and customer interaction with the chatbot to gain full context of the whole situation and resume the conversation) if the inquiries were not satisfied through the chatbot interaction (this can be done either by integrating with Dynamics 365 Omnichannel for Customer Service or through an engagement hub of the developer's choice).

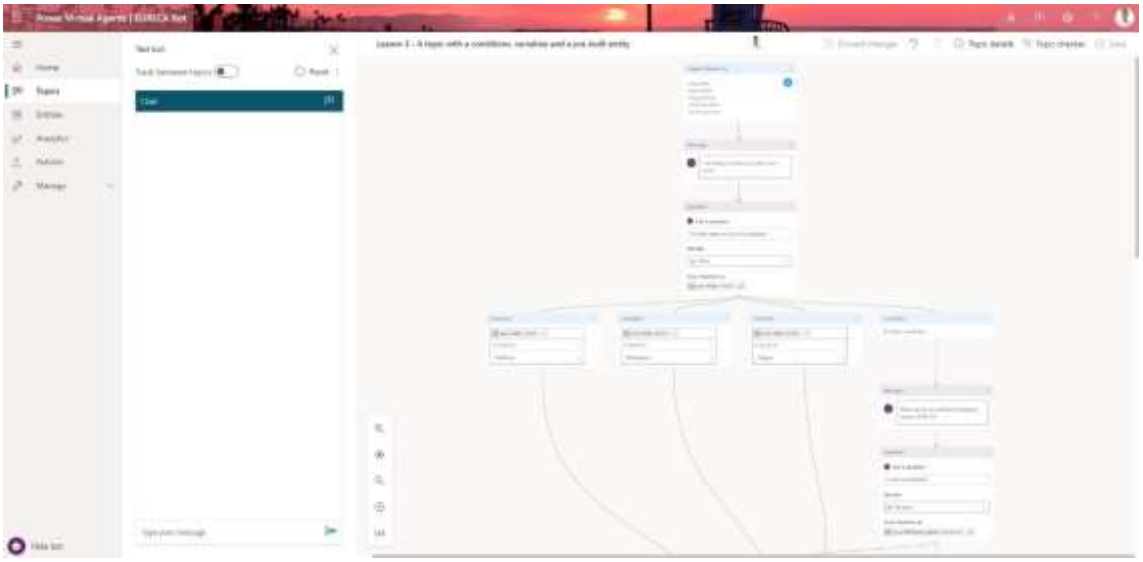

Figure 49 - Power Virtual Agents authoring canvas for an example topic

As the developer keeps adding conversation nodes into a given topic, the authoring canvas displays the full decision tree in a very user-friendly format. This helps the developer in understanding the flow of the conversation that the end user will establish with the bot, as well as immediate testing in the test bot panel on the left side of the interface. Whenever a change is done to a topic or to a specific action or integration, the developer can immediately test it in real time before shipping the update into production. Once the bot is developed and tested, the developer can choose from a wide array of different channels to publish the bot. These channels include Microsoft Teams (focused on LoB chatbots), custom enterprise websites or mobile applications through iframe embedding, and other popular online services like Facebook Messenger, Cortana, Slack or Twilio.

Much like all the other technologies in the Power Platform, Power Virtual Agents provide pro-code extensibility to pro developers, in this case by making use of the Azure Bot Framework. This is achieved either by extending the virtual agent with the Bot Framework Composer directly, or by reusing any possible bots that were developed in the organization using the Bot Framework and converting said bots into skills that can be embedded in Power Virtual Agents.

When it comes to the Bot Framework Composer, it provides unlimited functionality that might not be totally present in the low code environment of Power Virtual Agents, including the development of functionalities using regular expressions, adaptive cards and dialogs, and much more.

4.5. AI BUILDER

Having described the four Power Platform components, it is necessary to go in more detail into the capabilities that surround the platform that allow developers to add extra functionality with more advanced features in the organization's business solutions. AI Builder democratizes artificial intelligence by providing people who might not have a specific data science or machine learning background with the ability of adding AI capabilities into their Power Apps and Power Automate flows, through a low code interface. This opens the door to a variety of business performance improvements, as well as process automation, by unifying Power Platform artifacts and the underlying data platform (being it the Common Data Service or any data source that the organization makes use of) with predictive intelligence from AI Builder to enhance business activities.

Anyone can use AI Builder to create a variety of different AI models, by accessing it directly in the AI Builder tab from the Power Apps or Power Automate web home page. In both interfaces, developers can create and customize models or use prebuilt ones, as well as accessing previously created models to manage them directly, being it updating them to meet new business requirements, share them with end users for consumption purposes or deleting them if necessary. There are two main types of AI Builder models, custom models that developers train with their business production data to tailor the AI functionality for their specific enterprise needs, or they can also use prebuilt AI Builder models that are ready to use for a vast array of common business scenarios. All the models (even the custom ones) follow a wizard-like creation experience, to enable power users to create and deploy what would otherwise be very complex AI and ML models directly into their business solutions.

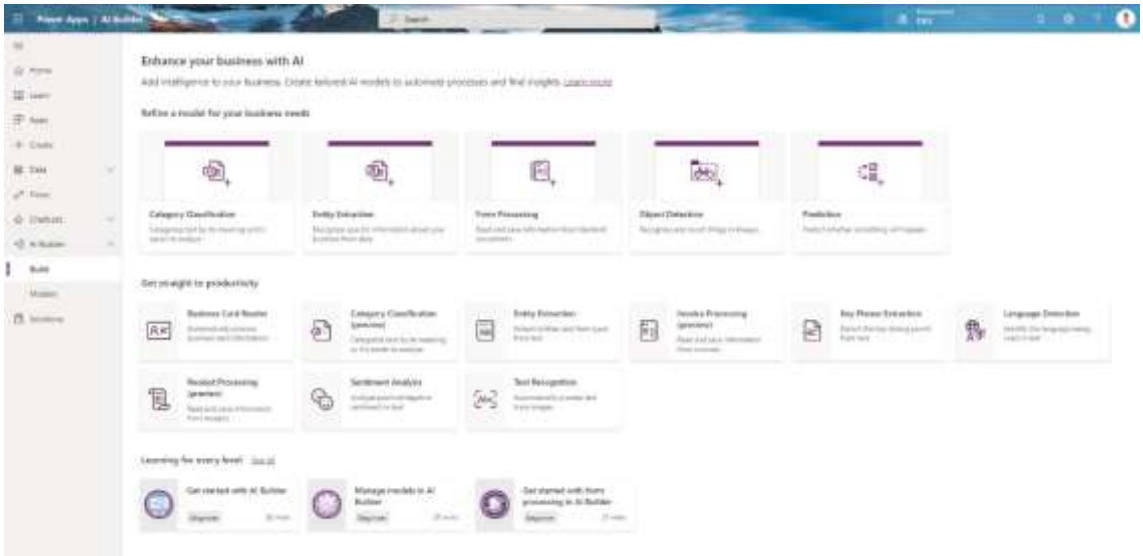

Figure 50 - The AI Builder tab in Power Apps Home Page

Starting with the custom models, there are five different ones (at the time of writing) that developers can use to further customize and tailor them to their business needs:

- Category Classification – this model uses AI to automatically classify text entries and extract intelligence from the organization's textual data. Tagging textual data is particularly useful in spam detection scenarios, customer request routing, just to name a few. Developers can use training data stored in a Common Data Service entity (the textual data and the respective tags) to predict future outcomes.
- Entity Extraction – this model can recognize specific data in targeted text based on specific business needs, by identifying key elements and classifying them into predefined categories. This means that this model can be used to transform unstructured data into structured data by identifying the different entities/data types and further retrieve information and extract facts. This model is available as a custom one (for users that need to use their specific training data and parameters that need to cater to unique business requirements that call for a purpose-built model) and as a prebuilt one that does not require training or management (more suitable for the variety of use-cases where there is not a need for customization).
- Form Processing – this model provides OCR features that can automate the reading and extraction process of data from any standard documents, like invoices, tax documents, supply-chain information, and much more. Developers can directly define the information that they wish to extract from the different documents, with the ability to create different collections inside the same model, each collection holding documents with the same layout, giving the ability to process documents with different layouts in the same model. Users can then teach the model to identify and extract the information by tagging and labeling the relevant fields, and process new documents automatically once the model is trained. This model can be applied to PDFs, image files or by taking a picture of a paper-based document directly using a Power App.

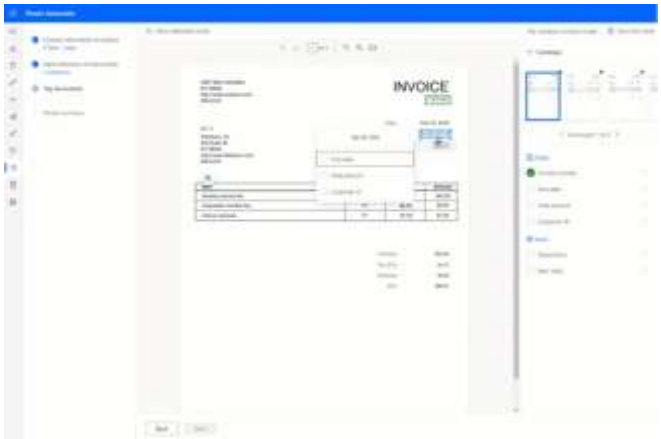

Figure 51 - Form Processing Model build wizard in AI Builder

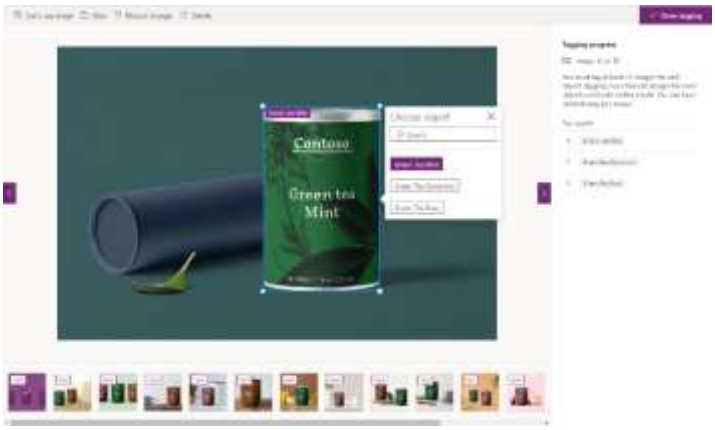

Figure 52 - Object Detection Model build wizard in AI Builder

- Object Detection – this model is able to automate business processes by identifying and counting objects in any image in real time. There is a variety of use cases for such functionality, such as streamlining inventory management in retail, identifying low supplies, flagging unexpected items, speeding information access to frontline workers,

just to name a few. Developers simply need to define the classes that need to be identified and upload images of each object class for training. After the upload, users simply need to tag and label the different object classes in each image and then train the model to predict further object identification based on new image recognition.

- Prediction – this last custom model unlocks common ML problems using a low code approach. The AI Builder Prediction model analyzes historical data to find patterns amidst data points to associate them with outcomes. After the model is trained, the model takes those patterns and applies them to a new set of data to predict future outcomes and aid in the decision-making process. The prediction model can support business questions for binary predictions, multi-class predictions and numerical predictions. This model can cover as many different scenarios as the organizations need to if they have the right data and have a concrete business requirement defined – ranging from automatic fraud detection, customer churn analysis, client targeting for marketing campaigns and so on.

Besides all these custom models that can be tailored to meet business needs, there are nine more prebuilt AI models that do not need to be trained or customized, and can address a huge variety of the most common use-cases:

- Business Card Reader – a prebuilt AI model that automatically processes business card information. It automatically scraps business details such as the person's name, job title, address, email, company, and phone numbers when the model detects a business card in the image. These can be image files or via a photo taken directly using a Power App.
- Category Classification – this is a prebuilt version of the custom Category Classification model. This particular version caters to customer feedback scenarios of classifying text categories, which is trained to identify the following categories form text data: issues, compliments, customer service, documentation, price & billing and staff.
- Entity Extraction – this is a prebuilt version of the custom Entity Extraction model. This model identifies the most common element from text and allocates them into predefined categories, addressing most common use cases. The model can identify entities such as age, language, duration, percentages, URLs, and dozens more.
- Invoice Processing – this is a prebuilt AI model that leverages OCR features for extracting key invoice data and ids in the automation process of organizations' invoicing system. This Invoice Processing Model is optimized for recognizing and extracting common invoice elements like the invoice ID, invoice date, amount due, items list and more.
- Key Phrase Extraction – this prebuilt model uses AI to identify and extract the main points in a text document. This is particularly useful in scenarios where there is a need for automatic summarization of textual data, for example when for automatically identifying the list of keywords from the synopsis of a piece of content. It is also able to identify and return a list of key phrases from unstructured text documents.
- Language Detection – this prebuilt model identifies the predominant language from textual data. The model analyzes all the textual data in the document and returns the detected language code as well as a confidence score ranging from 0 to 1. This is useful in a multitude of different scenarios, like for example automatically redirecting emails to specific inboxes based on the language used in the body of an email.

- Receipt Processing – this is a prebuilt model optimized for receipts. It uses OCR capabilities to detect printed and hand-written text and extract said data from receipts. Once a receipt is detected, it automatically scraps the data and outputs common receipt elements like the merchant name, the address, transaction dates, subtotal, tax, the tip value and more.
- Text Recognition – this is another prebuilt model that uses OCR features to extract words from documents and images, being it printed or hand-written text. This can cater to a variety of different use-cases, like automatically extracting the string value of license plates from vehicles photographs, extracting badge information from personnel pictures and more.
- Sentiment Analysis – this prebuilt model detects positive or negative sentiment in textual data. This is extremely useful in several use-cases, for example analyzing customer reviews in the company's website, doing sentiment analysis in tweets with a particular hashtag, or any kind of text data. The model outputs the sentiment (which can be positive, negative, neutral or mixed), the confidence level as well as individual analysis for each sentence in the document and an overall document score.
- Text Translation – this last prebuilt model is able to translate textual data in real time across more than 60 different languages. This is extremely useful for any scenario where a language barrier exists within a company.

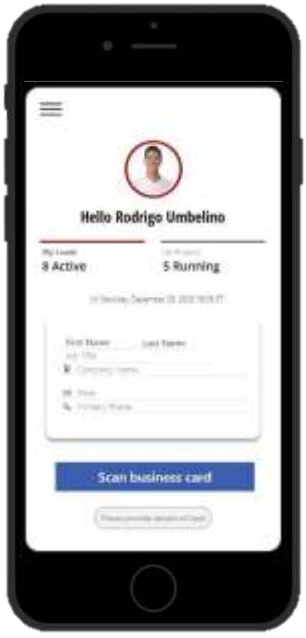

Figure 54 - Business Card Reader model embedded in a Canvas Power App

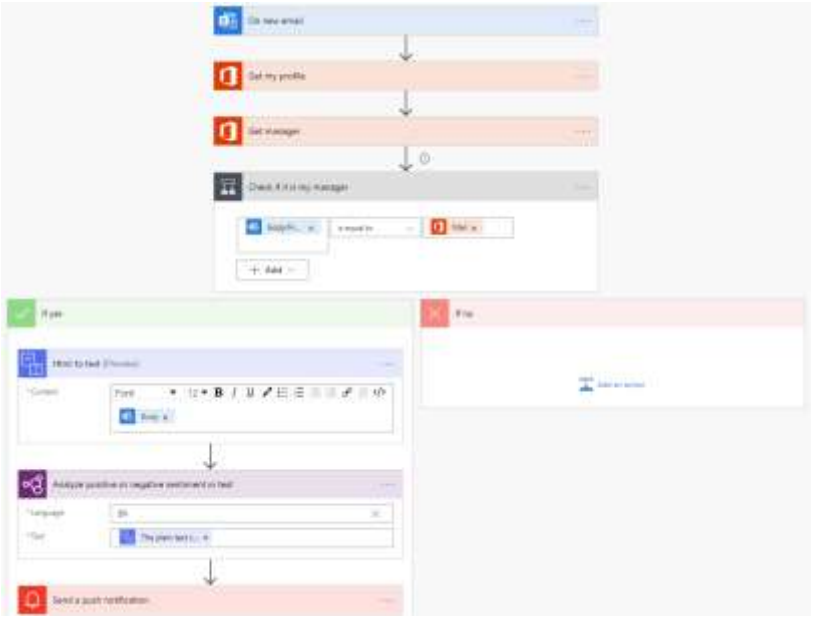

Figure 53 - Sentiment Analysis Model embedded in a Power Automate flow that automatically notifies the user with the sentiment of a manager's email

For every type of AI Builder model (being it prebuilt or custom), it is possible to use them both in Power Apps and in Power Automate flows (and consequently Power Virtual Agents since the chatbots can leverage flows as actions). Once a custom model is trained, AI Builder automatically calculates a performance score ranging from 0 to 100, based on the precision and recall metrics

from the model, as well as the F1 score. When a model meets the developer's and the company's standards, it can be published and shared with end-users that need to consume it.

4.6. COMMON DATA SERVICE

The Common Data Service is the cloud-based underlying low code data platform that powers several Microsoft Dynamics 365 solutions as well as the various Power Platform components. This direct integration with the Power Platform makes it easy to build scalable and impactful business solutions. The CDS is available globally but it is deployed at a geographical level to make sure that all the data storage procedures are done in a compliant manner. This technology aims to be the central business data storage system of any organization, since its low code interface enables a very easy structure definition and management of all the enterprise's business logic and data diversity, seamlessly supporting a huge variety of connected services, applications and processes with the appropriate security model implemented.

Similarly to AI Builder, the Common Data Service is accessed and configured entirely though the browser, in either the Power Apps or Power Automate web page. It is through that interface that users can create instances of the CDS, which basically comes as a low code database that contains several predefined tables and fields for holding business data that are commonly found in the majority of organizations. If the prebuilt tables (named entities in CDS) and fields are not suitable for a particular business solution, developers can always extend the out of the box functionalities and add custom fields or entities. This takes the burden of having to manage the infrastructure aspect of any cloud database, since the CDS is deployed in a point-and-click fashion (automatically provisioning the underlying standardized data model) and allowing developers to focus on the task at hands and develop the actual business solution.

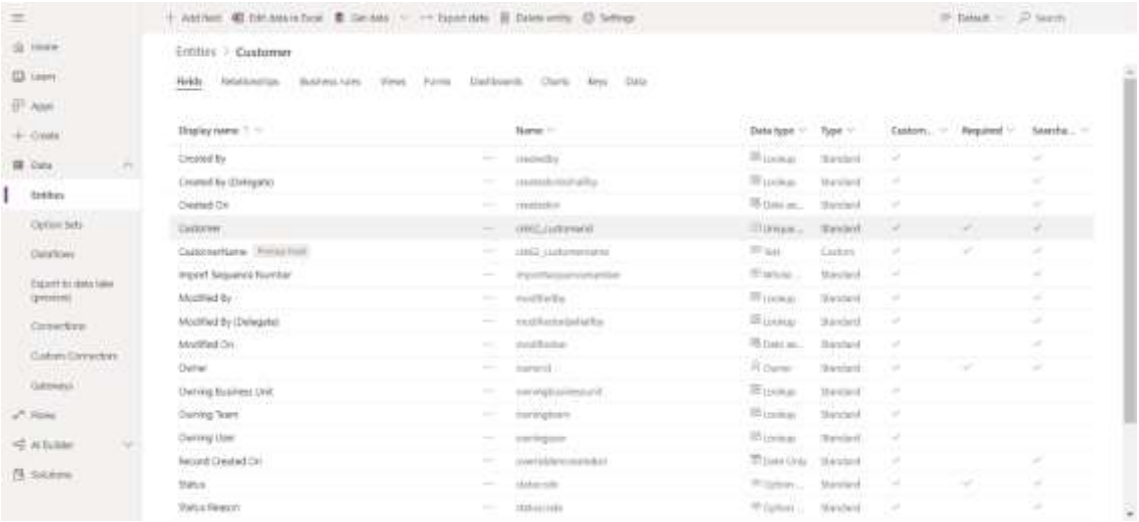

Figure 55 - The Entities tab of Common Data Service in Power Apps Home Page

One important thing to consider, is that the Common Data Service is far more than a database. As the self-service data platform that is it, it bundles together a great deal of functionality across several areas: the CDS API, security features, business logic, data definition, a PaaS storage service and integration capabilities. When it comes to security, the CDS allows for very granular

security role configurations (supporting authorization scenarios even at the field level with role-based security as well as advanced auditing capabilities) and handles its authentication process directly with Azure AD, to make use of features like multi-factor authentication and conditional access policies. At the business logic level, the CDS can apply custom logic directly at the data layer (using business rules, duplicate detection mechanisms, workflows, just to name a few) making sure that these rules that maintain business integrity are reinforced for every CDS user, no matter what solution they are using to interact with said business data.

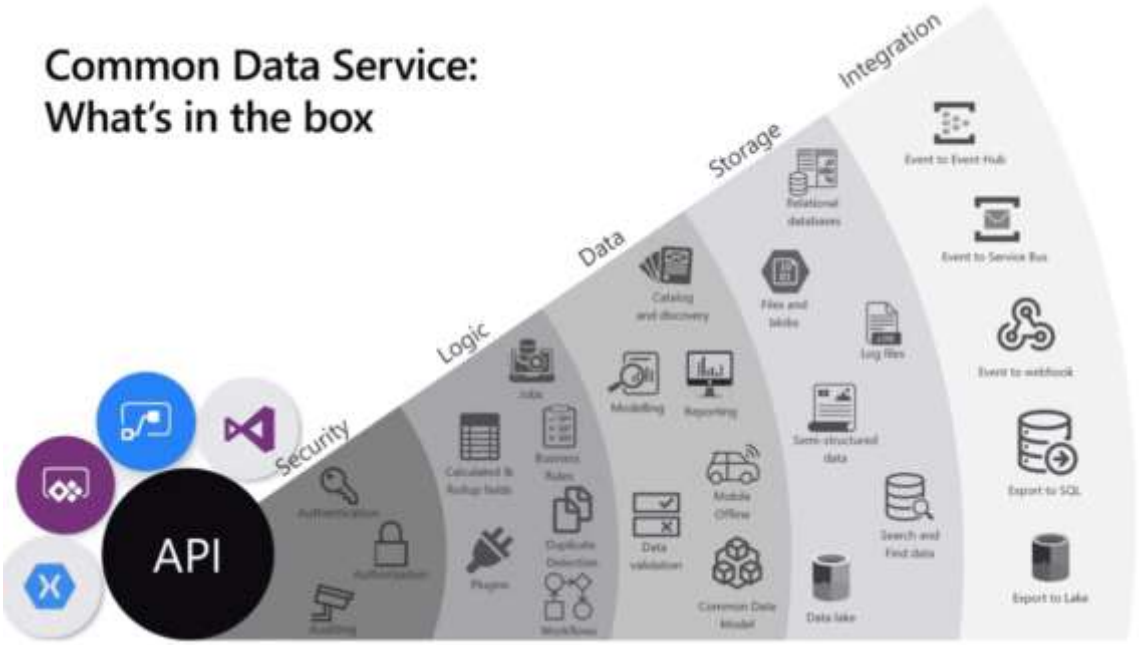

Figure 56 - Unpacking the full Common Data Service functionality

In the data side of things, the CDS provides native integration with dataflows technology as well as Power Query for robust low-code ETL – allowing database administrators to control the shape of their business data, as well as unlocking limitless validation, shaping, modelling, and reporting capabilities. Regarding the storage aspect of CDS, concerns about managing and scaling infrastructure are stripped away since the business physical data is stored directly in the Azure cloud. Lastly, CDS connectivity capabilities allow for powerful integration use cases, making use of webhooks, data exports to SQL or a data lake, APIs and eventing tools, making easy to get data in and out of CDS.

In CDS, entities are standard or custom data structures comprised by a set of records and are used to store data. They can be compared to tables in a relational database, although they differ in some respects. The records within each entity are comprised of different fields that hold and characterize individual pieces of data related to a single record. Every time a CDS database is instantiated, it will provision the same set of standard entities (which are the most commonly used entities in business solutions across organizations, data structures like Accounts, Contacts, Activities, etc), but the infrastructure offers developers the flexibility of extending and customizing the database to meet targeted business requirements. The layout of these standard entities is based directly on a set of standards from an open data model named the Common Data Model. This open model is a schema that manifests a set of open-sourced, standardized,

and flexible data entities, attributes, relationships, and semantic data that are the result of an initiative between Microsoft and several other organizations called the Open Data Initiative. The Open Data Initiative is a joint effort between Microsoft, Adobe, and SAP to create a solidified schema that provides a single view over enterprise data, allowing organizations to bring together all their data from all departments, systems, applications, and services using the guidelines of this standardized Common Data Model. Because CDS uses the Common Data Model schema in its infrastructure, it means that any integration with other solutions that also use this model are very simple to do because the standard entities between systems are the same.

Having gone through entities, fields, and records (which can be directly compared to tables, columns and rows using more mainstream nomenclature), there are other CDS artifacts that contribute with added functionality to the whole data platform. It would be extremely inefficient to bundle up all the information into a single entity, that is why developers break down the different silos of data into the most appropriate entities, and then use relationships to establish relational connections between entities. Like any traditional database system, the most used relationship types are available (many-to-one, one-to-many and many-to-many) in conjunction with any primary or foreign key definition. This allows developers to maintain a data infrastructure that is efficient and scalable to support their business solutions, and respective integrations. In addition to this, it is possible to leverage CDS business rules, rollup, and calculation fields to define and apply business logic at the data level instead of the service level. By transferring the business logic to the data level instead of the application level, organizations can have a tight control on their data and can create a system where their logic is agnostic to any kind of application being used (being it Power Apps, Power Automate, any Dynamics 365 solution or even via the API) since it is tied directly to their data. The following differences between the various business logic artefacts are as follows:

- Business Rules – there are a powerful engine for enforcing enterprise data rules that define server-side logic. These can automatically set or clear values from entities, validate stored data regardless of the way it was inputted, show error messages, and more. They are a very effective solution for simplifying application development, as well as increasing data accuracy and streamlining solutions.
- Rollup Fields – this functionality performs calculations on field values that are stored in related entities. These include calculating sums, counts, minimums and maximums in an asynchronous way.
- Calculation Fields – this feature allows developers to define calculation formulas that are run synchronously in real time as new data is saved to the CDS instance. It is an extremely useful capability in improving the integrity of business data and it is also agnostic to the input method used to submit data into CDS.

It is also through the Common Data Service that many of the management, administration and ALM capabilities of the Power Platform are unlocked. Power Platform environments can be seen as containers for storing and managing the different Power Platform resources (apps, flows, chatbots, custom entities, data, and much more). Environments are created under an Azure AD tenant, with its resources restricted in use to the users under the tenant's scope. In addition to the AD itself, each environment is also bound to a geographical location, and each environment

allows for the provision of one instance of the CDS. This means that the CDS instance, the business data and all the remaining Power Platform resources are bound to that geographical location as well. IT departments can make use of environments to target specific placeholders for their business solutions: environments targeted to a specific geographical location, development-only environments, test-only environments, and production environments, just to name a few.

In conjunction with environments, Power Platform solutions are mechanisms that can be used for implementing ALM across the several Power Platform components, mainly Power Apps and Power Automate. They can be seen as placeholders that encapsulate the different components (these can be apps, flows, chatbots, entities, AI Builder models, and more) that address a specific business scenario. There are two types of solutions:

- Unmanaged Solutions – these are solutions that are used while the development team is still building the solution, usually in a dedicated development environment. These can be seen as the source for the different Power Platform elements that make up that solution, and are used in the design, development and testing phases of the overall solution. Unmanaged solutions unlock the underlying components for customization, allowing developers to work on the backbone of the different assets.
- Managed Solutions – these are solutions that are used for testing or consumption-only purposes, meaning that these are deployed in dedicated testing or production environments. Managed solutions cannot be modified and should be created by exporting unmanaged solutions as managed ones, creating a dependency between the actual build artifact and the consumption artifact.

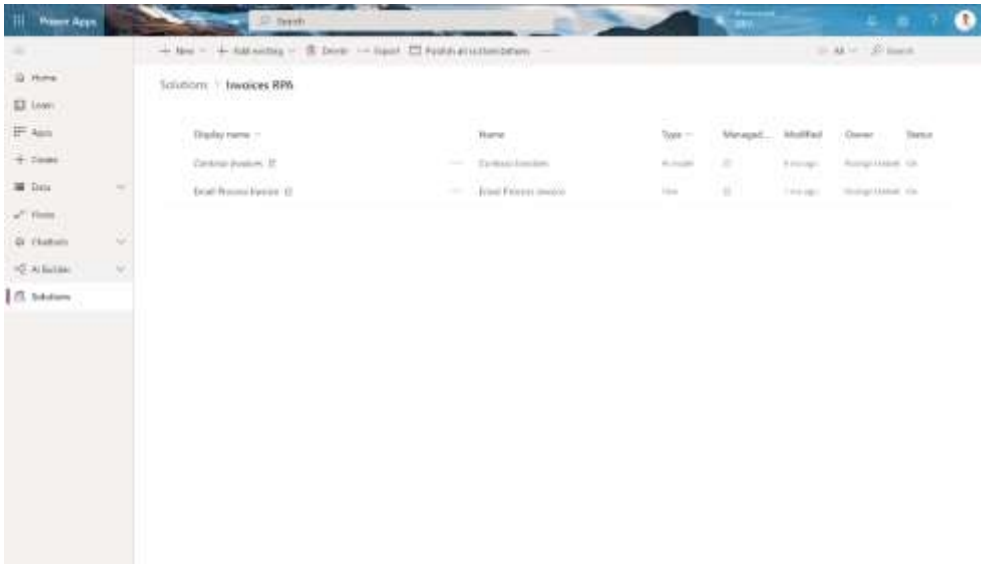

Figure 57 - The Solutions tab in Power Apps Home Page

This in conjunction with Power Platform environments makes it easy to update and move solutions from DEV/TEST/PROD by leveraging the dependencies of the different types of solutions. However, in enterprise scenarios, these steps should be embedded in the overall company's ALM strategy, meaning that a fair amount of these steps need to be automated. Whenever there is a case for a more sophisticated ALM implementation like this one, IT teams

can leverage Azure DevOps to automate all of CI/CD tasks, by using the Microsoft Power Platform Build Tools. This provides a way of automating a vast array of activities like synchronization between solutions, environment provisioning, deploying components to downstream environments, performing automatic unit testing and solution checkers, and more.

5. PROJECTS AND DEMONSTRATIONS

In this section, the different activities performed by the Technical Specialist will be described. These will not detail activities regarding implementation projects since these are out of scope when it comes to the duties of a pre-sale's technical specialist. Instead, this section will provide an overview of the activities that are part of the responsibilities of the TS, including the different trainings and different interactions with customers (PoCs and demonstrations). Licensing conversations as well as sessions regarding Power Platform governance and administration will be out of scope in this report.

5.1. TIMELINE

Since the fiscal year of 2020 was the first year that the TS was in this full-time position, it is expected to be a ramp-up year. Therefore, the first two weeks starting in September 2019 were purely dedicated to the onboarding of the employee. During these weeks there were daily sessions explaining to the new employees the several areas that are important for every Microsoft employee. These include the Microsoft organizational structure, the different segments and teams, a tour of the two offices in Lisbon and how they are structured. In addition to this, in these first two weeks, there were several sessions where seasoned Microsoft employees would present and explain their role and their vision of what it is like to work at Microsoft. After these two weeks, there was a dedicated training and up-skilling period of roughly two months. Although the role of the TS has a core responsibility of maintaining deep technical knowledge of the technologies that he/she is responsible for throughout the career, these two months were crucial for a deep dive in some of the Power Platform components that were a bit out of scope regarding the Master's degree. Despite not having worked with these tools during the first year of the Master's, the concepts and all the theory that was assimilated during that period was paramount for a very quick ramp-up in components like Power BI, AI Builder and Power Virtual Agents to some extent. What needed more focus from the TS in this initial phase were the components which tackled areas of less expertise by the TS, mainly Power Apps and Power Automate.

During these two months the training was done in several different ways and split into two distinct areas (commercial training and technical training): there are plenty of ramp-up materials provided by Microsoft for online technical training, that lead to the ability of taking exams for unlocking Microsoft Certifications. The most effective learning commercially speaking is on-the-job learning, which was done through participating in team meetings from a very early stage, as well as shadowing sessions with the TS mentor and manager. From the technical perspective, there was also on-site training for all the Western Europe Power Platform Technical Specialists, that happening in Munich, Germany for a period of two weeks.

In addition to the theoretical ramp-up, there was the need for some upskilling in the hands-on experience of some of the components. Power BI and AI Builder were not a concern because of their inherent connection with the master's degree, but some practical knowledge of working with tools like Power Apps and Power Automate was needed. In order to solve this matter, the TS in conjunction with the manager met and discussed with other Microsoft Portugal employees

to come up with possible internal use-cases that could be solved using the Power Platform components. Two came to mind, the first use-case was a management solution for the Microsoft Portugal Marketing team, that allowed the team to manage all the events at Microsoft Portugal using an applicational interface, and that same interface could be consumed by any employee to check the events that were happening via calendar, and even add them to their personal work Outlook schedule if they were interested in attending. The second use case came from the Finance team, following the Wellness initiative of providing thirty-minute massages to Microsoft employees in the office. They required a solution that allowed employees to check slot availability, book appointments and cancel existing ones. In addition to this, the solution should automatically send at the beginning of each week a list of all the time slots and location to the masseuse.

Figure 58 - Event management solution for the Marketing team

Figure 59 - Wellness scheduler solution for the Finance team

These in-house projects were key for the TS to gather the necessary hands-on knowledge in components that were not particularly connected to the contents taught during the master's degree. After the ramp-up period, the TS was integrated into the sales motion in alignment with account executives and solution area specialists, using the technical and commercial aspects of the Power Platform to deliver value to customers. These sales motions often included preparing and delivering compelling technical demonstrations of the Power Platform, either by using Microsoft internal demo resources or by custom building demonstrations to cater to a specific customer use case/scenario or to prove the technology using a PoC in a strategic customer. Since there were dozens of these engagements with every single account that was allocated to the TS, the scope of this thesis will cover two solutions that were planned, built, and presented by the TS to customers during the fiscal year of 2019, that used most of the components of the Power Platform integrated with the various services from the Microsoft cloud, with the objective of proving the technical value of the platform.

5.2. Mobile AI Product Recognition Solution

This first demonstration/PoC was prepared for a Portuguese customer in the Consumer Goods industry. The main objective of this sales motion was to prove the low time-to-market capabilities of a low-code platform like the Power Platform, and the opportunity was brought by the account executive after the customer feedback that was looking to invest in an advanced low code / pro code technology. The demonstration should focus on a solution for field service technicians from the Consumer Goods company to quickly inspect and inventory their products in large retail stores.

With this objective in mind, an approach was outlined taking inspiration from a Power Platform success story with G&J Pepsi, where they were facing similar challenges as this Portuguese customer. Therefore, a Power Platform solution was developed to illustrate how field workers could perform the regular inspection in a more efficient and secure manner using Power Apps embedded with AI capabilities, in alternative to the current paper-based format.

5.2.1. Solution Overview

To prove the low time-to-market possibilities that a low code platform can provide, the inventory inspection process of this customer was illustrated using the Power Platform. Instead of paper-based forms, field technicians would only need a Power App. In that mobile application, the worker can always check where is the next retail store that he/she must go to perform the next inspection. Arriving at the location, a blueprint of the store is displayed, with the aisle where the company's products are stored for sale, and where the inspection takes place.

Upon arriving at the exact product location in the retail store, the technician can inspect the overall condition of the facilities, including product placement, location system, report any malfunction if there are any cooling systems in the store, etc. This is done via the task menu. In the auditing tab itself, there is an AI model embedded in the application, so that the worker can very easily take a picture of the products in the aisle or in the cooler, and the AI model will recognize the trained objects and provide a count of each item. This technology can be used for inventory use cases or explored in competitor analysis scenarios.

Figure 60 - Welcome screen, map screen and task screen of the demo, respectively

After the AI model recognizes the products, the fields are automatically populated with the data, and the technician can finish submitting the report. Leveraging the full integration with Microsoft services, the mobile solution also provides a feedback tab that the technician can use to collect feedback from the retail store manager, with a Microsoft Forms questionnaire embedded in the application itself.

Figure 61 - Microsoft Forms survey screen, AI object recognition screen and summary screen of the demo, respectively

After the auditing process is finished, the technician can submit all the collected data, and the application will give a summary of every single data point that was collected, as well as redirect the technician to his/her next location for inspection in the itinerary.

The architecture of the solution is composed by five core components: the Common Data Service, with custom entities, relationships and business rules that hold and store the data model that is handled by the mobile interface; Power Apps, which was the tool that was used for designing the mobile interface, leveraging elements such as text boxes, forms, galleries and reusable components (headers, footers, displays) for multi-use purposes; Microsoft Forms, for creating a questionnaire available to external users (in this case, the retail store manager) that is embedded in the mobile interface; Data connectors, these were used to connect the Power App to online services, including the CDS connector (to retrieve data from the service for end-user display, and to patch data back to the cloud), the Office 365 Users connector (to access logged in user data in the Power App including logged in user name, email, profile picture, and more), and the Bing Maps connector for computing the next store for inspection based on the field technician route as well as the distance until the next stop; the last component is the AI Builder, that allowed to configure the Object Detection AI model that is embedded in the mobile application for field technician use (the details regarding said model are described in the next subchapters).

5.2.2. Data Description

For this PoC, there was no image data available whatsoever to be used in the context of this project, which means that it had to be collected. Because this is a pre-sales scenario, the PoC did not need to address all the items that were in the catalog of the Consumer Goods company. Instead, three classes of different products were selected *apriori* to be the focus of the data collection process.

Because these products are commercially available in a retail format in plenty establishments in Portugal, this meant that there was an opportunity of manually collecting the image data of the three different classes of products from these establishments. Using a mobile device, the TS took pictures of the referred three different products in different contexts: in store shelves, with different lighting and backgrounds, amongst different products that were not the target of the model, in different quantities and angles, etc. By taking pictures in different circumstances, the reliability of the model can be improved in a production scenario, where more often the end user will encounter very different scenarios for image submission. This is true not only when using a low-code approach like AI Builder, but also when there is the need for developing a custom object detection model using for example neural networks in a deep learning approach. Image data quality, quantity and variety are key for the performance metrics in an object detection scenario.

Figure 62 - A sample from the image data of the manually collected dataset

Having manually collected the image data, the final dataset is comprised of three different classes of products to be trained in the model: “Sumol”, “Compal” and “B!”. The whole dataset is comprised of 236 images, resulting in a very slightly imbalanced dataset of 84 images of the feature “Sumol”, 100 images of the feature “Compal” and 82 images of the feature “B!”.

5.2.3. Model and Analysis

With the image data collected and the final dataset ready, it was time to use it in AI Builder. Logically, the model that was chosen amongst the AI Builder catalog was the Object Detection model (explained in further detail in chapter 4.5).

The first step in the model definition is to choose the model's domain. The Object Recognition model features three distinct domains, including Common Objects, Objects on retail shelves and Brand logos. Each domain caters and tailors the model to a more specific scenario and, in this case, the chosen domain was "Objects on retail shelves". The next step is to specify the three different classes of objects ("Sumol", "Compal" and "B!") that we want the model to be able to identify. Once the classes are defined, the image data needs to be uploaded to the service (these can be done either from local storage or through other storage methods like SharePoint or Azure Blob Storage).

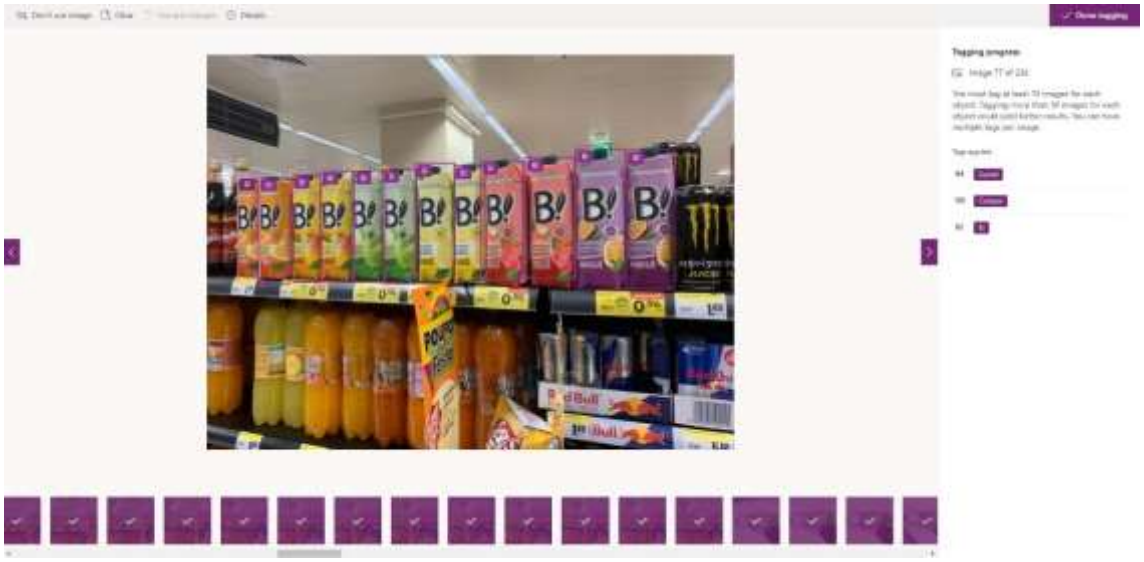

Figure 63 - The tagging phase of the Object Detection Model in AI Builder

Once all 236 images were uploaded, the image tagging phase begins. This can be a rather lengthy process, that involves going individually through each uploaded image and manually tagging the targeted objects in each photo. The engine supports multiple tagging per image, allowing the developer to train the model with several different objects in the same image. This process consists of drawing a container around each object that is present in the picture that we want to train the model to identify and associate each tagging with one of the classes that were defined previously. Once this phase is completed, a model summary is presented stating the instantiated model type, the model's domain, the number of images per data source that comprise the dataset, as well as the different object classes that the engine will train the model to identify and the respective number of tags.

Once all image data is tagged, the model is ready to be trained. The AI Builder engine automatically partitions the dataset into training data and test data and leverages the cognitive services from the Azure cloud under the hood to find the model that best fits each individual dataset. Depending on the dataset size, the training phase can vary in duration.

5.2.4. Results and Conclusions

After the model is trained, the engine outputs a performance score based on the precision, recall and F1-score of the model. In this PoC, the model's performance was 83 points from a 0 to 100 scale. Although 83 is a very respectable score in this context, like any other Machine Learning problem, this score could be improved by training the model with even more image data. This option was not explored, because of the tight deadlines in the presentation delivery and because the concept of the solution was proved with the obtained score.

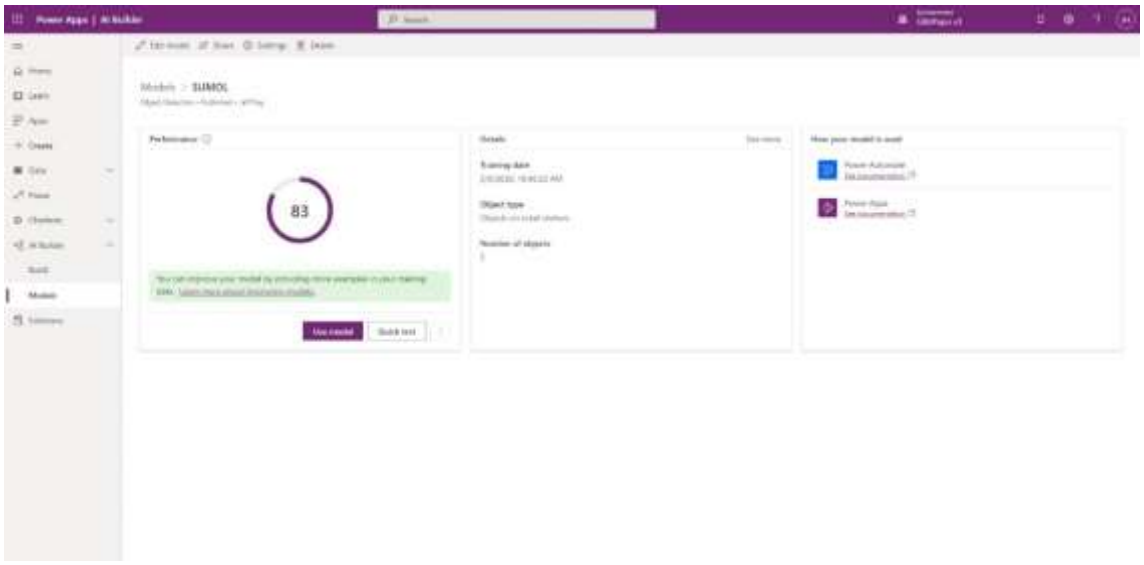

Figure 64 - PoC Object Detection model with the performance score

Once the model is trained and its results are up to standards, it is published, making it discoverable and usable in Power Apps and Power Automate, by developers who have permissions to do so. At any point, a model can be unpublished if there is the need for editing said model. This means that if more data is collected or any business requirements change, developers can retrain the existent model with this new data while retaining the work that was previously done, simply adding on top of what was already configured.

In this presentation, an overview of the entire platform was delivered so that the possibilities of all components were clear from the customer's side. A demonstration of this solution was done, which culminated in a live test of the Object Detection model with physical products that were present in the customer's meeting room, making the feedback of this engagement positive. The customer was pleasantly surprised with the speed that the Power Platform could be used to deliver an otherwise very lengthy and complex solution. The main strengths included the agility with which the technology could embed AI capabilities in an enterprise business solution, the configurable data platform of the Common Data Service that supports all the infrastructure underneath, as well as the component framework for the applicational interface with which the end users interact, that allows the reutilization of premade components in a very efficient way, decreasing even more the time to market of line of business solutions. One improvement point that was made from the customer's side was that the out of the box UI that Power Apps provide is a bit old-fashioned and could be more user-friendly, increasing the development effort from developers in making the user interface compliant according to usability heuristics.

5.3. Invoice Processing Solution with OCR and RPA

This demonstration was prepared and presented to a variety of different customers, since it addresses a very critical and common cross-industry use case that many organizations are challenged with. In addition to this, it is a solution that combines nearly every component of the Power Platform, and it also showcases the integration with some components of Office 365, making it a very powerful way of showing platform capabilities to customers. Nowadays, plenty of organizations still deal with manual processes behind their invoicing system, some of them with paper-based invoices, others with invoices as PDF files or in any image format. In addition to this, many of them still have an ad hoc approval system for invoices and, pending approval, they might store the data in a particular invoicing system.

Having this in mind, this demonstration shows how it is possible to use the Power Platform to completely revamp an invoice processing system at the enterprise scale. The scenario starts with a typical manual invoicing protocol that many companies use today, often starting with invoices in a paper format or in a digital format, depending on the vendor that they come from. These invoices require approval, either from the direct manager of the employee seeking approval, or from a centralized financial team. Once the invoice is approved, another employee is assigned with the task of manually entering the invoice data in a legacy application that the company uses for managing and storing invoice data. With this current scenario in mind, the demo aims to show how the Power Platform can evolve this process into a far more efficient one.

5.3.1. Solution Overview

Since most companies still deal with two types of invoices (paper-based and digital) from various vendors, the Power Platform solution should be flexible enough to tackle both situations. That is why this solution covers two distinct points of entry when submitting invoices, one strictly for PDF based invoices, and one that allows to take pictures of paper-based invoices (that usually exist in fewer quantities than digital ones) to follow the process flow. In this revamped invoicing system, taking into consideration scenario A of submitting digital invoices, the employee would simply need to send an email to a service account dedicated to the invoicing system, with the PDF files of the invoices that he/she needs to submit in attachment, and they would not need to do anything else, since the email alone is enough to trigger the whole invoicing process.

Figure 65 - Power Automate cloud flow that triggers upon email arrival, with AI model embedded (PDF/scenario A)

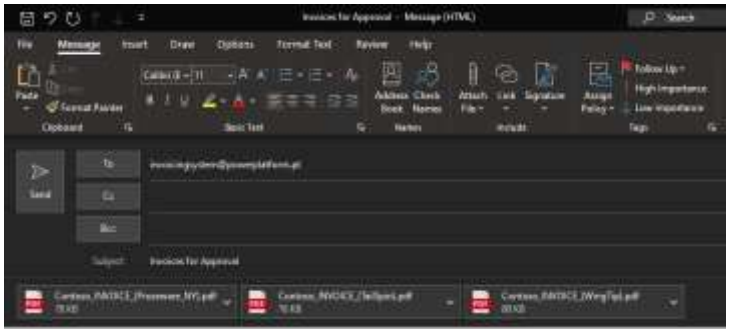

Figure 66 - Employee email with invoices attached for approval (scenario A)

Once the email arrives at the service account inbox, it is automatically scanned to check for invoice attachments present in the email, meaning that multiple PDF invoices can be submitted into the system via one single email. A Power Automate flow is triggered upon arrival of the email that will analyze the attachments by running an AI Builder Form Processing Model that will use OCR capabilities to automatically extract the invoice data from the PDF files. Once the model finishes running, the extracted data is then sent for approval, either to the direct manager of the employee who submitted the approval or to a centralized finance team. This approval comes in the format of an adaptive card that displays the invoice data extracted by the model, and it is delivered to the approver via a Power Automate bot in Microsoft Teams, as well as a new email in the respective Outlook inbox. Besides displaying the invoice information (which is detrimental for the approver to know if the invoice is suitable for approval or not), the adaptive card also presents the approver with two buttons with distinct options: one for approving the invoice submission and another one for rejecting the invoice submission.

Figure 68 - Invoice approval adaptive card through Outlook email

Figure 67 - Invoice approval adaptive card through Microsoft Teams chat

If the latter one is chosen, the employee that submitted the invoice simply receives an email stating that the invoice was not approved with the approver's reasoning. If the invoice submission is approved, the solution continues to process the submission by calling a Power Automate Desktop flow, that will use RPA to open a legacy invoicing application to submit the approved invoice data. This demonstrates the flexibility of the technology in integrating all the components of the Power Platform with the Office 365 ecosystem, as well as integrating with any third-party or legacy systems that sometimes are still crucial for many companies operating at the enterprise scale.

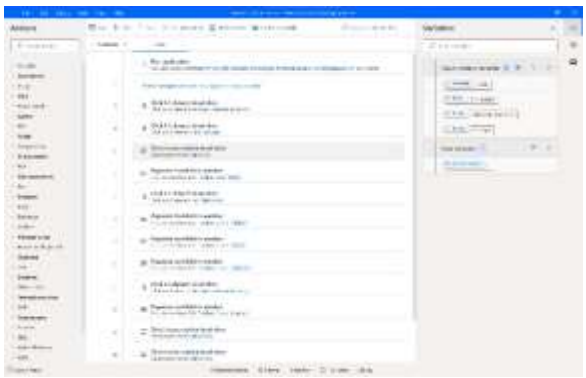

Figure 70 - Desktop Flow that interacts and submits the invoice data in the legacy application using RPA

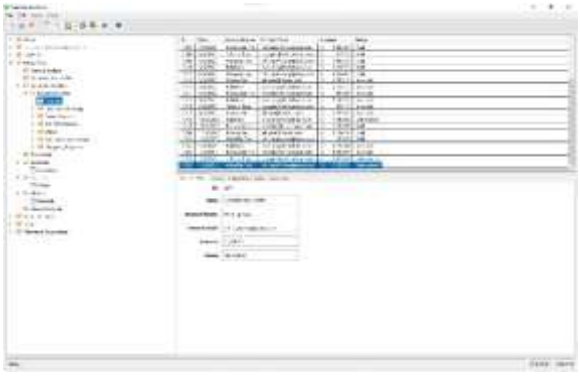

Figure 69 - Contoso Invoicing, the Windows application that serves as the legacy invoicing system for this demonstration

If scenario B is considered (where there is a need of submitting a paper-based invoice), the employee would simply open a Power App that he/she can use to take a picture of the paper-based invoice. Once the picture is taken, the same AI Builder Form Processing model used in scenario A extracts the data from the invoice picture inside the mobile application. Once the data is extracted, it is displayed in the application itself where the user then submits the data for approval. After it is submitted, the process follows the exact same steps as in scenario A, where the data is sent for approval through Microsoft Teams and Outlook, and, if approved, the data is submitted in a legacy application using the RPA capabilities of Power Automate.

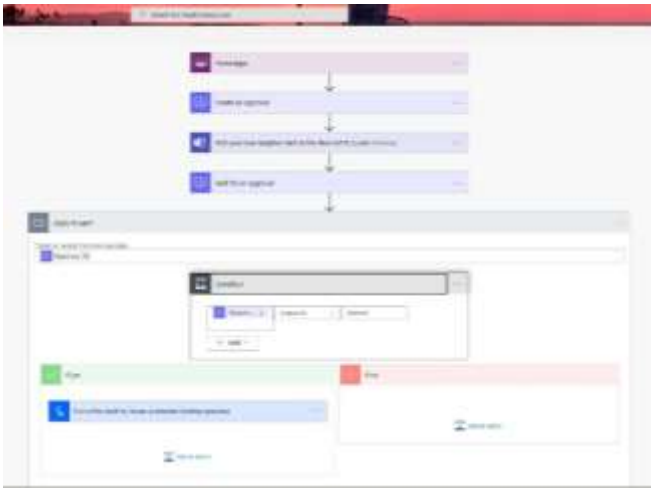

Figure 71 - Power Automate cloud flow that triggers upon manual Power App submission (paper-based/scenario B)

Figure 72 - Mobile Power App with embedded AI model (paper-based/scenario B)

The architecture of the solution is composed by several components: Power Apps, which was the tool that was used for designing the mobile interface, leveraging elements such as text boxes, images and the AI Builder Form Processor component; Power Automate, for defining the several workflows that process the invoice submission, either from the Power App itself or directly form the Outlook inbox; Data connectors, these were used to connect the Power App and the Power Automate cloud flows to online services, including the Approvals connector (to generate the approval artifact regarding the submitted invoice), the Office 365 Users connector (to access logged in user data in the Power App including logged in user name, email, profile picture, and more), the Microsoft Teams connector (for sending and displaying the adaptive card with the invoice data and approval option to the approvers Microsoft Teams chat through the Power Automate bot), and the Desktop Flows connector (for connecting the cloud flow to the Power Automate RPA flow); AI Builder, that allowed to configure the Form Processor AI model that is embedded both in the mobile application as well as in the Power Automate cloud flows (the details regarding said model are described in the next subchapters); Power Automate Desktop, which allowed to define and configure the Desktop Flow that uses RPA technology to enter the invoice data and interact with a legacy invoicing system; two Office 365 applications that serve as the approval interface of choice, either Microsoft Teams or Outlook; lastly, the last component is Contoso Invoicing, which is a legacy Windows application that serves as the in-house invoicing system in the context of this demonstration.

5.3.2. Data Description

Since this is a cross-industry all-purpose demonstration, the data that was used to train the AI Builder Form Recognizer model comes from public sample invoices that Microsoft made available to the public. In the context of a PoC, the model is trained with specific invoices from the customer that the company wants to process automatically.

To showcase the flexibility of the model, the dataset is constituted by invoices with two different layouts, meaning that in this demo the AI model is able to analyze and extract data from two different typologies of invoices. In production scenarios, a single AI Builder Form Recognizer model is able to be trained to handle up to 100 different document layouts.

In this scenario, the dataset is comprised of invoices from two different fictional vendors, “Contoso” and “Adatum”. Each vendor has five different invoices similar in layout, but different in content, making a total of ten invoices for training data. Since this is a purely demonstrative situation, it is only being used the minimum number of documents (five) per layout as training data. When it comes to enterprise adoption scenarios, the Form Recognizer model behaves just as any other AI or Machine Learning problem – the more quality and quantity of data that the model is trained with, the better it will perform. For each vendor, there is also one extra invoice that is reserved as test data for each layout, so that developers can make sure that the trained model performs accurately against unseen data. The five different training invoices from the “Contoso” vendor are image files in PNG format, whereas the training invoices from the “Adatum” vendor are PDF files. This is intentional, and since this demonstration often involves training the model in front of customers, it shows a lot of versatility and ease of use in handling different data types, when non-technical users are concerned.

Figure 73 - One of the training invoices from the "Contoso" vendor

Figure 74 - One of the training invoices from the "Adatum" vendor

All invoices from both layouts contain data that are typical of real-world invoices. This data includes fields and tables like issue and due dates, balances, tax values, shipping costs, billing information like addresses and contacts, as well as a table discriminating the different items that were invoices, as well as the respective quantities, rates, and total amounts. Although there is none, the Form Recognizer model can also extract hand-written data in addition to printed text.

5.3.3. Model and Analysis

With all the necessary invoices needed to train the model, it was time to create a new Form Recognizer model in AI Builder (explained in further detail in chapter 4.5). The first step when creating a new AI Builder Form Recognizer model, it is to define what are the labels of the information that is needed to extract from the documents. These can be either field names for extracting single blocks of text/values, or table names when there is data that needs to be extracted from the documents that reside in a tabular structure. For this specific demonstration, and because the extracted data from these documents will then be inserted into the legacy application, there were four fields that needed to be defined at this stage so that the model could be trained to extract them: the due date of the payment, the name of the entity in the billing information as well as the contact, and the total balance due in the invoice.

The next stage in the model creation, was to define the number of collections that the model needs. Similar documents belong to the same collection, giving the model a way of analyzing their shared structure. Because this demonstration has two distinct structure types amidst the invoice dataset, two collection were created, one for the “Contoso” vendor and another for the “Adatum” vendor. Once the collections are created, the corresponding training data of each collection is uploaded, which can be done directly via local storage, or through online services like SharePoint or Azure Blob Storage.

Once each class of training documents is uploaded to the respective collection, the engine will analyze the uploaded documents. This is an initial study performed by AI Builder to identify the overall structure of the documents in the different classes and make an *apriori* detection of form fields and values. After this is accomplished, the tagging phase begins. This is the step where developers match the labels that were crated in the first phase of the model creation with the actual extracted text strings. This is done through a visual interface where developers can interact with the extracted text from the documents and match them with the predefined labels.

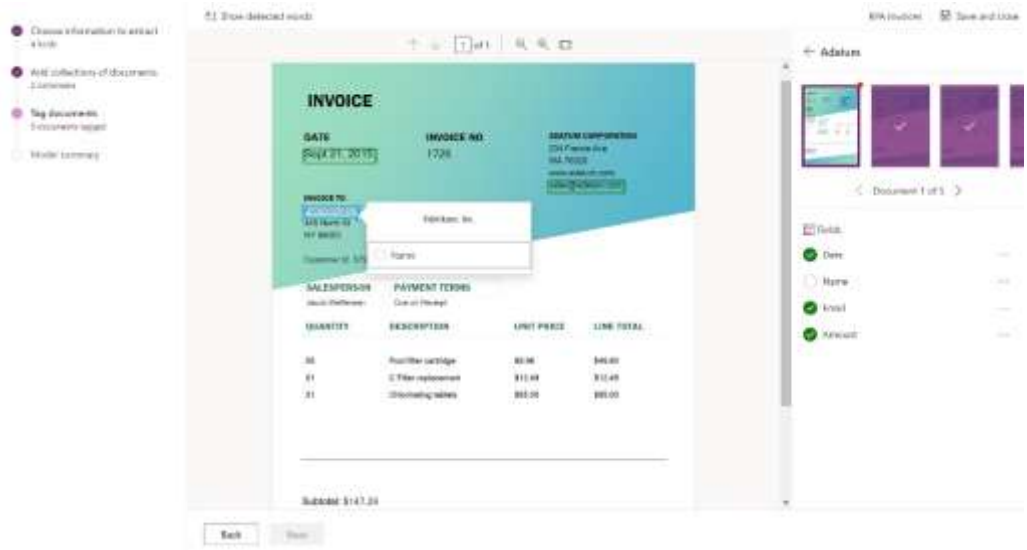

Figure 75 - The tagging phase of the Form Recognizer Model in AI Builder

5.3.4. Results and Conclusion

Once the model is trained, it is ready to be tested by the development team. AI Builder provides an on-the-spot interface for immediate model testing, without forcing developers to embed the model in a Power App or Power Automate flow just for testing purposes. It is in this interface that the test documents that were separated from the training data can be utilized, to check if the model accurately learned the different fields and text data that it was trained to extract.

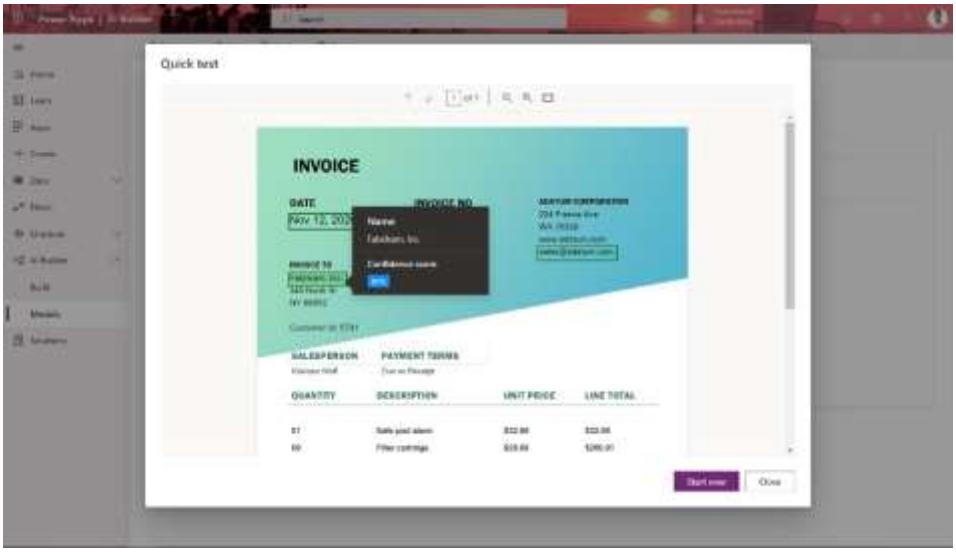

Figure 76 - Quick test pane with detected fields and confidence scores of the Form Recognizer AI Builder Model

Once the testing is completed, the Form Recognizer model should be able to highlight the different fields that were manually tagged on the training data during model creation. By highlighting each identified field, developers can check the actual text data that it is being extracted by the model and compare with the real one to make sure that the model is performing accordingly. For each identified field, the model also outputs a percentual confidence score from 0 to 100.

Once the results are acceptable for production instances, the model is then published to be used in conjunction with the other Power Platform components. In this demonstration, the model is published, and it is then used twice in the invoice processing solution: it is embedded in the Power App for extracting the data from paper-based and invoices, and it is also embedded in a Power Automate flow to extract invoice information from invoice PDFs that arrive as email attachments to a service account inbox. The outputs of the Form Recognizer Model will then feed into the workflow of the approval process and, if approved, that same data will be entered in the legacy system through RPA technology.

This is a very strong demonstration that resonates with nearly every customer that it is presented to, mainly because it addresses a very common business case and because it broadens the horizons to limitless automation scenarios of business processes that still rely heavily on human intervention for analyzing digital/paper-based documents. In addition to this, the demonstration of this solution unlocks several opportunities for proofs of concept using customer's document data, to help them digitize their processes.

6. CONCLUSIONS

Low code platforms are one of the biggest investments being made nowadays by software vendors. They are constantly evolving with more advanced features and new components, to tackle more complex business scenarios and to cater to a wider range of audiences. The opportunities from low code adoption arise not only by empowering non-technical citizen developers in building their own solutions, but also by providing a flexible tool that allows seasoned pro-developers to create and deploy advanced solution faster and in a more efficient manner than with traditional methods.

As of today, there is too much code being written by data scientists, data analysts and software developers that could be made more efficient by using a low code platform, in the appropriate scenarios. Low code has its time and its place, it will never replace custom development, nor it should. Custom development is what drives new technology to be created but, because of the ever-changing enterprise evolution and the demand for custom business solutions, low code platforms are the answer for creating the mindset of a low code/pro code environment, that work better together in helping and empowering everyone to work and perform at their best.

6.1. CONNECTION TO THE MASTER PROGRAM

The master program was detrimental for obtaining a deep understanding about data. The hands-on deeply technical projects from courses like Machine Learning, Deep Learning and Data Visualization provided the pro-development experience that made clear what use cases are out of bounds for a low code approach. In addition to this, the whole master program provides a fundamental overview of every phase of a data related project, which is invaluable in customer conversations where data processes, cloud architectures and tool choice eventually come into play.

6.2. WORK EVALUATION

During this year working as a full-time Microsoft employee, I have learnt more than in any other period in my life. Working in a tech multinational provides limitless learning resources, and Microsoft thrives on growth mindset, by encouraging peer collaboration and tech intensity. I could not be more grateful to work in a company that puts the same care into its employees as it does with its customers and partners, as well as crucial topics like sustainability, equality, diversity, inclusion, and so much more.

6.3. FUTURE WORK

The work that I have done during the period received extremely good feedback from my manager, peers, and colleagues. I will continue to work with Power Platform technologies in evangelizing the value of the platform for citizen and pro developers, as well as data analysts and data scientists.

7. BIBLIOGRAPHY

Aalst, W. M. P. Van Der, Bichler, M., & Heinzl, A. (2018). Robotic Process Automation. *Business & Information Systems Engineering*, *60*(4), 269–272. https://doi.org/10.1007/s12599-018-0542-4

Aguirre, S., & Rodriguez, A. (2017). Automation of a Business Process Using Robotic Process Automation (RPA): A Case Study. *Applied Computer Sciences in Engineering*, *2*, 65–71. https://doi.org/10.1007/978-3-319-66963-2

Brandtzaeg, P. B., & Følstad, A. (2017). Why People Use Chatbots. *Lecture Notes in Computer Science*, 377–392. https://doi.org/10.1007/978-3-319-70284-1

Burns, K. J., Molina-markham, A. D., & Sexton, J. T. (2019). A Taxonomy and Terminology of Adversarial Machine Learning. *NIST: National Institute of Standards and Technology*, 29. https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8269-draft.pdf

Cava, M. della. (2016). *Microsoft CEO Nadella: "Bots are the new apps."* USA TODAY. https://eu.usatoday.com/story/tech/news/2016/03/30/microsof-ceo-nadella-bots-new-apps/82431672/

Chen, I. J., & Popovich, K. (2003). Understanding customer relationship management (CRM). *Business Process Management Journal*, *9*(5), 672–688. https://doi.org/10.1108/14637150310496758

Chen, L., & Nath, R. (2004). A framework for mobile business applications. *International Journal of Mobile Communications*, *2*(4), 368–381.

Cheshire, J. (2020). *Exam Ref AZ-900 Microsoft Azure Fundamentals, 2nd Edition*. Microsoft Press.

Christie, A. M. (2012). *Software Process Automation: The Technology and Its Adoption* (First Edit). Springer Science & Business Media.

Cichocki, A., Ansari, H. A., Rusinkiewicz, M., & Woelk, D. (2012). *Workflow and Process Automation: Concepts and Technology*. Springer Science & Business Media.

Collier, M., & Shahan, R. (2015). *Fundamentals of Azure - Microsoft Azure Essentials*. Microsoft Press. https://books.google.pt/books?id=EfFxBgAAQBAJ&lpg=PA3&ots=1uSRZRWW86&dq=microsoft azure&lr&pg=PA1#v=onepage&q=microsoft azure&f=false

Crichtley, S. (2018). *Dynamics 365 CE Essentials: Administering and Configuring Solutions*. Apress.

Dale, R. (2016). The return of the chatbots. *Natural Language Engineering*, *22*(5), 811–817. https://doi.org/10.1017/S1351324916000243

Evelson, B., Sridharan, S., Perdoni, R., & Yunus, A. (2019). *The Forrester Wave™: Enterprise BI Platforms (Vendor-Managed), Q3 2019: The 14 Providers That Matter Most And How They Stack Up*.

Fayyad, U., Grinstein, G. G., & Wierse, A. (2002). *Information Visualization in Data Mining and Knowledge Discovery*. Academic Press.

Følstad, A., & Brandtzaeg, P. B. (2017). Chatbots and the New World of HCI. *Interactions*, *24*(June), 38–42. https://doi.org/10.1145/3085558

Gebauer, J., & Shaw, M. J. (2014). Success Factors and Impacts of Mobile Business Applications: Results from a Mobile e-Procurement Study. *International Journal of Electronic Commerce*, *June 2015*, 37–41. https://doi.org/10.1080/10864415.2004.11044304

Gens, F., Turner, M. J., Jyoti, R., Rizza, M. N., & Wester, J. (2019). *IDC FutureScape IDC FutureScape : Worldwide IT Industry 2019 Predictions*. *October 2019*, 1–21.

Govindan, G., Hora, S., & Palagachev, K. (2020). *The Data Analysis Workshop*. Packt Publishing.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques* (Third Edit). Elsevier.

Indurkhya, N., & Damerau, F. J. (2010). *Handbook of Natural Language Processing* (Second Edi). CRC Press.

Jackson, P., & Moulinier, I. (2002). *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*. John Benjamins Publishing.

Jämsä-Jounela, S.-L. (2007). Future Trends in Process Automation. *IFAC Proceedings Volumes*, *40*(1), 1–10. https://doi.org/10.3182/20070213-3-CU-2913.00003

Knaflic, C. N. (2015). *Storytelling with Data: A Data Visualization Guide for Business Professionals*. John Wiley & Sons, Inc.

Liddy, E. D. (2001). Natural Language Processing. In *Encyclopedia of Library and Information Science* (Second Edi, p. 15). Marcel Decker, Inc.

Manning, C., & Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Mell, P., & Grance, T. (2011). The NIST-National Institute of Standars and Technology-Definition of Cloud Computing. *NIST Special Publication 800-145*, 7.

Michalak, R. (2015). *Technical Services Quarterly Office 365*. *7131*(April). https://doi.org/10.1080/07317131.2015.1059695

Microsoft Corporation. (2020). *Earnings Release FY20 Q4 Microsoft Cloud Strength Drives Fourth Quarter Results*. https://www.microsoft.com/en-us/Investor/earnings/FY-2020-Q4/press-release-webcast

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. The MIT Press. https://books.google.pt/books?id=0eznlz0TF-IC&lpg=PP9&ots=shrI41Y9Nj&dq=An Introduction to Genetic Algorithms &lr&pg=PP5#v=onepage&q=An Introduction to Genetic Algorithms&f=false

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine Learning* (Second Edi). The MIT Press.

Moore, S. (2015). *Gartner, How to Deliver Enterprise Mobile Apps Faster 2017*. https://www.gartner.com/smarterwithgartner/how-to-deliver-enterprise-mobile-apps-faster/

Myatt, G. J., & Johnson, W. P. (2014). *Making Sense of Data I: A Practical Guide to Exploratory Data Analysis and Data Mining, 2nd Edition* (2nd Editio). Wiley.

Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning*. Packt Publishing.

Richardson, C., & Rymer, J. R. (2016). *The Forrester Wave™: Low-Code Development Platforms, Q2 2016 The 14 Providers That Matter Most And How They Stack Up*. http://agilepoint.com/wp-content/uploads/Q2-2016-Forrester-Low-Code.pdf

Richardson, J., Sallam, R., Schlegel, K., Kronz, A., & Sun, J. (2020). *Gartner 2020 Magic Quadrant for Analytics and Business Intelligence Platforms*. https://www.gartner.com/doc/reprints?id=1-1Y7VEZB3&ct=200128&st=sb

Ruecker, B. (2020). *Practical Process Automation*. O'Reilly Media, Inc.

Rymer, J. R., & Koplowitz, R. (2019). *The Forrester Wave™: Low-Code Development Platforms For AD&D Professionals, Q1 2019 The 13 Providers That Matter Most And How They Stack Up*.

Sallam, R., & Howson, C. (2017). *Why BI and Analytics Adoption Remains Low and How to Expand Its Reach*.

Scheer, A.-W., Abolhassan, F., Jost, W., & Kirchmer, M. (2004). *Business Process Automation* (First Edit). Springer-Verlag Berlin Heidelberg.

Tornbohm, C., & Dunie, R. (2017). *Market Guide for Robotic Process Automation Software* (Issue December).

Varshney, U., Malloy, A., Ahluwalia, P., & Jain, R. (2004). Wireless in the enterprise: requirements , solutions and research directions. *Int. J. Mobile Communications*, *2*(4), 354–367.

Varshney, U., & Vetter, R. (2002). Mobile Commerce: Framework, Applications and Networking Support. *Mobile Networks and Applications*, *7*(7), 185–198. https://doi.org/https://doi.org/10.1023/A:1014570512129

Vincent, P., Iijima, K., Driver, M., Wong, J., & Natis, Y. (2019). *Gartner 2019 Magic Quadrant for Enterprise Low-Code Application Platforms*. https://www.gartner.com/doc/reprints?id=1-1ODOM46A&ct=190812&st=sb

Ward, M., Grinstein, G., & Keim, D. (2010). *Interactive Data Visualization: Foundations, Techniques, and Applications*. CRC Press.

Xia, B. S., & Gong, P. (2014). Review of business intelligence through data analysis. *Benchmarking An International Journal 21*. https://doi.org/10.1108/BIJ-08-2012-0050

Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017). A New Chatbot for Customer Service on Social Media. *CHI*, *May*. https://doi.org/10.1145/3025453.3025496