MODERN DATA MANAGEMENT STRATEGIES FOR MACHINE LEARNING TASKS: A SPORTS ANALYTICS USE CASE ON CLOUD PLATFORM

A Thesis

by

Emrah Mete

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Data Science

Özyeğin University
June 2021

MODERN DATA MANAGEMENT STRATEGIES FOR MACHINE LEARNING TASKS: A SPORTS ANALYTICS USE CASE ON CLOUD PLATFORM

Approved by:

Asst. Professor Erinç Albey, Advisor
Department of Industrial Engineering
Özyeğin University

Assoc. Professor Okan Örsan Özener
Department of Industrial Engineering
Özyeğin University

Assoc. Professor Mehmet Güray Güler
Department of Industrial Engineering
Yıldız Techinical University

Date Approved: 14 June 2021

To My Family

ABSTRACT

There is no doubt that data is the most valuable asset today. The efforts of enterprises in digital transformation and creating a data-driven culture are the most concrete indicators of this. Nowadays, where data transforms all industries, it is possible to follow the rapidly developing technological developments in this field.

Appropriate data management strategies are the basis of creating data-driven organizations. When the evolution of data management architectures is examined, it is possible to say that the biggest factor that triggers this evolution is the changing and increasing data sources and the velocity of data production. In addition, with the increase in the importance of business use cases that need to be done in real time, it has become a very crucial need to process data quickly and turn it into action.

Today when data is strategic importance, enterprises that can manage data correctly could gain competitive advantages. Being able to the correct data management can be built with the support of up-to-date and modern approaches. The infrastructures established by the integration of new and modern methods into the platforms turn into more agile structures. This increases the number of value added services to be produced from data by providing speed and flexibility to organizations. Today, the outputs expected to be produced from data management platforms go beyond descriptive and diagnostic analytic. Now, artificial intelligence, machine learning and data science are important parts of these platforms and these are opened new channels for the future of enterprises.

In this thesis, basic needs and capabilities of modern data management architectures are described and detailed explanations were made on reference architectures in the industry. Besides, data management strategies and expectations were discussed.

An example prototype of the data management platforms, which is explained in detail in this thesis, has also been developed on the cloud platforms. In this prototype, the entire life cycle of the data was considered and each step was developed in detail. In addition, a data science project was developed using the data collected on the platform. Thus, an end-to-end solution has been implemented.

ÖZETÇE

Günümüzde verinin en değerli varlık olduğu şüphe götürmez bir gerçek. Kurumların dijital dönüşüm ve veriye dayalı kültür oluşturma konusundaki çabaları bunun en somut göstergesi. Verinin tüm endüstrileri dönüştürdüğü günümüzde bu alanda hızla gelişen teknolojilik gelişmeleri de takip etmek mümkün.

Veriye dayalı organizasyonlar kurmanın temelinde ise doğru veri yönetim stratejileri ve platformları yer almaktadır. Veri yönetim mimarilerinin evrimi incelendiğinde bu evrimi tetikleyen en büyük etkenin değişen ve çoğalan veri kaynakları ile verinin üretilme hızı olduğunu söylemek mümkün. Bunun yanında gerçek zamanlı uygulanması gerken iş senaryolarının da hayatımıza girmesi ile beraber veriyi hızlı bir şekilde işleyip aksiyona çevirebilmekte oldukça önemli bir ihtiyaç.

Verinin stratejik bir öneme sahip olduğu günümüzde veri yönetimini doğru yapabilen kurumlar rekabette avantaj sağlayabilmektedirler. Doğru veri yönetimi ise güncel ve modern yaklaşımlar ile desteklenerek kurgulanabilmektedir. Yeni ve modern yöntemlerin platformlara entegre edilmesi ile kurulan alt yapılar daha çevik yapılara dönüşmektedir. Bu da organizasyonlara hız ve esneklik sağlayarak veriden üretilecek katma değerli servislerin sayısını arttırmaktadır. Günümüzde veri yönetim platformlarından üretilmesi beklenen çıktılar tanımlayıcı ve teşhis edici analatik çalışmaların ötesindedir. Artık yapay zekaninın, yapay öğrenmenin ve veri biliminin bu platformların önemli bir parçası olmasıyla birlikte modern veri yönetim platformlarından katma değeri yüksek çıktılar da elde edilebilmektedir.

Yapılan bu tez çalışması ile modern veri yönetim mimarilerinin temel ihtiyaçları ve yetenekleri tariflenmiştir. Endüstrideki referans mimariler üzerinden detaylı incelemeler yapılarak veri yönetim stratejileri ve beklentileri tartışılmıştır. Bu çalışmada

detaylı bir şekilde anlatılan veri yönetim alt yapılarının bulut platformları üzerinde örnek bir prototipi de geliştirilmiştir. Geliştirilen prototip mimaride verinin kaynaklardan toplanmasından merkezi veri yönetim platformuna getirilmesine ve bu platform üzerinde ileri analitik senaryolar geliştirilmesine kadar örnek bir çözüm ortaya koyulmuştur. Ayrıca platform üzerinde toplanan veri kullanılarak geliştirilen ileri analitik modelleme çalışması ile tanımı yapılan güncel bir problem çözümleye çalışılmıştır.

ACKNOWLEDGEMENTS

Firstly, I would like to say thank you to Assistant Professor Erinç Albey for his guidance and valuable support. The constant support he gave throughout my thesis period was very important to me. I have been successful thanks to this important contribution he made. In addition to that, I gained a different perspective thanks to what he taught me during my graduate studies.

Secondly, I would like to say thank you to Turkcell as my previous company. Turkcell always supported me during graduate studies that I had been working in company. They encouraged me to pursue a master's degree. Therefore, I am grateful to Turkcell for contributing to my career.

Finally, I would like to say that I am grateful to my family who have always been with me throughout my life, especially to my father Kainat Mete, my mother Belgin Mete and my brother Ömer Mete. I couldn't have accomplished anything without them. Very special thanks to my dear wife Zeynep Mete and my daughter Nil Mete for being my hope every time that I was in despair.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Data and information management approaches are developing and increasing their importance day by day. Today, It would not be wrong to say that data is the most valuable asset for everyone. As stated in many recent studies, it is very clear that data is the new oil [1]. The main purpose is to manage this asset in the most accurate way and to get the maximum benefit from it.

Since the years we started implementing modern data management architectures, we have tried to answer different questions in different times using the data that we store.

At first, we tried to make historical analyzes using the data that we had accumulated. Because in the period when data management was immature, there were big problems with reporting the past and it was not done correctly. For example, companies could not report yesterday's sales accurately. For these reasons, we have built our data management architecture on the correct reporting of past activities. We tried to find the answers that we were looking for by asking questions such as "What happened" to the data that we collected. This approach was referred to as descriptive data analysis. Today, be able to do descriptive data analysis is one of the most important responsibilities of data and information management architectures as it was yesterday.

Although descriptive data analysis solves some of our important needs, the need to learn what the causes of past events emerged after a while. An answer to the question "why did it happen like this?" The answer to this question is very important because we have the chance to shape our future strategies by understanding the reasons for the

events in the field. This approach was referred to as diagnostic data analysis. Today, be able to do diagnostic data analysis is one of the most important responsibilities of data and information management architectures as it was yesterday.

Although descriptive and diagnostic data analysis are very important to better understand what happened in the past, it is not enough to accurately determine our future strategies. It has been found that it is possible to predict the future using the data that we have accumulated over time. Companies applying these methods have managed to gain competitive advantage by getting to know their customers better. With the Predictive and Prescriptive data analysis methods, it is possible to obtain predictions about the future by using the data we have. While these analyzes raise the analytical maturity of the companies to the next level, on the other hand, they provide maximum benefit from the data. Nowadays, it is very important for the future of companies that modern data and information management architectures provide these analyzes to be made. For this reason, almost every company invests platforms where they can make future analysis.

In addition to these needs that we have described, modern data and information management platforms should also provide suitable working environments for decision makers, corporate architects, software developers and data scientists.

With this thesis, a data platform that can meet many needs of today will be constructed. The analytical maturity of the platform will be increased by developing advanced analytical models on the platform. Moreover, it will be obtained that maximum benefit will be getting from the data.

CHAPTER II

BACKGROUND

In this chapter, we examine the information and data management architectures. Especially we overview this concepts, its characteristics and which needs it address. In addition, we discuss modern approaches to how it can be implemented on different platforms.

2.1 Information and Data Management

Approaches to data and knowledge processing are developing and rising their significance day by day. Today, it would not be incorrect to suggest that knowledge is the most important resource for all. As several recent studies have shown, it is very clear that the new oil is the data [2]. The primary aim is to handle and get the full value from this asset in the most precise way.

It is possible to define data management from many different angles. One of the most comprehensive definitions made describes data management as follows. "Data management is the practice of collecting, keeping, and using data securely, efficiently, and cost-effectively. The goal of data management is to help people, organizations, and connected things optimize the use of data within the bounds of policy and regulation so that they can make decisions and take actions that maximize the benefit to the organization. A robust data management strategy is becoming more important than ever as organizations increasingly rely on intangible assets to create"[3]

In today's world where data has a strategic importance, it is desirable to be able to manage this asset properly and to turn it into information. However, today's complex business needs, diversified and growing data sources make this job very difficult. In particular, the growth of data volumes is one of the main challenges in performing

data organization correctly.

The articles [4][5] about data management show that most of the problems that make data management difficult are common. These problems are;

- Data Varieties and Data Volumes
- Data Quality
- Data Governance and Ownership
- Lack of Automation and Manuel Process
- Security and Compliance Issues
- Complex Business Needs

As can be seen, there are several important issues that make data management difficult. In order to manage data correctly, it is necessary to apply specific solutions for each of these issues. On the other hand, also it is very important to keep up-to-date our data management platforms. Applying trend topics such as cloud computing, dev-Ops, continuous development, continuous integration and application modernization in data management platforms is so important for to be modern and agile data management platforms.

When we examine the up-to-date data management architectures, we see that open source applications were used more frequently than before. The main reason for this is that big data technologies are being used more frequently. The increase of data volumes brings along many problems. The inadequacy of conventional systems in processing big data has enabled the development of dozens of open source big data technologies in the industry. Following this development, these tools have taken their place in many data management architectures. At the point that we have reached, we can see open source big data solutions in almost every data management platform.

In this journey where data is transformed into information, it is very important to use technologies that will facilitate data management. For this reason, keeping the data management platforms up-to-date as possible will be beneficial in adaptation to changing needs. It should not be forgotten that the main purpose of each architecture to be built is to use the all data to create data-driven organizations.

2.1.1 Information Driven Business

Obtaining information from data is one of the most important purposes of data management. Many processes are applied to the data to obtain information. The information that obtained is a strategically important value for companies. Today, the most important capitals of businesses are their data and information that obtain from data [3]. The information is the most fundamental object that shapes almost all of the strategies of businesses. Thanks to the information that they produce, companies can both get to know their customers better and identify their basic needs better. Companies are able to better organize all of their systems by managing the information. Organizations managed with a information driven approach can be innovative both the management of internal functions and customer management. Today, companies try to maintain their recruitment workflows, sales and marketing activities, IT systems installations and manufacturing processes with the information they obtain by processing the data that they collect from the field. The most fundamental ability of organizations that work well and healthy is that they have created a corporate culture based on information. Therefore, nowadays, almost every company is working to create a information driven company culture for making difference.

2.1.2 Data Management Reference Architectures

Many different designs have been used in data management architectures from past to present. Generally, each piece that makes up these architectures has been added to the architectures considering current needs. In this topic, we will examine the details

of commonly used data management architectures.

2.1.2.1 Conceptual View of Data Management

When designing data management platforms, all the needs of the industry are taken into account. Therefore, It is highly possible to find similar components in many different reference architectures. Examining these components through Oracle's reference architecture (Figure 1), which is the most widely used in the industry, will provide a more comfortable understanding of the concepts.

Before we get into architectural details, we should look at what today's needs are. In this context, it will be useful to first touch on which business requirement triggers the creation of what kind of architectural components.

When data was moved to data management platforms in batches, what business could do using this data was limited. During this period, the biggest output produced by data management platforms was the daily reports used by business. However, the biggest handicap in these systems was that data management systems lagged behind real time. For this reason, real-time action needs could not be met with these platforms. Although it has been worked in this way for a long time, we can say that business requirements trigger the development of technology in meeting these needs. Because, being able to take instant action through data management platforms was a very powerful skill and had become one of the most fundamental needs of the industry. Due to these needs, following the development of streaming technologies, the data generated in data sources were instantly transferred to data management platforms, and actions were taken over these systems in near real time. Therefore, in modern data management models, it was possible to transfer real-time data as well as batch data from source systems. Today, modern data platforms have a component where real-time data is processed. With this support, our platforms have gained new competencies.

It is seen that the real-time streaming data is processed through the event engine and turned into action. In addition, structural data is taken from sources with batch or mini-batches and transferred to a different component of the data management platform. Nowadays, we can say that RDBMSs are the places where structural data are produced. Certain transformations are applied to data from these sources on the platform, and the results are reported through this platform. Structural data are collected, reported and also stored in the enterprise information store in this environment. In this architecture, it seems that real-time streaming data and data that collected from structural sources are evaluated in different components. The main reason for this is that data are structurally different from each other and require different technologies to be processed. However, it should not be forgotten that after real-time streaming data is processed instantly, it can be stored on Data Reservoir in order to be combined with other data.

It is very important for any company to make predictions about the future and develop advanced analytical applications using data. Therefore, creating value from data is one of the fundamental tasks of data management architectures. In line with this need in the industry, concept units where advanced analytical work can be performed have been added to data management platforms (Discovery Lab. [6]). Within these units, it is aimed to build an infrastructure that will allow companies to develop artificial intelligence, machine learning and data science projects. Studies in Artificial Intelligence, Machine Learning and Data Science play a key role in determining company strategies in all areas. Therefore, these components will increase their importance in data management platforms day by day.

To conclude, data management architectures must be capable of meeting all of today's needs. In addition, the qualified manpower who can use these platforms correctly and apply best practices are the biggest supporters of these platforms.

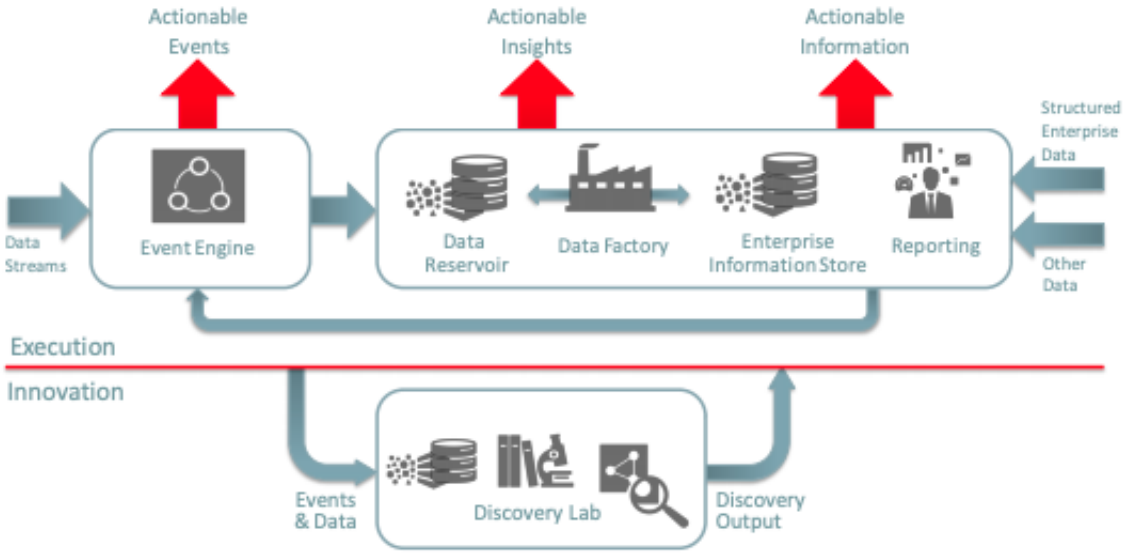

Figure 1: Information Management Conceptual Architecture View. [6]

2.1.2.2 Logical View of Data Management

Logical designs are detailed representations of conceptual architectures. Source systems, data management platform and peripherals can be seen detailed in logical designs. Moreover, the aims and positions of the each conceptual components are described in logical architecture. We can see great number of logical references in the industry. When we examine Oracle's logical reference architecture (Figure 2), which is the most widely used in the industry, will provide a more comfortable understanding of the logical architecture.

There are many potential data sources in the industry that can be a source for data management platforms. While some of these data sources are generated in-house, rest of these come from external data sources. For example, social media platforms are an external data sources. Data sources generate two different type of data. These are structured data and unstructured data. Generally, while social media platforms are generated unstructured data, internal data sources like transactional systems are generated structured data. These data sources are combined in data management

platform using appropriate tools. Generally we can say that the main data sources for data management platforms are transactional systems which these are internal data sources.

Reading data from source systems and bringing it to the data management platform is not a simple operation. This operation varies according to the volume, type and layer of the data to be placed. These operations are performed in data ingestion [6] or data transformation layer. The data passes through this area and is transmitted to the data management platform. The cost of the operations to be performed on the data in this layer is variable. If data is to be modeled, the cost of data modeling will incur. But if the data is sent to the platform to be stored with its raw form, the cost of transformation will decrease here.

There are multiple data storage units within the data management platform. The data held by each of these units has different characteristics. Raw and unmodified data can be kept in the data reservoir [6]. This data can be structured or unstructured. The data to be kept in the data reservoir unit are brought from source systems without transformation. Therefore, transformation costs are low. The data to be kept in this unit can be stored with big data technologies depending on their size. On the other hand, Modeled data are kept in data warehouses. In data warehouses, data are kept in accordance with some rules. Because it is necessary to apply some approaches to make a data warehouse. Generally, Data warehouses are built on RDBMSs. Bill Inmon and Ralph Kimball have revealed in their books what features modern data warehouses should have. One of the most common architectural approaches in the industry belongs to Bill Inmon. According to Inmon, the features that the data warehouse should have are as follows [7].

- Subject-oriented
- Time-variant

- Non-volatile
- Integrated

Another unit in data management platforms is the data discovery area. In this area, artificial intelligence, data science and machine learning projects are developed by using the data accumulated on the data management platform. Mostly data engineers and data scientists work in this field. Designing advanced analytical applications and predictive models are developed in this area.

Any results produced in the data management platform are served through the interpretation [7] layer. Some transformations may be required to serve data in the interpretation layer. The cost of transformation is higher for the data coming into the system without being processed. However, this cost is less for the data obtained by processing.

Outputs produced from the data management platform are used by different consumers. While this platform can generate reports to be used by business units, it can also generate data to be used by 3-party applications. The results produced by data management platforms feed both internal and external business.

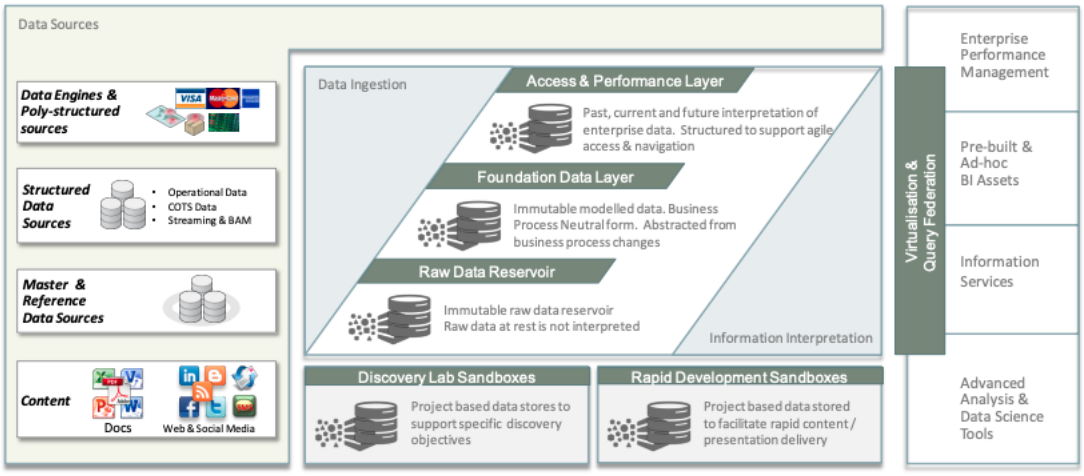

Figure 2: Main components of Information Management Reference Architecture. [6]

2.1.3 Concept and Characteristics of Data Lake

Data lake is a central data management unit that can keep all kinds of structural and unstructured data. In data lakes, data are kept in their raw form, so there is no transformation cost during the transfer of the data here. Data comes from different data sources to data lakes. These sources; It can be relational databases, NoSQL databases, social media data, data files (.csv, .dat, etc ..), iot sensors, mobile apps, event engines.

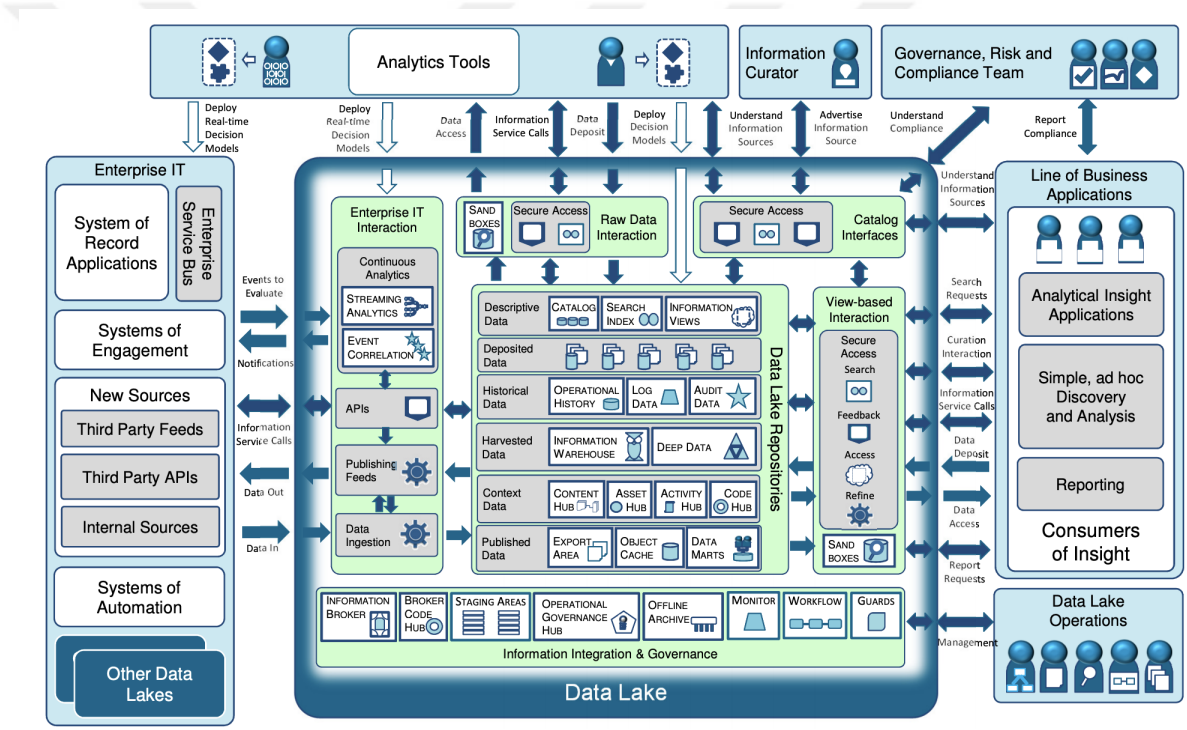

Figure 3: Data Lake Logical Architecture, IBM Corporation. [10]

The main purpose of data lakes is to provide an environment for analytical studies on the data collected here. It is easy to create because the data comes as it is.

Since the volume of data collected in data lakes is very large, today data lakes are built on big data platforms. The main reason for this is that big data platforms can store and analyze structured or unstructured data sets easily. The data collected here is processed by combining appropriate technologies. At the same time, these data are

fed into machine learning models and advanced analytical models, allowing the rapid development of predictive models.

Figure 4: Repr. of Data Lake [9]

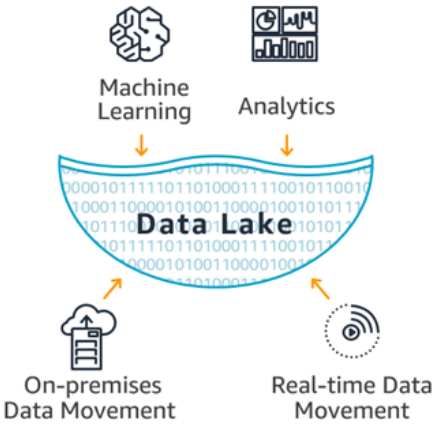

Figure 5: Data Lake [8]

2.1.4 Concept and Characteristics of Data Reservoir

Data reservoir not only indicates a physical environment, but rather informs the characteristics of the data and the rules to be followed. This structure points to different types of data storage units rather than addressing a specific technology. These units can contain relational databases, NoSQL databases, files, key-value stores, graph databases or message buses. Therefore, we can say that this infrastructure is a polyglot [6] architecture.

The data in the data reservoir can be transformed according to their location. Therefore, governance is important for this environment and some rules are applied in data management. Therefore, it is possible to talk about data modelling for this environment.

Data Reservoirs can be created by using many different data storage technologies together today. However, today, with the development of data management technologies, there are solutions that can store and use different types of data on a single platform.

2.1.5 Concept and Characteristics of Data Warehouse

A data warehouse is a specialized database for analytical queries on data. Its main purpose is to keep up-to-date data of a company and to help identify business needs and renew and improve company functions by providing easy and fast analysis on these data. It is to provide convenience to the business intelligence needed for the company.

A data warehouse centralizes and consolidates large amounts of data from multiple sources. Their analytical skills enable organizations to gain valuable business insights from data to improve decision making. Also it can be considered that the data warehouse is the "single version of truth" for an organization [11].

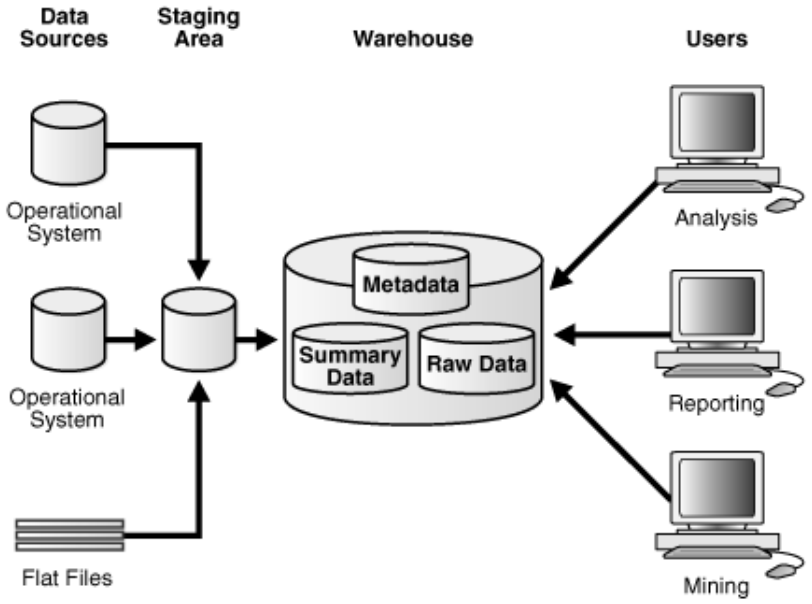

Figure 6: Architecture of a Data Warehouse with a Staging Area [12]

Data warehouses separate the OLTP and OLAP workloads in companies. Therefore, analytical and reporting workloads are not imposed on mission critical OLTP systems.

Data warehouses play an important role in converting raw data into information

thanks to their ability to perform ETL (Extraction, Transformation, Load) operations, data mining and reporting on the data.

A typical data warehouse usually includes the following elements [12]:

- A RDBMS for storing and managing data
- ETL (Extraction-Transformation-Load) tools for preparing data
- Statistical analysis, reporting and data mining skills
- Data visualization tools to design dashboards and report for business users
- Discovery lab for developing machine learning and artificial intelligence applications.

Bill Inmon and Ralph Kimball have revealed in their books what features modern data warehouses should have. One of the most common architectural approaches in the industry belongs to Bill Inmon. According to Inmon, the features that the data warehouse should have are as follows [7].

- Subject-oriented
- Time-variant
- Non-volatile
- Integrated

There are some characteristic features that many major technology manufacturers in the industry have set for data warehouses. These features have been achieved through experience gained over time. According to Oracle, one of the leading technology manufacturers in data management, the data warehouse characteristics are defined as follows.

- Data is structured for simplicity of access and high-speed query performance.
- End users are time-sensitive and desire speed-of-thought response times.
- Large amounts of historical data are used.
- Queries often retrieve large amounts of data, perhaps many thousands of rows.
- Queries often retrieve large amounts of data, perhaps many thousands of rows.
- Both predefined and ad-hoc queries are common.
- The data load involves multiple sources and transformations.

When talking about the data warehouse, it is important not to mention the Data marts. Data marts are subsets of data warehouses. While data warehouses provide an overview of data, data mart's focus only on a specific subject (such as Sales, Marketing, Campaign). Data Marts allow analysis depending on the data needed by certain units and they make analysis easier thanks to the relevant data mart without dealing with all the complexity in the data warehouses.

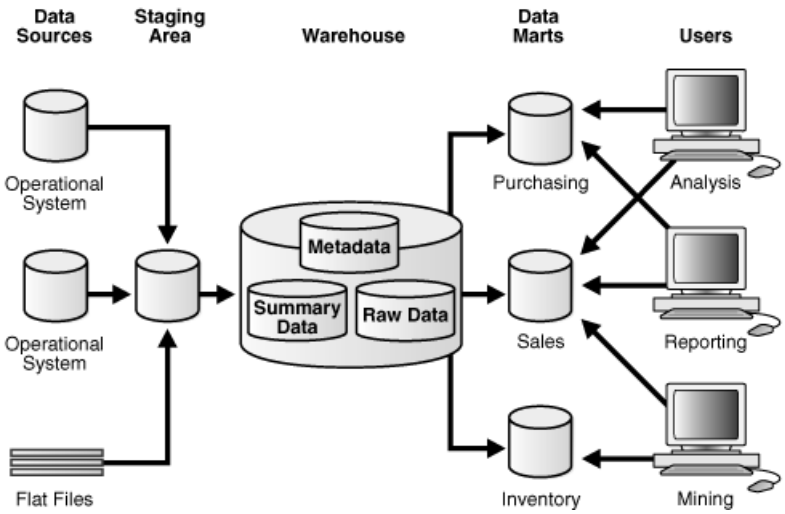

Figure 7: Architecture of a Data Warehouse with a Staging Area and Data Marts [12]

2.1.6 Data Lake vs Data Warehouse

There are some fundamental differences between the data warehouse and the data lake. These differences occur in many areas. The creation processes, user profiles, and the purposes for use differ from each other. Generally, companies create both a data warehouse and a data lake by applying a hybrid architecture. We talked about this situation in the previous sections where we touched on modern data management architectures. Since the two designs address different needs, it would not be wrong to say that companies should create on both platforms. We can see differences between data warehouse and data lake in Table 1.

Data Warehouse using RDBMS technology and typically the data warehouse will have a foundation data layer and an access and performance layer.

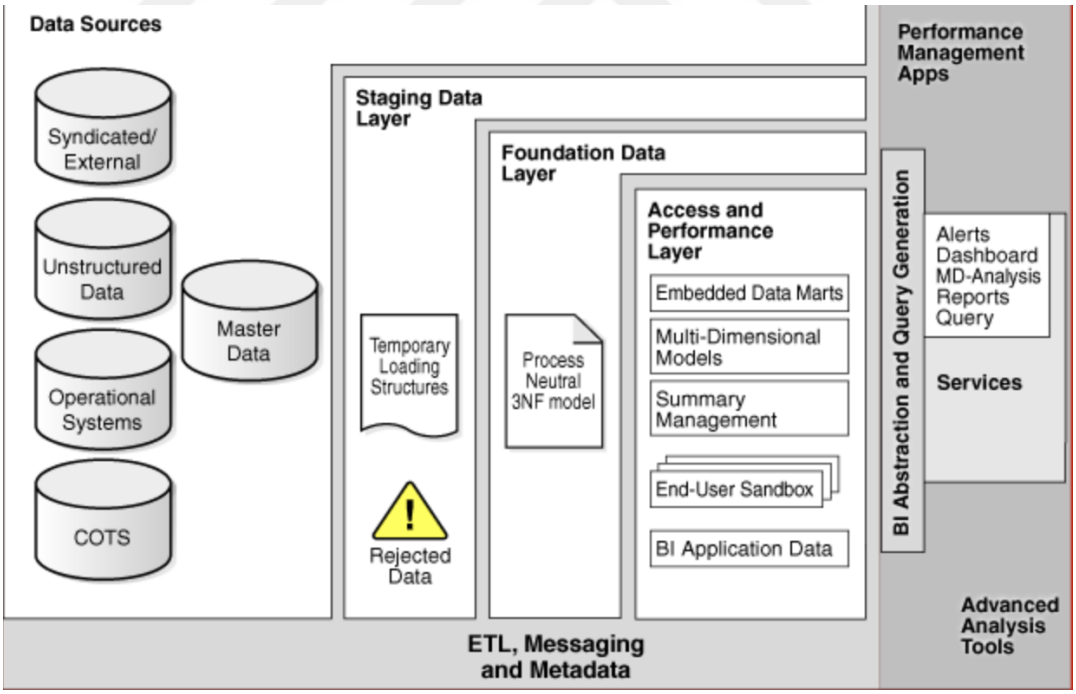

Figure 8: Architecture of a Data Warehouse [13]

The Foundation Data layer is a canonical business-neutral representation of the

data (third normal form) by its nature. The foundation data layer is highly governed; the process of agreeing, creating or changing the canonical data model ensures a substantive depth of understanding of the data and provides at least a notion of the relative data quality of the upstream sources the access and Performance layer is one or more data models built from the Foundation data model, either a bus-specific representation or a performance-related representation the most common transformation for performance is a dimensional data model (or star/snowflake data model) which has a central fact and supporting dimensions other approaches to performance layer improvements include materialized views, in-memory columnar structures and cubes [13].

Data Lake will consist of either a file or object store, and possibly a message bus. Object stores are now preferred to file stores as they provide more ubiquitous access to enterprise data and greater scalability, however file stores may be used as intermediate stores for performance reasons. Data ingested to a data lake will tend to be unaltered from the data source.

Table 1: Data Warehouse vs Data Lake [14][15]

	Data Warehouse	**Data Lake**
Data Structure	structured, processes	structured, semi-structured, unstructured, raw
Strategy	schema-on-write	schema-on-read
User Profile	business users	data scientists
Subject Area	multiple	one
Accessibility	secure and mature	flexible and maturing
Storage	expensive	low cost
Agility	less agile	highly agile
Technology	rdbms	hadoop

As we can see in Table 1, there are many basic differences between data warehouse and data lake. By using these structures that we will establish to meet different needs, we can mature our data management strategy and create more value from data.

2.1.7 Lambda Architecture

In modern data management architectures, the ways and methods of transferring data from source systems to central data platforms are determined with some architectures. One of these architectures is Lambda Architecture proposed by Nathan Marz.

There are two different points in the lambda architecture where the data enters the system (Figure 9). Through these points, data from different sources at different frequencies are received and the relevant fields are fed in central data management platforms. In Lambda architecture, data flows over the batch layer or speed layer to the data management platform.

- **Batch Layer:** Data are transferred from source systems as batches in this layer. Since the transferred data will be taken to more regulatory environments such as data warehouse or data reservoir and transformation is applied to these data. Therefore, it may take a long time for enter data into central data platform. In addition, the data entering the system through the batch layer is written into the platform incrementally. Update and delete operations are prohibited. The purpose of this is to record the history of incoming data in the platform. Also, data transfer and data transformation are performed using E-T-L (Extraction, Transformation, Load) tools in this layer.

- **Speed Layer:** This layer is focused on retrieving data as it occurs in source systems. The reason for this is to be able to run real-time scenarios instantaneously. Since speed is very important in this layer, almost no transformation is applied to the data. Data coming from this channel is usually sent to the data lake part of the central data platform. The main purpose here is to analyze data in real time and trigger relevant actions that determined before. For example, it is very important to be able to understand in real time whether transactions on credit cards are suspicious or not. A speed layer is needed to

instantly receive and analyze this transactions data. Then, system triggers an action if transaction would be suspicious. Producer and consumer based instant message queue technologies are used in this layer. Apache Kafka, Rabbit MQ and JMS are the most frequently used technologies in this layer.

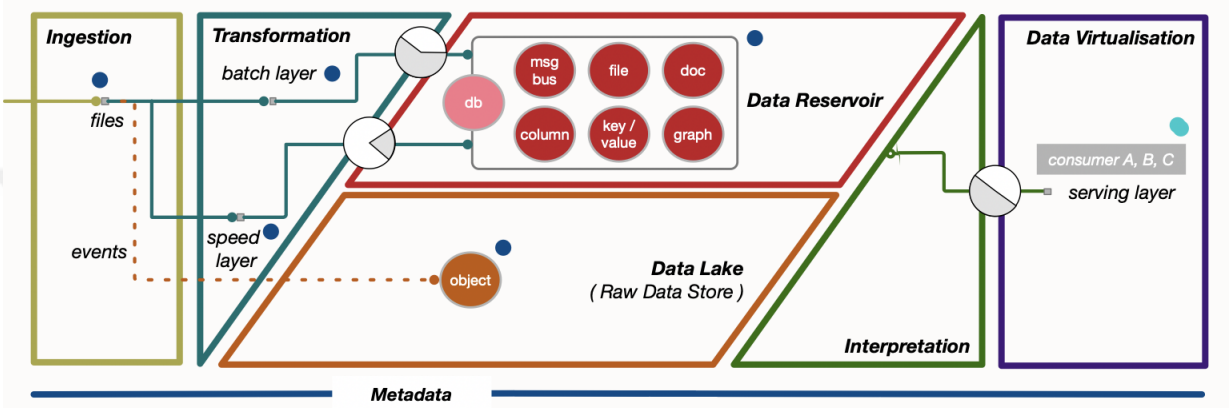

Figure 9: Lambda Architecture [12]

2.1.8 Kappa Architecture

Kappa Architecture has been proposed by Jay Kreps, CEO of Confluent, Inc. and co-creator of Apache Kafka, as an alternative to lambda architecture. It was designed to eliminate the existing complexity of the Lambda architecture. In this architecture, data flow is provided through a single route. Batch layer is not available in this architecture (Figure 10). Therefore, all data flow is sent to the central data platform in real time over the speed layer. The first place where data enters the platform is usually data lakes. Later, the data accumulated here can be transferred to more regulatory environments by applying transformations (data warehouse, data reservoir). Any scenario that needs speed, such as social network applications, cloud based monitoring systems, Internet of things (IoT), can be easily implemented on the kappa architecture.

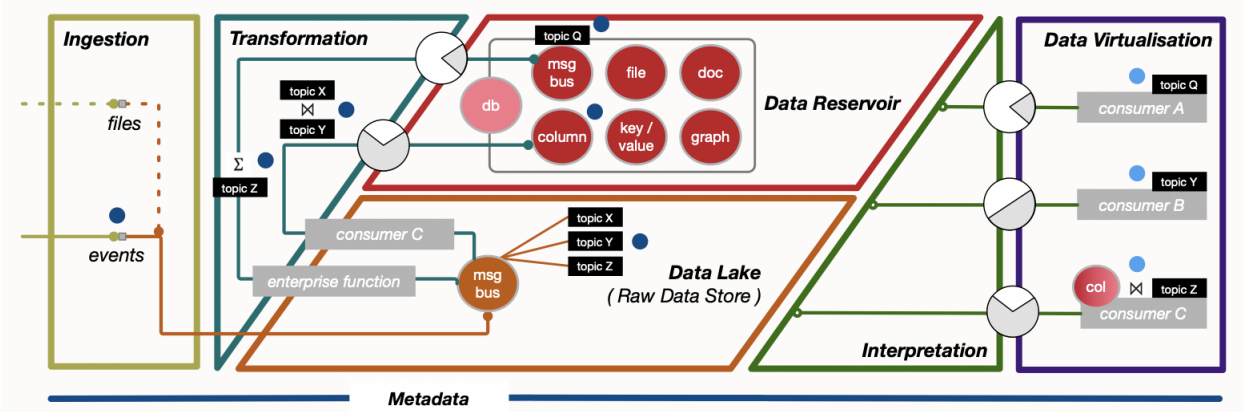

Figure 10: Kappa Architecture [12]

2.1.9 Discovery Laboratory

One of the most important parts of data management architectures is the discovery lab. With the widespread use of advanced analytical applications and methods, the demand for these structures is increasing. In line with this requirement, discovery lab. took its place as an important component in central data platforms (Figure 1).

This platform is designed so that data scientists and machine learning engineers can work freely. In this component, the tools frequently used by the experts are ready-made. In addition, copies of the data collected on the central data platform can be placed in this area. If a data storage unit is to be located in this environment, big data platforms are generally preferred.

While many advanced analytical software can be used in the Discovery lab, it can be preferred in fully open source platforms. With the frequent use of open source libraries in this field, many large technology manufacturers also include these technologies on their platforms. Notebook-based application development platforms are generally preferred by data scientists.

2.2 E-T-L Operations and Reporting for Data Mng. Platforms

We emphasized that we have designed data management architectures to obtain information from data in line with our needs. We need to manage the data well organized so that information can be obtained from the data and evaluated in decision processes. In the flow of data, information and knowledge, some operations may need to be performed directly or indirectly. ETL (Extract, Transform, Load) is one of the most important concepts that help us on this journey. ETL is a three-step workflow (Figure 11). In the first step, data is taken from source systems. This process is called extraction. Then the second step, transformation step is passed. In this step, some transformations are applied on the data and then passed to load stage. In the load stage, data is loaded into the central data management units, which are the target systems. Target systems can be data warehouses, data lakes, or large data pools.

ETL is critical to building central data platforms. Therefore, there are specialized software solutions that can perform all phases of ETL operations. These solutions are called ETL tools and they are very widely used. The most commonly used ETL tools in corporate scale projects are; Oracle Data Integrator, Informatica, Ab Initio, SSIS, Talend, Pentaho and Data Stage.

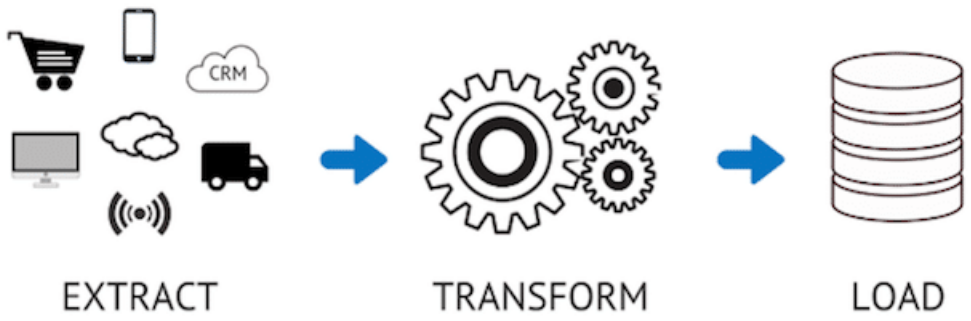

Figure 11: E-T-L Process

2.2.1 Data Acquisition and Extractions

As can be seen in Figure 11, extraction (data acquisition) is the first step of ETL process. data can be taken from data sources in different formats (csv, text file, rdbms, NoSQL databases, etc.). The data is then passed to the transform stage to be passed through a set of rules or functions to prepare them for loading on targets. The type of data source and data dimensions may cause changes in the techniques used in this phase. For example, if the amount of data is small, general purpose drivers such as jdbc or odbc can be used. However, when the amount of data grows, different solutions are needed to read the data effectively.

2.2.2 Transformation

The transform stage includes operations aimed at bringing the data to be written to the target into a certain format. At this stage, the data collected from different systems are transformed into the same formats and integrated with each other. For this reason, the transform stage is where the most effort is made. An important function of this phase is to improve data quality and control data quality. For this reason, various cleaning processes are also applied on the data in this phase. In addition, column selection/filtering, data translation, de-duplication and data joining with keys are the most frequently transformations that applied in this stage.

2.2.3 Load

Load is the last step of ETL process. In this stage, transformed data is transferred to central data management platforms. In this phase, Some different strategies are also followed while uploading data to target systems. These strategies can be determined that depending on the amount of data loaded and the format in which the data will be kept. Truncate/Insert, Incremental Update and Slowly Changing Dimension are the most common used data loading strategies in this stage.

2.2.4 Reporting

It is very important to report the data that collected and processed on central data management platforms. Companies need to analyze and report their data in order to improve their internal processes. As important as analyzing the data, it is also very important to be able to report the analyzed data with the right techniques and tools. In addition, banking, insurance and telecommunication companies are obliged to make legal reporting against the government. Therefore, companies invest in reporting technologies and reporting infrastructures.

One of the most important components of the reporting area is data visualization. Data visualization allows us to better understand the data sets we examine (Figure 12). Thanks to modern data visualization tools, many different graphics can be used in reports. Line graphs, bar graphs, pie charts, flow charts, scatter plots, sankey diagrams, tree views are the most commonly used graphs.

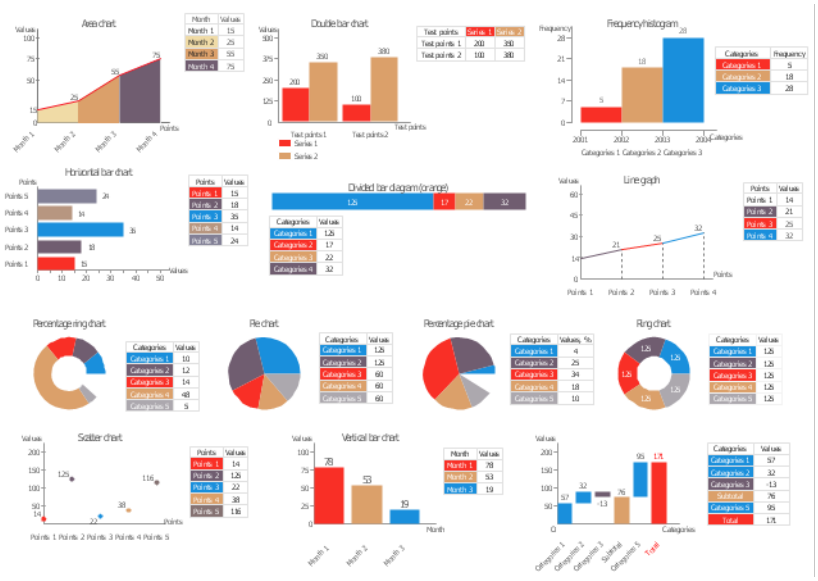

Figure 12: Different Type of Data Graphics

With the development of reporting tools and mobile technologies, companies can consume the reports they produce on different tools such as tablets, smartphones or

web pages. In addition, with the development of artificial intelligence technologies in recent years, it has been possible to work reporting software with digital assistants. Anymore users can create reports by speech.

2.3 Running Machine Learning Models on Data Management Platforms

We have discussed in previous sections that being able to do predictive and prescriptive analytic are one of the most important functions of data management platforms. We even mentioned that these studies are handled in specialized components such as the discovery lab. Predictive and Prescriptive analytic require more sophisticated techniques, unlike traditional data analytic approaches. At this point, machine learning comes to our aid.

2.3.1 Machine Learning

Machine Learning is a method that makes inferences from existing data using mathematical and statistical techniques, and makes predictions about the future with these inferences [16][17].

Machine Learning algorithms learn through data collected in the past or gained experience. The size and quality of the data used in the learning phase is very important in machine learning. If working with high quality and large amount of data, the learning capacity of the algorithms can increase. If data sets are small and low quality, algorithms will be under-fit. For this reason, data collection and data pre-processing are the most important steps in the life cycle of machine learning projects. Then, The golden rule for learning is to have a reliable data set.

Nowadays, we can easily see machine learning applications in many areas. Voice recognition, image recognition, natural language processing are among the most popular machine learning topics today. Many applications such as autonomous vehicles, robots, recommendation engines, digital assistants, chat-bots, language translation

applications are developed with machine learning techniques. In this context, we observe that the learning capacities of machine learning algorithms are very high and this capacity is increasing day by day. It is obvious that with this increasing capacity, it can solve much more difficult problems in the future.

Fundamentally, it is possible to collect machine learning methods under three different headings. These methods are supervised learning, unsupervised learning and reinforcement learning. The types of problems and solution approaches that each method focuses on are different from each other. Therefore, the problem must be defined correctly before deciding which method to use in a job.

2.3.1.1 Supervised Learning

In supervised learning, learning is done through labels on data. Therefore, the model knows which tags it has learned during training. In the training phase, it is tried to understand how the target variable is determined by using the whole dataset. The algorithm tries to find a relationship between the features and the target variable. That is the relationship between the features and the target variable is modeled. After the training, the model is tested with data that it has never seen before. The aim is to make accurate predictions by using data that the model has not seen before.

Regression and classification are problems solved by supervised learning. The main difference that distinguishes regression and classification from each other is the results that these produce. While continuous values are estimated with regression, discrete values are estimated in classification. Weather forecast, sales price forecast, population (age, height, weight) predictions can be shown as examples of regression. Image classification, voice recognition, fraud detection, and disease detection are examples of classification.

2.3.1.2 Unsupervised Learning

Unsupervised Learning does not use any tag or class information for the data in the data set. There is usually no tag or class information in the data sets that are the subject of these problems. Unsupervised learning algorithms try to find vectors that best represent each sample in the data set. After the vectors representing the data are found, all newly arriving data can be encoded with these vectors.

The most frequently solved problem with unsupervised learning is clustering. In addition to clustering, dimensionality reduction can also be solved with unsupervised learning. Recommendation engines, customer segmentation and targeted marketing are clustering problems that we can show as an examples. Visualization of large-scale data, feature selection and structural data discovery are dimensionality reduction problems that solved by unsupervised learning algorithms.

2.3.1.3 Reinforcement Learning

Reinforcement Learning (RL) is a machine learning technique that deals with the problems of finding the optimum actions that must be done in a given situation in order to maximize rewards.

This learning technique, which is inspired by behavioral psychology, is usually described as follows. An agent in any environment makes certain movements in this environment and gains rewards as a result of these movements. The main aim is to maximize the total reward and learn the optimal policy for the longest range.

The most commonly used applications of reinforcement learning are games. Backgammon computer program that trained by reinforcement learning have beaten human in 1979. Also recently, AlphaGO [18] that trained by reinforcement learning defeated the best GO player in the world. In addition, RL is a commonly used learning technique in robotics, control systems, autonomous vehicles and many other industrial areas.

2.3.2 Machine Learning Algorithms

We mentioned that ML algorithms are able to make predictions without the need for human intervention by learning from data. These algorithms can be collected in different groups according to various problems. These groups are Supervised Learning, Unsupervised Learning and Reinforcement Learning. The list of commonly used algorithms in each group is as follows.

- **Supervised Learning Algorithms:** Linear Regression, Logistic Regression, Decision Tree, k-Nearest Neighbour, Support Vector Machine, CART, Naive Bayes, Ensemble Learners (Random Forest, Bagging, Boosting, XGBoosting), Neural Networks (Multi Layer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM)).
- **Unsupervised Learning Algorithms:**K-Means, Hierarchical Clustering, Mixture Models, Principal Component Analysis (PCA), Apriori, Singular Value Decomposition (SVD), Ensemble Learners, Neural Networks (Auto-encoders, Generative Adversial Network (GAN), Self-Organizing Map (SOM))
- **Reinforcement Learning Algorithms:** Q-Learning, Neural Networks (Deep Q-Learning)

2.3.2.1 Ensemble Learning Algorithms

Ensemble Learning algorithms are one of the most frequently used methods when solving machine learning problems. The main reason for this is that it can be used in almost all machine learning problems. Classification, regression and clustering problems can be solved easily with ensemble learning algorithms.

Ensemble learning is instead of using a single basic learning model, it is to create a new main model by using multiple algorithms together. This is the main idea of

ensemble learning. In this context, ensemble learning generally consists of three steps.

- **1)**Dividing data into subsets.
- **2)**Train each subset with a different model.
- **3)**Find the final result by combining each model output.

Ensemble learning can achieve more accurate results. The reason for this is that the decision is taken jointly through the voting method. The average error is calculated over the error made by the community (1).

$$EnsembleError = \sum_{i=[T/2]+1}^{T} \binom{T}{i} \varepsilon^i (1-\varepsilon)^{T-i} \tag{1}$$

Ensemble learning performance should be constructed on two different perspectives.

- **Accuracy:**Base learners performance should be as high as possible. The higher the accuracy of the learners, the higher the overall performance of the model.
- **Diversity:** Putting the results together does not work if the decisions of all the basic learners are the same. So the greater the variety of predictions, the better.

Almost all ensemble learning methods use diversity and accuracy metrics. However, it is not easy to meet both conditions at the same time. Because these two components are inversely correlated to each other. Highly successful base learners make very close decisions so diversity is reduced.

We can categorize ensemble learning algorithms into two groups as bagging and boosting. Bagging and boosting solve problems with different approaches.

- **Bagging:**It is a method that aims to retrain the basic learner by deriving new training sets from an existing training set (Figure 13). New training sets are generated by selecting n random samples from the entire data set. Each selected sample returns to the training set. Therefore, some samples can be used more than once in different learners. Conversely, some samples may not be included in the new training sets. Each classifier is trained with a different training set, but each classifier is tested using the same test data. Then, the prediction of different learners are combined with reach the final result.

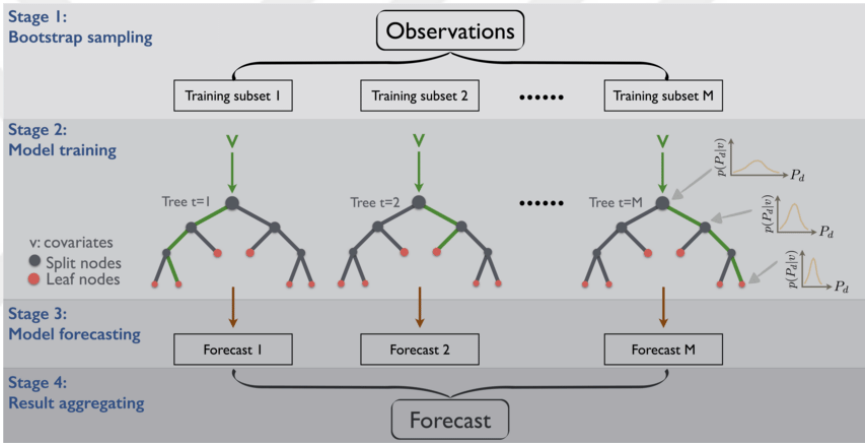

Figure 13: Bootstrap + Aggregating (Bagging) Method [19]

- **Boosting:** Unlike bagging, it works iteratively. First, a subset is selected from the data set. Then, equal weight values are assigned to each point in this data set. The learner is trained with this dataset. Predictions are made with the new model and errors are examined. New weights are assigned considering the examples where the model made mistakes. So the weights are updated. Then new learner focuses on the mistakes made by the previous model and trained with new weights. The one of the most important reason is that errors made by previous model are tried to be fixed. Similarly, by repeating this routine as many as the number of iterations (Figure 14), the best model is tried to be

reached.

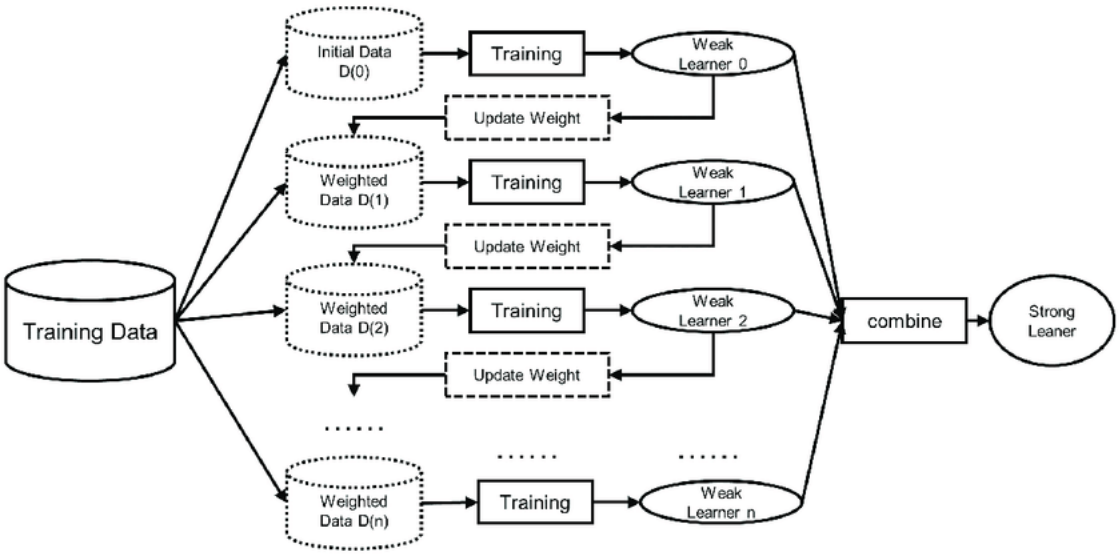

Figure 14: Boosting Process [29]

CHAPTER III

DATA MANAGEMENT PLATFORM

In this section, we examine the details of the data management platform we developed. We define the data model on which we construct and maintain the logical and physical design of the platform. Then we get to know our data sources and examine how data be transferred to the system. At the same time, we get to know the components of the platform and explain which technologies and methods we use here.

3.1 Logical Architecture

In the sections where we mentioned the logical designs of data management architectures, we talked about the components of modern data management platforms. As can be seen in Figure 15, the data management platform that we developed consists of the combination of these components.

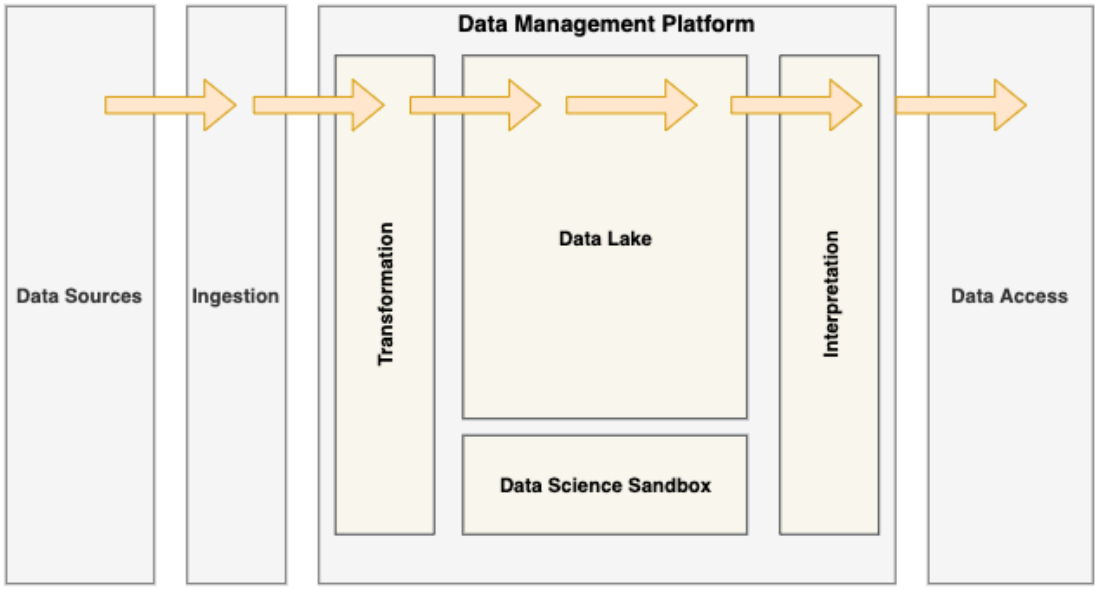

Figure 15: Data Management Platform Logical Architecture

In the selection of the components that platform have, it was decided by considering the characteristics of the data to be managed and the business needs. Since the data we keep on the platform is transferred from different web resources (web services, web crawling) to the platform, the data does not need to be modeled. The data collected from the internet will be transferred to the platform in their raw format without any transformation. For these reason, a component with strict data management rules such as data reservoir or data warehouse was not preferred. The main purpose of the platform is to transfer the data collected from the web directly to the platform and to work on the data. Running queries on data, report generation and development of advanced analytical models will be carried out on the platform. This is why the data lake component, which is a more flexible platform, was preferred for data management and querying of data.

One of the most important purposes of the platform is to be able to develop advanced analytic scenarios on it. For this reason, the data science sandbox component has taken its place in logical architecture. Machine learning and data science applications will be carried out on this component.

Data sources, data extraction and access to data components are also shown in the logical design. By means of these components, the life cycle of the data will be ensured from the platform entrance to the exit.

3.2 Physical Architecture

In the logical design, we decided which components should be in the architecture. In this section, we will talk about which technologies we will use for the components that we have determined in logical architecture. The details of the technologies used and how this infrastructure will work is a very important issue. Also, we will go over these issues in this section.

Apart from the technologies to be used in physical architecture, it is very important to keep the data in the system with a model. Because the data model within the system affects the querying and data integration capabilities of the system. Therefore, we will explain that which logical and physical data model does it suitable for our data to store in the platform.

In Figure 16, the technologies used in each component and the physical organization can be seen. Let's examine the details of these components.

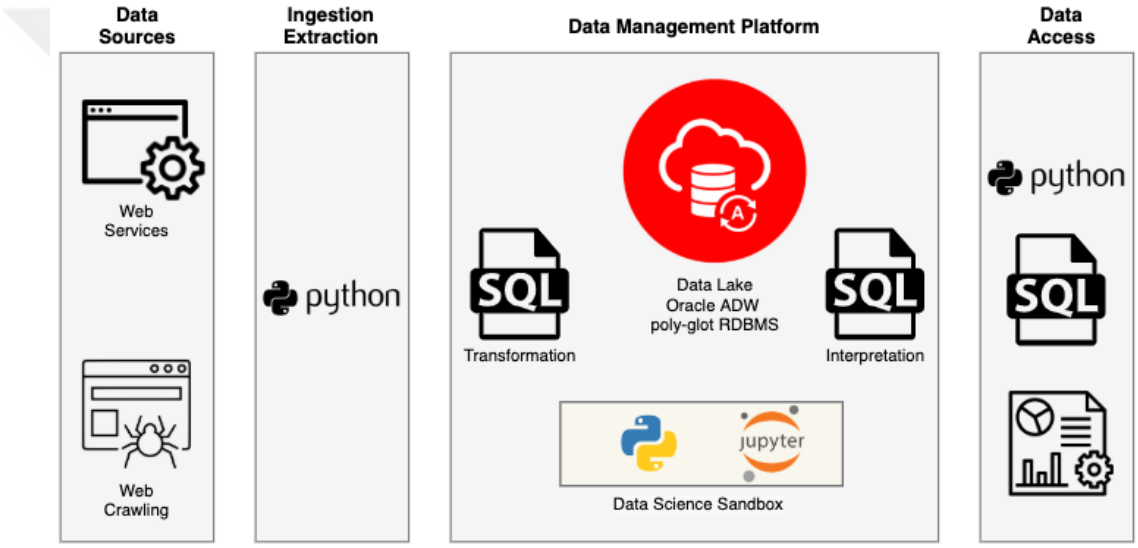

Figure 16: Data Management Platform Physical Architecture

3.2.1 Data Sources

The data we will manage on the data platform is about football, the world's most popular sport. It is possible to find great number of data sources about football on the internet. In this platform, we use api-football web services and Transfermarkt web pages as data sources. It is highly possible to find different types of information about football that we needs from these data sources. In summary, the data sources that we will transfer to the data management platform are web services and web pages.

3.2.1.1 Api-Football Web Services

Api-football offers a platform where we can obtain more than seven hundred league and cup data. With the web services running on this platform, you can get the data you need such as player information, player statistics, transfer information, teams, top scorers and much more.

With the Api-Football service infrastructure, we extract player information, player statistics, season information, player awards, coach data, coach awards, league information, teams and country information's to our central data platform.

The web services and endpoint information that we use are shown in Table 2.

Table 2: Api-football Web Service End Points

Data	**Web Service End Point URL**	**Description**
Player	https://api-football-v1.p.rapidapi.com/v2/players/player/	player information
Player Statistics	https://api-football-v1.p.rapidapi.com/v2/players/player/	player statistics
Countries	https://api-football-v1.p.rapidapi.com/v2/countries	country details
Leagues	https://api-football-v1.p.rapidapi.com/v2/leagues	league information
Coach	https://api-football-v1.p.rapidapi.com/v2/coachs/coach/	coach information
Coach Trophies	https://api-football-v1.p.rapidapi.com/v2/trophies/coach/	history of coach awards
Team	https://api-football-v1.p.rapidapi.com/v2/teams/team/	team details
Player Trophies	https://api-football-v1.p.rapidapi.com/v2/trophies/player/	history of player awards
Season	https://api-football-v1.p.rapidapi.com/v2/seasons	season information
Coach Career	https://api-football-v1.p.rapidapi.com/v2/coachs/coach/	coach career details

3.2.1.2 Transfermarkt Web Pages

Transfermarkt is the world's most popular football information portal. It is possible to find all data related to football on this platform. However, this platform does not have a data service infrastructure. Therefore, it is not possible to receive data from this platform through any services that provide from Transfermarkt. In other words there is a limited amount of data available from Transfermarkt. For this reason, only the market values of the players are obtained from the Transfermarkt web pages and transferred to the central data platform.

We extract the market values of the players that we acquired from the Api-football platform using the Transfermarkt web crawler that we developed.

3.2.2 Data Model

Data Modeling is the process of creating a data model for the data to be stored in the database. This data model is a conceptual representation of data objects, their relationships between different data objects. Data modeling enables data to be visually represented and enforcement of business rules, legal compliance and government policies on data. Data Models ensure consistency in default values, semantics, security and naming conventions while ensuring the quality of the data.

The data model is concerned with what data is required and how it should be organized rather than what operations should be performed on the data. The Data Model is like the architect's building plan, which helps to create a conceptual model and establish the relationship between data items.

In this section, we will show you how to keep the data we collect from our data sources with a model on the data management platform with logical and physical designs.

3.2.2.1 Logical Data Model

Logical data model shows us that relationship between each entity that stored in data management platform (Figure 17). Our data management platform designed as data lake. Therefore, we do not need to store data with third normal form (3NF) and data are stored as raw format on the platform.

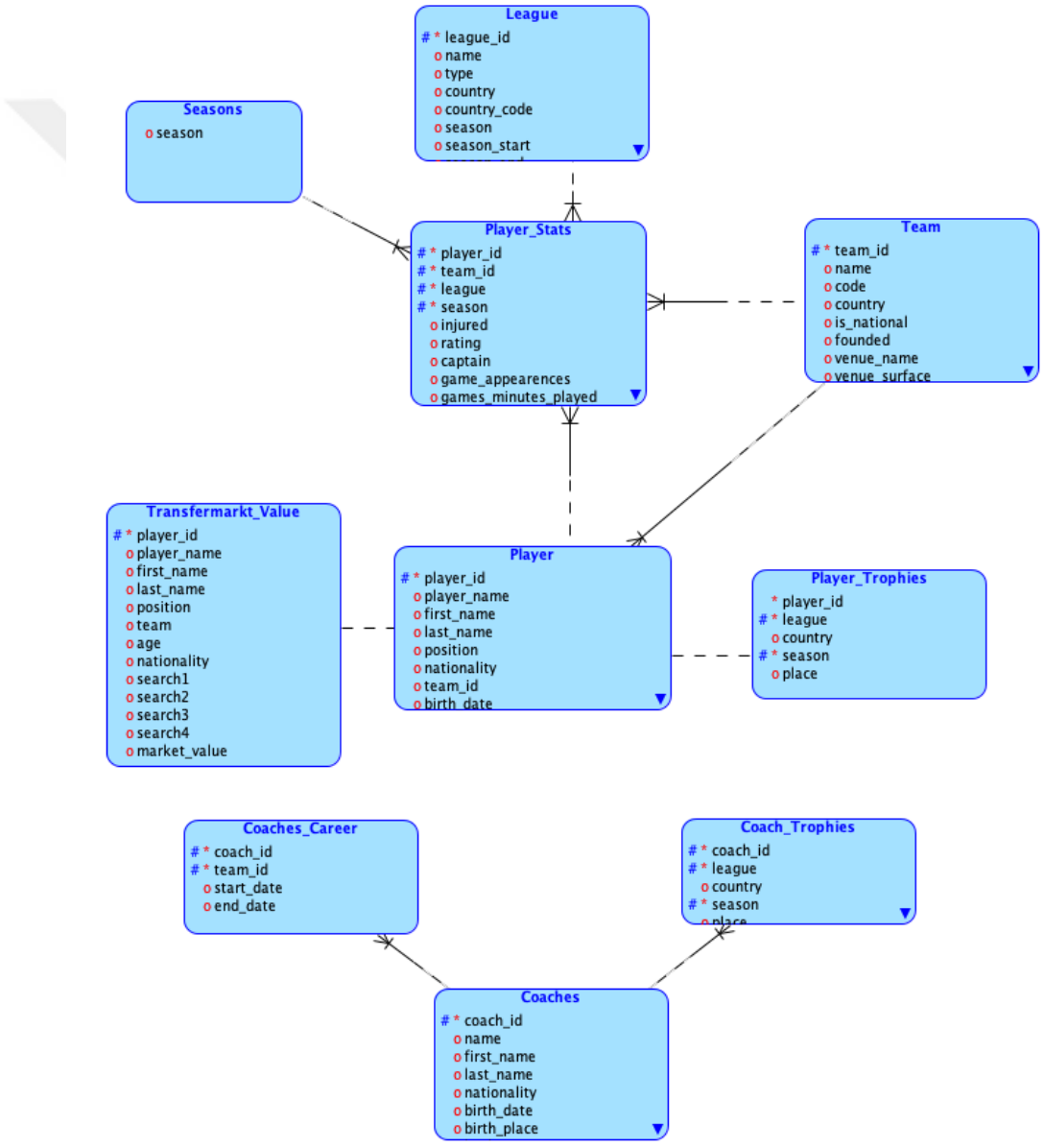

Figure 17: Data Management Platform Logical Data Model

3.2.2.2 Physical Data Model

The physical tables that we built on the logical model are shown in Figure 18. The transformation of entities into physical tables and data types are shown in physical design. Tables specified in the physical model will be created in the data lake where we will store the data.

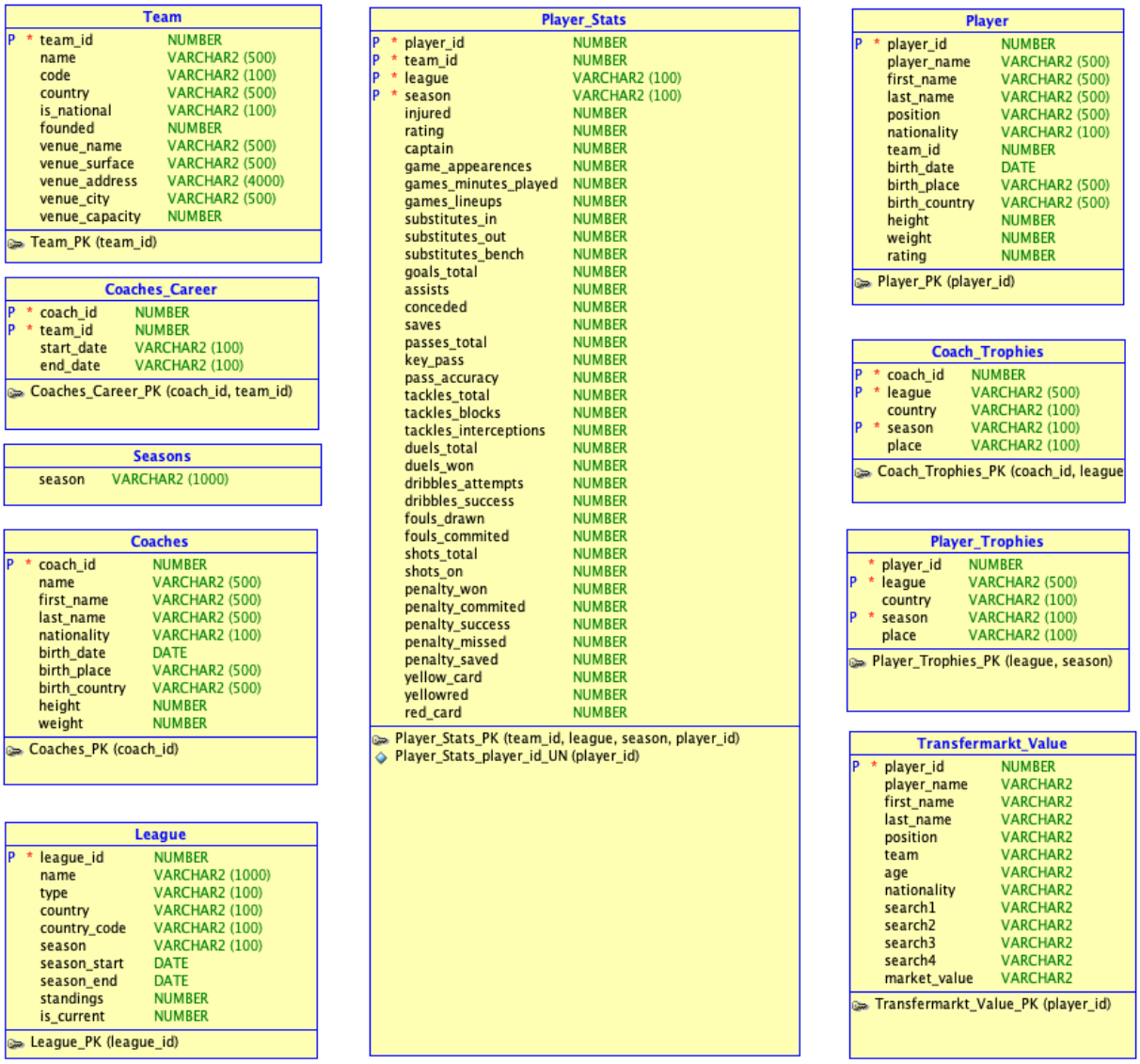

Figure 18: Data Management Platform Physical Data Model

3.2.3 Ingestion/Extraction

The process of extracting data from source systems and bringing it to the data management platform is done in this layer. Today, there are many ETL products that can be used in this layer. However, traditional ETL products are not sufficiently successful in being able to read data from the web or perform web crawling. One of the most effective ways to read data from web resources is to develop custom code. However, we should not forget that developing custom code is a more difficult option.

This part of the platform was implemented by writing custom code. The extraction codes that developed in this stage were written using web libraries in python. Before writing extraction codes, the class diagram was drawn. Then, in the development phase, the specified classes were developed with python. Figure 19 shows the classes and their details (fields and methods).

While we were developing extraction codes, some optimizations have been made to ensure that the codes can work in parallel. In particular, the extractions such as player and player stats that extract large amount of data were enabled to run in parallel by developing a parametric structure. Thus, running a much faster and more efficient infrastructure has been obtained. While we were coding this structure, Pool methods of the Multiprocessing library were used.

Logging is one of the most important parts of software development. Therefore, all the codes we write in the extraction layer record the transactions and transaction results with the log mechanism we have developed. When there is a problem in the system, the problem can be easily detected by examining the logs. Therefore, this infrastructure is used to understand that the system works correctly. The system keep logs with two different granularity level. These are transaction level and record level. The log infrastructure were developed using Logging library in python.

Another mechanism established in this layer is the object relational mapping (ORM) infrastructure. ORM is a programming technique that defines metadata for

connecting an object with a relational database. Objects are developed with languages that support OOP (Object Oriented Programming). Java, C ++, Python are examples of these languages. In short, ORM acts as a bridge between software and database. Data communication between the database and the program has been facilitated by using the ORM structure in the extraction layer we have established. The matches between objects and database tables are shown in Table 3. This mechanism was developed using Sqlalchemy that ORM library in python.

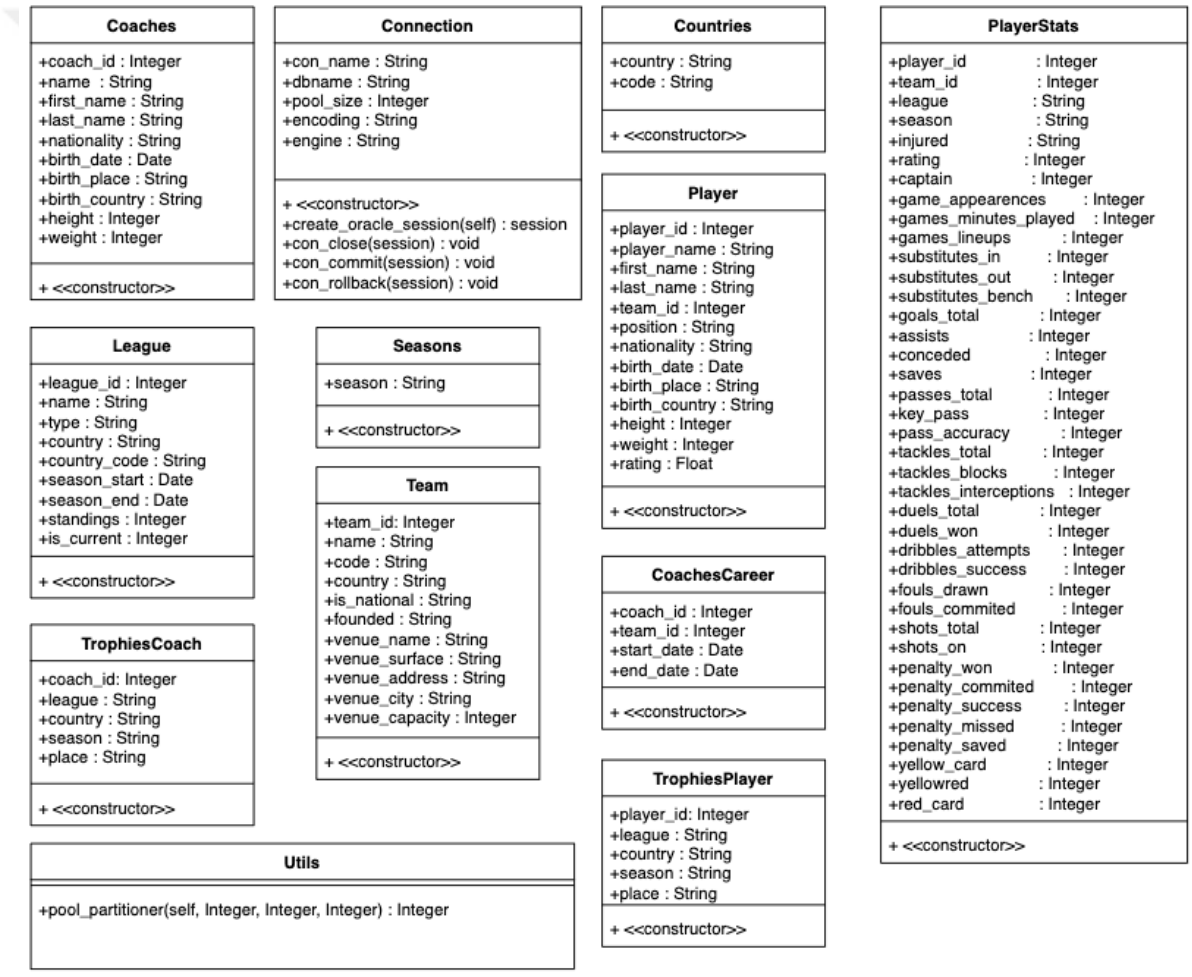

Figure 19: Object Design for Data Extractions: Class Diagram

Table 3: Mapping between Objects and Database Tables (ORM)

Object Name	Database Table Name
Coaches.py	coaches
Countries.py	countries
CoachesCareer.py	coaches_career
League.py	league
Player.py	player
PlayerStats.py	player_stats
Seasons.py	seasons
Team.py	team
TrophiesCoach.py	coach_trophies
TrophiesPlayer.py	player_trophies

3.2.4 Data Lake Implementation

In the previous sections, we mentioned about data reservoir, data lake and data warehouse concepts, which are the basic components of data management architectures. At the points we touched on these concepts, we have also mentioned the purposes of use of these structures and their differences from each other. From this point of view, considering the basic needs in building the platform, it is possible to say that it would be correct to construct the platform as a data lake. For this reason, the perspective and purpose of data management will be to bring the data to the platform as raw as possible and to meet the needs of advanced analytic and real-time reporting.

Data are kept in their raw form in data lakes. This is one of the main features of data lakes. Another important feature is that data lakes can store many different data formats. That is, it can be capable of holding structured, unstructured or semi-structured data. Today, we see that structured data is mostly stored in RDBMSs. This is because RDBMSs work with high performance in storing and querying structured data. However, we can say that most of the traditional RDBMSs are not sufficient for storing unstructured or semi-structured data and querying effectively. For this reason, big data technologies are mostly preferred instead of RDBMSs in data lake environments. The ability of big data technologies to process different data

formats effectively and make them available for use is one of the biggest advantages. However, the applicability, management and maintenance of big data technologies are more difficult than RDBMSs. The fact that there are multiple technologies that should be used for different purposes in the big data world makes the system complex. Therefore, integrating multiple technologies, performing software maintenance and platform management without downtime are very difficult operations in big data environment.

In addition to the capabilities of Big Data platforms (Hadoop Ecosystem), some difficulties in their operations have enabled the development of different approaches in this area. In particular, with the development of cloud technologies, the solutions developed by cloud providers (Oracle, Google, Amazon Web Services, Microsoft, Ali Baba) in this area have made data lake implementations simpler. In addition, in today's proposed data lake solutions, it has become a priority to keep both structured and unstructured data together in a central structure and to use them efficiently.

With the development of cloud computing, the provision of data lakes to users as a managed service is one of the revolutionary changes made in this field. Considering the management, maintenance and cost advantages of the services offered over cloud infrastructures, it is possible to say that almost every company is turning to cloud-based solutions. In this context, many cloud service providers have reference architectures and components for data lake implementation.

Cloud providers have come up with different solutions that will make it easier to use for every subject that needed in their data lake architectures. Now let's examine the data lake reference architectures of Oracle Cloud (Figure 20), Amazon Web Services (Figure 22), Google Cloud (Figure 23) Platform and Microsoft Azure (Figure 21).

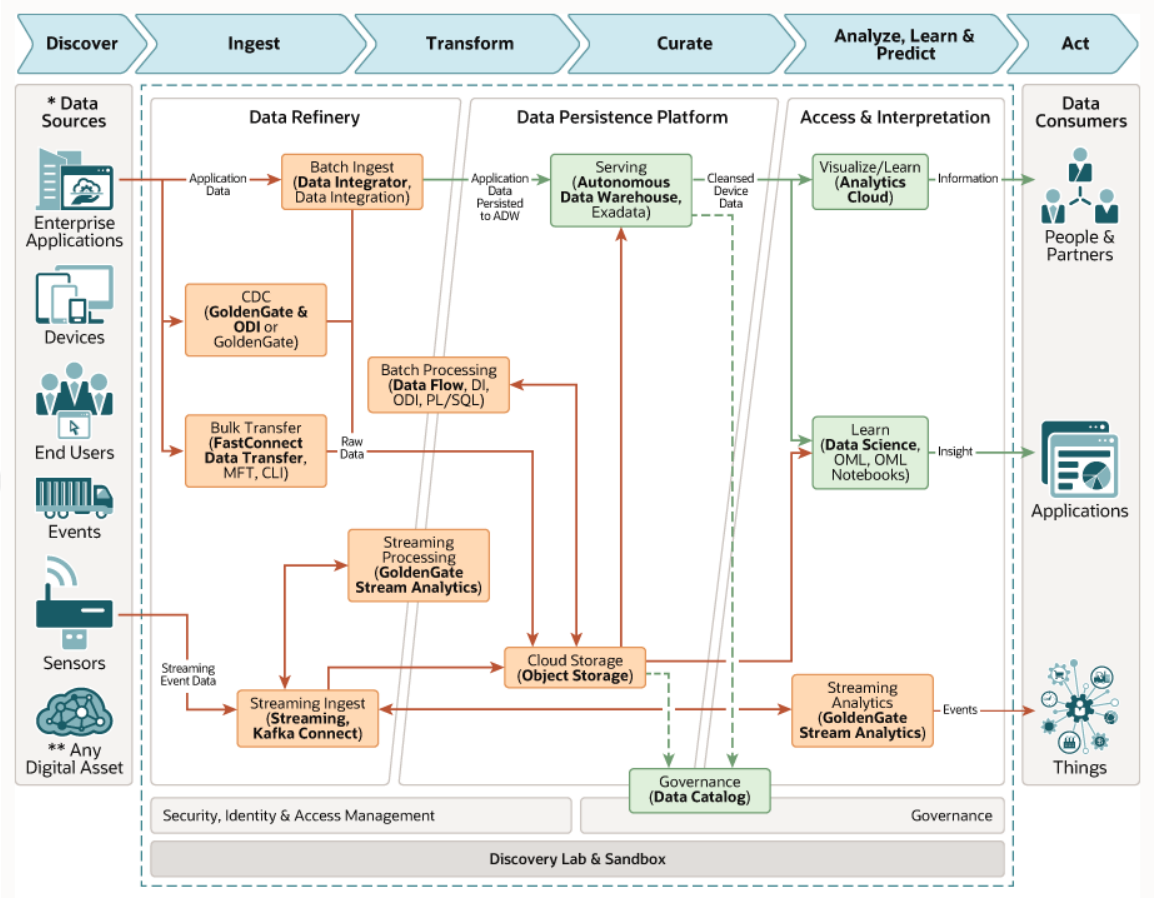

Figure 20: Oracle Cloud Data Lake Reference Architecture [20]

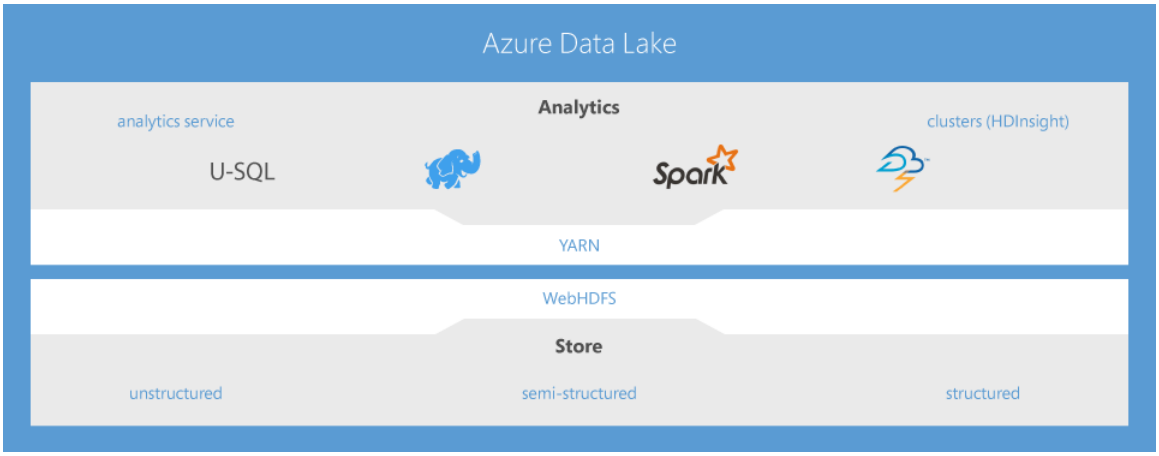

Figure 21: Microsoft Azure Data Lake Reference Architecture [21]

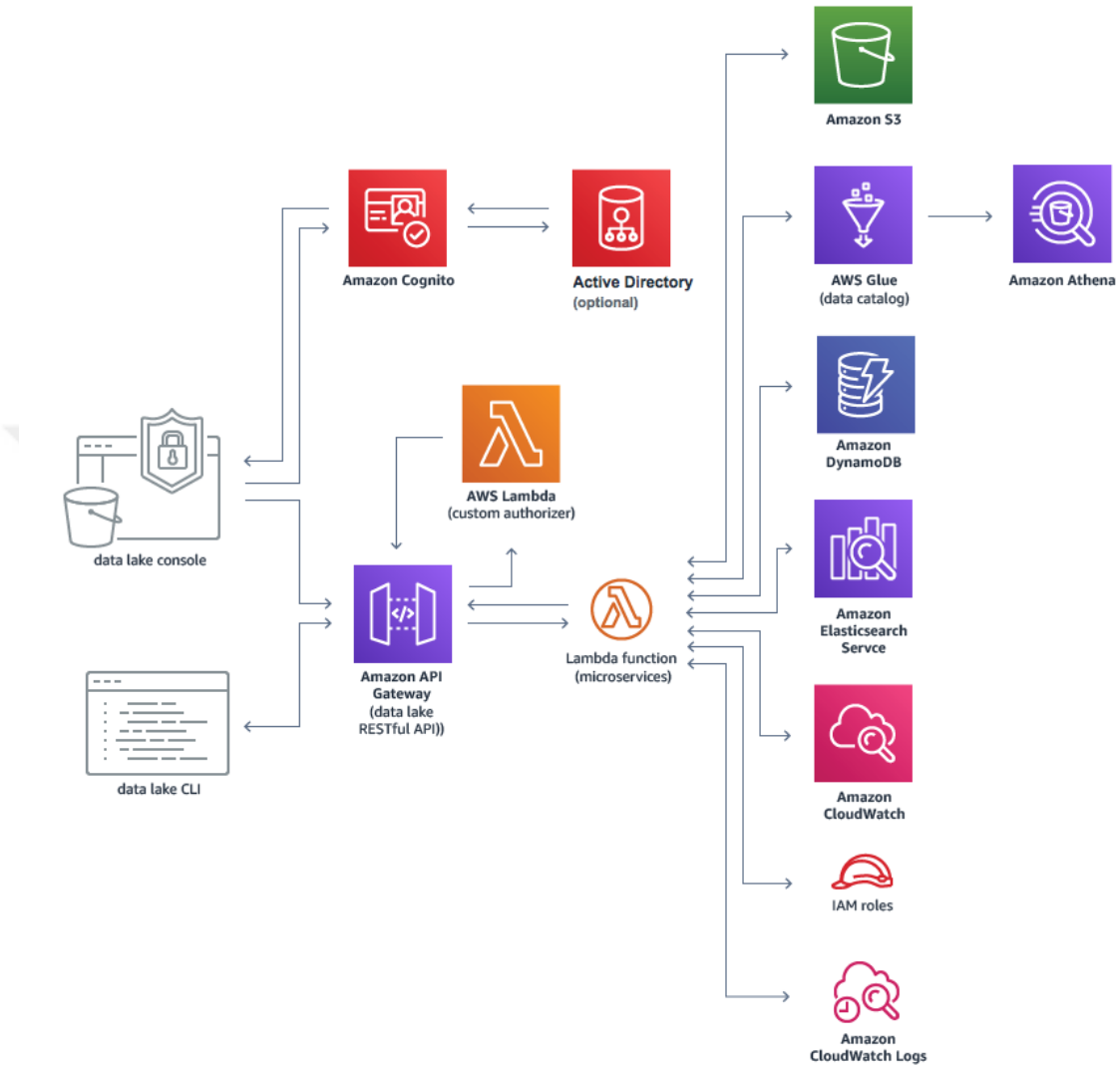

Figure 22: AWS Data Lake Reference Architecture [22]

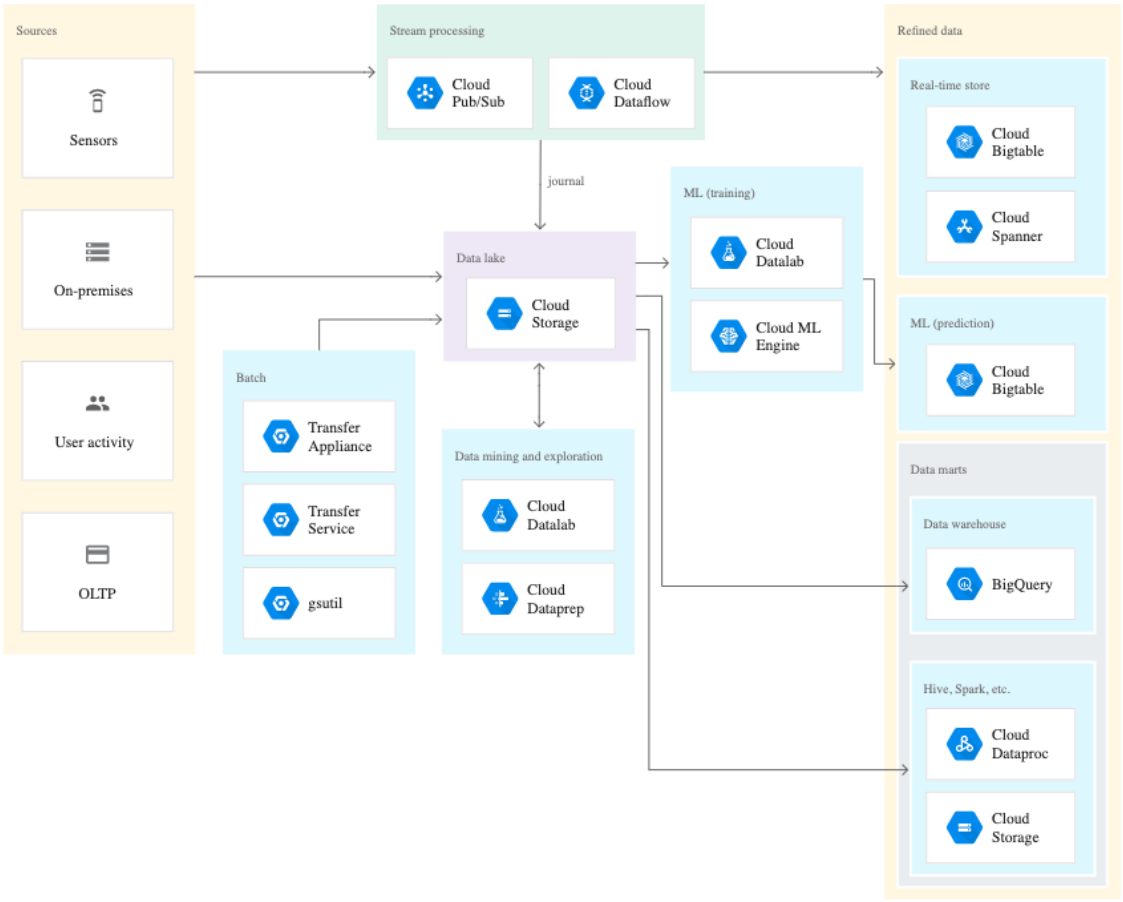

Figure 23: Google Cloud Platform Data Lake Reference Architecture [23]

As can be seen from the reference architectures, cloud providers have produced a service within their own architectures for every need. For this reason, it is possible to implement the data lake implementation, in which we describe the concept design, with the existing services of all popular cloud providers. We will implement this infrastructure with the Autonomous Data Warehouse service offered by Oracle Cloud.

3.2.4.1 Oracle Cloud Autonomous Data Warehouse

Oracle Cloud Autonomous Data Warehouse (ADW) is a cloud database service that Oracle launched in 2018. Although ADW is an RDBMS, it can also store unstructured data as a result of the developments made by Oracle. At the same time, the data stored in this infrastructure can be queried effectively. Building advanced analytical

models on ADW is useful feature for data scientists. It can run very complex queries. Moreover, Data scientists can develop machine learning models on ADW without moving data (aka in-database machine learning). ADW also supports multi model data types. It can store variety types of data. This feature is very important for using ADW as data lake [24].

ADW is an automatically managed infrastructure. In other words, a human resource is not needed to manage ADW. The system is self-managing, taking its own security measures, patching itself, and increasing system performance with continuous optimization [24]. For this reason, it does not face with many problems caused by human errors. In addition, it offers advantages in terms of cost since no human resource is required for system management.

As a result, ADW is one of the suitable solutions for data lake or data warehouse architectures, thanks to its features. In particular, the automatic system management, automatic maintenance and automatic optimization provides time and cost advantages. For these reasons, ADW was preferred in the data lake implementation.

3.2.4.2 Data Transformation and Load

After the logical and physical design of the data model was created over ADW, the data brought from the data sources were loaded into the system. Since we built the system as a data lake, complex data transformations were not performed at the data loading stage. Simple transformations such as data type conversions, NULL value checks and data type checks have been made. The purpose of the simple transformations applied is to keep the data quality under control and to obtain data sets that can be used comfortably in subsequent analytical studies. The condition for building accurate analytical models is to work with a healthy and quality data sets.

As we mentioned in the previous sections, software developments that transfer data from sources to the platform during the extraction phase were made in the

python language. Since some of the data transformation is done during the extraction phase, the operations performed in this part are also developed in the python language. The other part of data transformation is performed in the data lake, so the technology used in this part is SQL.

3.2.5 Data Science Sandbox

We mentioned that data management platforms should provide an environment for developing artificial intelligence, data science and machine learning projects. In this component of the platform, an environment where data scientists and artificial intelligence experts can easily work on the collected data has been set up. There should be a lot of software that an expert needs while developing a project on this environment.

When we examine the development process of artificial intelligence, data science and data mining models (Figure 24), we see that these projects are developed in multiple steps. There are specific open source tools that are almost standardized in the industry and used in each of these stages [26]. Therefore, almost all of these tools must be installed in the data science sandbox.

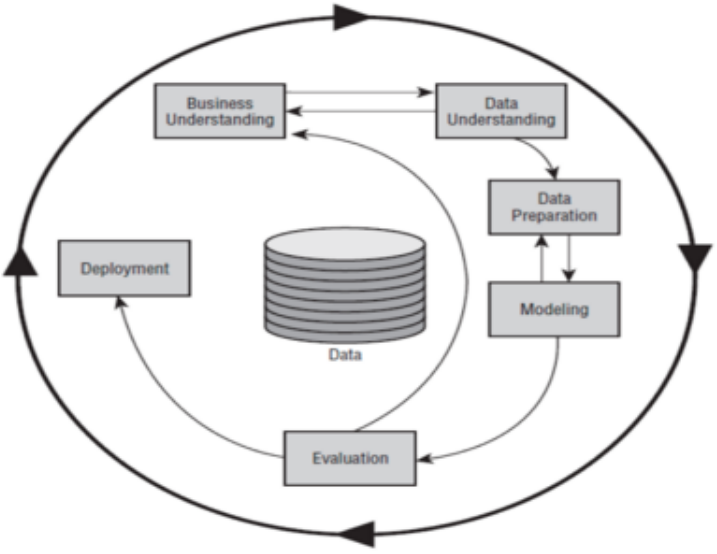

Figure 24: CRISP-DM Process Model [25]

When we examine the most common programming languages used in Data Science projects, we see that the python is the most popular programming language [27]. Second in the list is the R programming language [27]. Kaggle is the world's most popular platform for data science, machine learning and artificial intelligence. Data scientists can share their work on this platform. They can participate in competitions and share their experiences through Kaggle. When we examine this platform, we see that the projects are mostly developed using python. In addition, The open source python libraries and application development interfaces frequently used in these projects are as follows.

- **Development Environment:** Anaconda, Jupyter Notebook
- **Data Transformation:** Pandas, Dask, SciPy, NumPy, Numba, NLTK, Spark
- **Data Visualization:** Matplotlib, Seaborn, Plotly, Bokeh
- **Model Training:** Scikit Learn, Tensorflow, XGBoost, Keras, PyTorch, Prophet, Mxnet

It is very important to install all of the software listed above in the component to built. The fact that the environment is ready for use will make things easier for people who will work in this environment. Multiple methods can be used to provide this infrastructure. However, with the development of cloud computing, it is very advantageous to implement these environments with the services offered by cloud service providers. Thanks to these services, which are offered as fully pre-installed, data science project environments can be opened up to the service of multiple users very quickly. The platform services that popular cloud service providers have built for this purpose are as follows.

- **Amazon Web Services:** SageMaker

- **Microsoft Azure:** Azure Machine Learning Studio, Azure Databricks
- **Google Cloud Platform:** AI Hub
- **Oracle Cloud Infrastructure:** OCI Data Science Service

Almost all of the infrastructures offered by cloud providers are similar to each other and have standard features. Therefore, the Data Science Sandbox component can be easily implemented with all of these services. However, we thought it would be appropriate to implement this sandbox infrastructure with Oracle Cloud Infrastructure Data Science Service, since we implemented the data lake implementation with Oracle Cloud Infrastructure Autonomous Data Warehouse. Therefore, OCI Data Science service was preferred as a data science sandbox component.

Within the OCI Data Science service, besides the project development tools and libraries, there are different options that we can easily deploy the developed models on the Oracle Cloud platform. In addition, the platform's model explainability and model management capabilities are also features that can facilitate the work of data scientists. In addition, the installation of different open source libraries can be easily done with the terminal screen offered by the service (Figure 25).

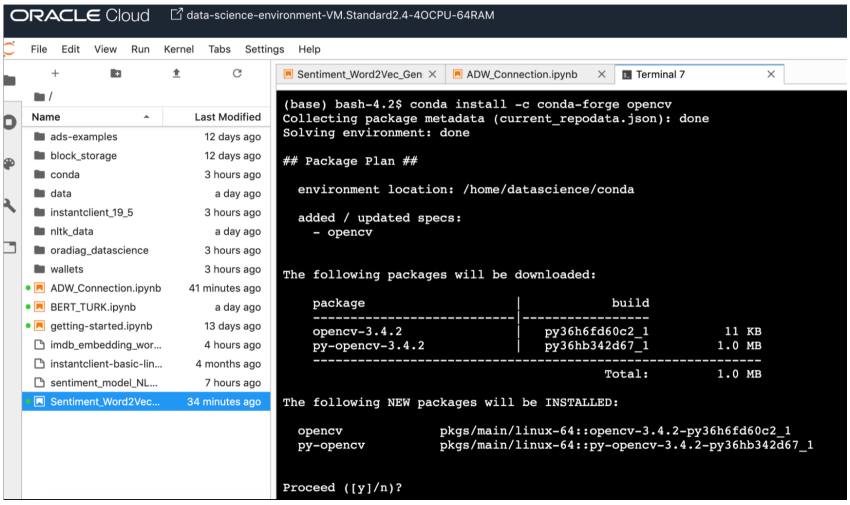

Figure 25: OCI Data Science Terminal View

3.2.6 Data Access

The data access layer is where data produced and managed in the central data management platform reaches consumers. In this layer, data produced by different tools can be served according to the needs of consumers. Line of business users, operational apps or analytical apps can be potential consumers. Therefore, central management platforms should include different technologies for each usage scenario at the data access layer.

Users connected to central management platforms mostly meet their data query and self-service reporting needs. Therefore, we can say that the most used technology in this layer is SQL. With the SQL language, users can access the data that produced within the platform. Therefore, it is one of the most important needs of today that central data platforms support SQL language for querying data.

We mentioned in the previous sections that data is the most valuable asset. In particular, data that has been processed and on which new values are derived has strategic importance for organizations. From this point of view, it is important for users of all levels to be able to report data stored on central data platforms with data visualization tools. Therefore, we can say that one of the most important parts of the data access layer is self-service BI and reporting tools. Today, almost every enterprise investing in central data management platforms uses products for their data visualization and professional reporting needs. Also, these products are known in the industry as business intelligence tools and the number of enterprise-scale products used in this area is quite high. This is because almost every enterprise needs these tools. Some of the most frequently used products in data visualization and reporting are Microstrategy, Qlick, Tableau, Oracle Analytic Cloud, Microsoft Power BI, SAP BO and IBM Cognos Analytics.

Since there is no need for reporting at enterprise scale on the platform we have built, a reporting tool has not been integrated directly into the system. However,

all of the reporting products we have listed above can be easily integrated into this platform. The only support we provide for accessing data managed on the platform is SQL. By connecting to the platform with SQL IDEs (e.g. SQL Beaver, SQL Developer, Toad, PL / SQL Developer), the requested queries can be run easily on the platform. In addition to supporting the ANSI SQL standard, the platform also brings many options that Oracle SQL offers.

3.2.7 Code Repository Management

Custom code developments have been made in many different layers within the platform. These developments cover from data models to data extraction codes and machine learning models. The management, storage and versioning of all the codes developed within the scope of the project are done on the repository created on Git. Git, produced by Linus Torvalds, the founder of Linux, is a speed-oriented and distribution-oriented control and source code management system.

All the codes and contents used within the scope of the development of the platform are kept in remote GitHub repositories using the GitHub infrastructure. The structure of the repository that we created within the scope of the project is as in Figure 26.

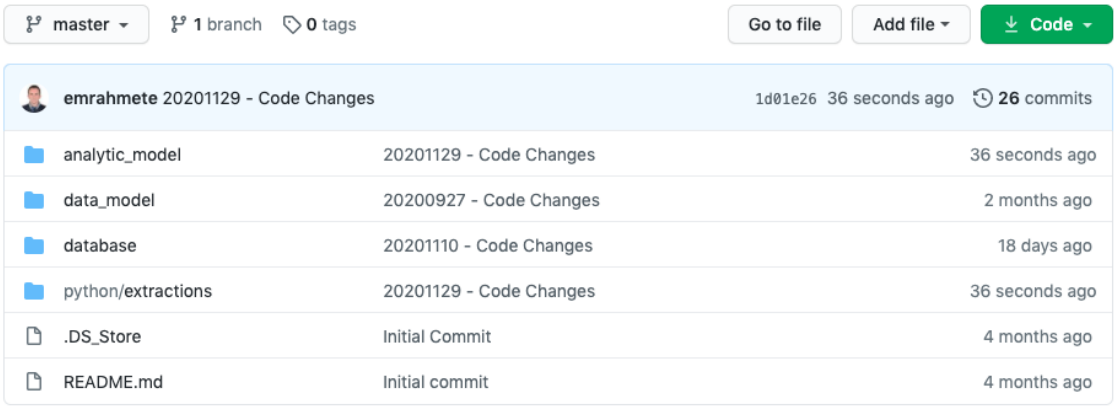

Figure 26: Github Project Repository Directory View

CHAPTER IV

ANALYTICAL MODEL DEVELOPMENT

We mentioned that data sources of the central data management platform in the previous sections. Also, we explained the dataset and its features elaborately. In this section, we will develop an analytical model using the data the we collected on the platform and examine the results.

4.1 Problem Definition

Football is the most popular and followed sports branch in the world. This interest in football has built an industry whose value is measured in billions of dollars. Undoubtedly, one of the most important issues in football is player transfers. Each football club transfers players in specified periods to strengthen their current staff. For this reason, thousands of players transfer from one club to another every year. Especially the transfers made in the major football leagues are very important news all over the world. When revenues and player profiles are examined among these leagues, it is seen that the English Premier League is the most valuable league (Figure 27) [28].

A player can be transferred from one team to another depending on many parameters. Undoubtedly, one of the most important variables that will determine the transfer is the performance of the player. Therefore, we can easily say that the season statistics of the player have an effect on his transfer. Based on this point, we will develop a predictive model and this model will be able to predict whether the player will be in the Premier League for next season, using end of season statistics of a player who playing in the Premier League in that season.

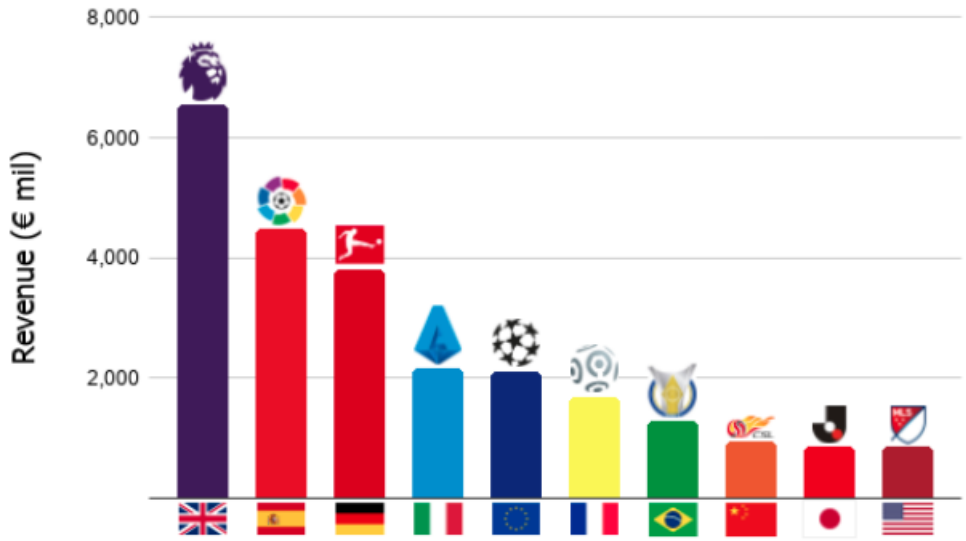

Figure 27: Top 10 Most Valuable Football Leagues in 2019 (by Revenue) [28]

4.2 Data Exploration

We extract many different data from sources to the central data platform. To solve the problem that we defined, we develop model by using player, player statistics and team datasets that collected on the platform. In this section, we get to know the data by examining the basic information of the datasets.

Some of the basic information about the data sets that we use in modeling are as in Table 4.

Table 4: Basic Information of Datasets

Dataset Name	**Number of Rows**	**Number of Columns**	**Date of Extraction**	**Description**
player	268964	13	2020-08	player information and demographics
player_stats	1583663	39	2020-08	season statistics of the player
team	14110	11	2020-08	team information

Table 5: Column Definitions of Player Dataset

Column Name	Number of Unique Values	Column Description
player_id	268964	player identifier
player_name	251940	player name
first_name	67178	player first name
last_name	153809	player last name
position	11	player position
nationality	237	player nationality
team_id	8241	player current team identifier
birth_date	11158	player birth date
birth_place	19107	player birth city
birth_country	242	player birth country
height	69	player height
weight	89	player weight
rating	2515	player rating

Table 6: Column Definitions of Team Dataset

Column Name	Number of Unique Values	Column Description
team_id	14110	team identifier
name	13987	team name
code	22	team code
country	249	country where the team is located
is_national	2	type of team national or club
founded	165	year of foundation
venue_name	7704	venue name
venue_surface	4	venue surface
venue_address	6407	venue address
venue_city	6196	city where the team venue is located
venue_capacity	1935	venue capacity

Table 7: Column Definitions of Player Statistics Dataset - 1

Column Name	**Number of Unique Values**	**Column Description**
player_id	268964	player identifier
team_id	8471	player team identifier
league	311	league
season	79	season
injured	1	how many times was injured in that season
rating	5042	rating of player in that season
captain	52	how many times was captain in that season
game_appearences	55	how many games did he play that season
games_minutes_played	4249	how many minutes did he play that season
games_lineups	65	how many games lineup did he play that season
substitutes_in	38	how many games did he substitutes-in in that season
substitutes_out	38	how many games did he substitutes-out in that season
substitutes_bench	54	how many games did he stay substitutes bench in that season
goals_total	46	how many goals did he score in that season
assists	22	how many assists did he make in that season
conceded	81	how many goals did he conceded in that season
saves	167	how many goals did he save in that season
passes_total	2021	how many passes did he pass in that season
key_pass	132	how many key passes did he pass in that season
pass_accuracy	93	what was the percentage of pass accuracy in that season
tackles_total	140	how many tackles did he make in that season
tackles_blocks	55	how many tackles did he block in that season

Table 8: Column Definitions of Player Statistics Dataset - 2

Column Name	**Number of Unique Values**	**Column Description**
tackles_interceptions	153	how many tackles did he intercept in that season
duels_total	699	how many duels did he do in that season
duels_won	388	how many duels did he win in that season
dribbles_attempts	228	how many dribbles did he attempt in that season
dribbles_success	149	how many dribbles did he success in that season
fouls_drawn	143	how many fouls did he drew in that season
fouls_committed	119	how many fouls did he commit in that season
shots_total	176	how many shots did he do in that season
shots_on	83	how many shots did he find target in that season
penalty_won	10	how many penalties did he win in that season
penalty_committed	8	how many penalties did he commit in that season
penalty_success	14	how many penalties did he success in that season
penalty_missed	5	how many penalties did he miss in that season
penalty_saved	7	how many penalties did he save in that season
yellow_card	26	how many yellow cards did he get in that season
yellowred	7	how many match did he get second yellow card in that season
red_card	8	how many matches did he get red card in that season

4.3 Data Preparation

It is not possible to directly use the data sets we examined in the previous section in the modeling phase. Therefore, some preparations should be made on these data sets before the modeling phase.

One of the first steps in the data preparation stage was to increase data quality. Especially, all features of 209222 records in player statistics data were found to be zero. First of all, these records were cleaned from the data set.

Since the model we will develop will predict the transfer movements in the English Premier League, the data set to be created for model training should be filtered through the whole cluster. Since the statistics of thirteen seasons in total will be used for model training, all teams that competed in the Premier League between the 2007-2008 and 2019-2020 seasons were determined using Transfermarkt website. These teams that found from web were written in another table for filtering data in the next phase.

```
CREATE TABLE tmp_premier_league_teams AS
SELECT * FROM team WHERE
country IN ('England','Wales') AND
name IN (Hull City','West Brom','Stoke City','Burnley','Wolves',
        'Blackpool','Swansea','Norwich','QPR','Southampton','Cardiff',
        'Crystal Palace','Leicester','Watford','Bournemouth','Brighton',
        'Huddersfield','Sheffield Utd','Manchester United','Chelsea',
        'Arsenal','Liverpool','Everton','Aston Villa','Blackburn',
        'Portsmouth','Manchester City','West Ham','Tottenham',
        'Middlesbrough','Wigan','Sunderland','Bolton','Fulham','Reading',
        'Birmingham','Derby','Newcastle');
```

Using the season and team columns on the player statistics dataset, players who

played in the Premier League in the relevant seasons were found.

```
SELECT DISTINCT player_id
FROM player_stats t WHERE
season IN ( '2007-2008','2008-2009','2009-2010','2010-2011',
            '2011-2012','2012-2013','2013-2014','2014-2015',
            '2015-2016','2016-2017','2017-2018','2018-2019',
            '2019-2020') AND
team_id IN (SELECT team_id FROM tmp_premier_league_teams);
```

In the player statistics data set, there can be more than one record of a player's performance for a season. The different league and cup matches played by the player during the season are shown on separate lines in the dataset. Since the only data we want to use in modeling will be the Premier League statistics of the player, other leagues and cups are filtered from the dataset.

```
SELECT * FROM player_stats t
league NOT IN ('Cup','Coppa Italia','League Cup','Copa del Rey',
'UEFA Europa League','FA Cup','UEFA Champions League','Play-offs 1/2',
'UEFA Nations League','DFB Pokal','Coupe de la Ligue','UEFA Super Cup',
'Schweizer Pokal','KNVB Beker',CONCACAF Champions League','Cupa',
'Football League Trophy','Non League Div One','Magyar Kupa',
'Welsh Cup','CAF Champions League','CAF Super Cup','Copa MX',
'Challenge Cup','FA Trophy','Cupa României','David Kipiani Cup',
'Scottish Cup','Coupe de France','Irish Cup','Presidents Cup',
'Coppa Titano','Copa Chile','Suomen Cup','Super Cup','Svenska Cupen',
'Taça de Portugal','Recopa Sudamericana','CAF Confederation Cup',
'Coupe Nationale','Copa Venezuela','Toto Cup Ligat Al');
```

The knowledge of which league and which club the footballer plays in the next season is a very important detail to create our target variable. For this reason, the information in which league and team the footballer played in the next season was added to each row in the data set containing football player statistics. After this information is obtained, the value of the target variable was calculated for each record. If the footballer stayed in the Premier League the next season, the target variable is set to 0, if the footballer is transferred to another league, the target variable is assigned to 1. Considering the values we assign to the target variable , our problem is binary classification.

In the dataset, nationality and position columns are categorical fields. Label encoding was applied to these features in order to feed them into the model. In addition, the games_lineups column was discretizationed and divided into buckets. New category labels were assigned to each new bucket. Besides, the age column of sixty-three records was null. The most frequent value in age column in the entire dataset (27) was assigned to the age columns of these records.

The final data set is ready for the next phase after filtering and data cleaning. In the final data set obtained, there are 5446 records belonging to thirteen seasons in total and there are 40 features of each record.

4.4 Feature Engineering

In this section, new features have been produced to enrich the data set and make it ready for modeling. While producing new features, all career data of the players was processed. Details of new features that produced using existing datasets are as follows.

- **lastSeasonPL:** A value of 1 is assigned if he played in the Premier League the previous season, and 0 if not.
- **lastSeasonStats:** How many game minutes did he played in the previous

season in Premier League? If the player has not played in the premier league the previous season, this variable is assigned -1.

- **plExperience:** How many seasons of Premier League experience does a player have until the current season?

- **teamTenure:** How many seasons he played in total for the last team that he played?

- **statsChange:** The game minutes played of the previous season and current season was compared for each player. Then, the determined rules were applied to the result that obtained in previous step. The final values to be assigned to this variable were calculated by looking at the rules shown in Table 9.

Table 9: Rule Table of statsChange

Rule	**Value**
-%20 < difference < %20	0
difference < -%20	-1
difference > %20	1

- **last3SeasonTeamChange:** First, it was calculated how many different teams the player had changed in the last three seasons. While calculating this change, all team changes in the player's career were examined. Then, the number of changes obtained was passed through the determined rules and the final value of the variable was reached. The values to be assigned to this variable have been calculated by looking at the rules shown in Table 10.

Table 10: Rule Table of last3SeasonTeamChange

Rule	**Value**
#change $\geq$ 2	1
#change < 2	0

With the new features obtained during the feature engineering phase, 45 features

and 1 target variable were created in the data set. After this stage, we will produce prediction models based on these variables and examine their results.

4.5 Modelling

In the modeling phase, we created models using different algorithms with the final training set we obtained. We optimized the models by examining the outputs of the models we created.

Before starting modeling, it was understood whether the dataset is unbalanced by examining the distribution of the target variable (Figure 28). As a result of this examination, it was decided not to apply any sampling method on the dataset.

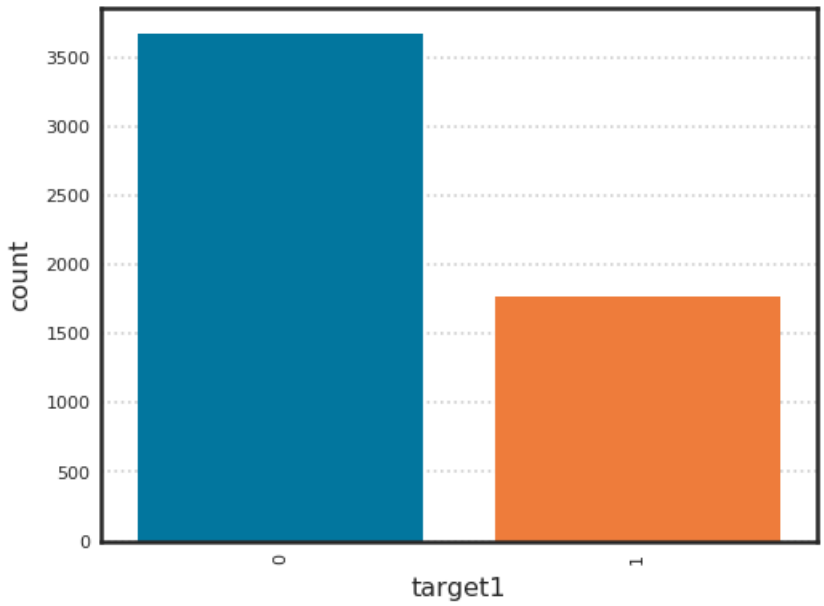

Figure 28: Class Distributions of Target Variable

After examining the distribution of the target variable in the dataset, the relationships between the variables produced in the feature engineering phase and the target variable were examined (Figure 29, Figure 30, Figure 31, Figure 32, Figure 33, Figure 34).

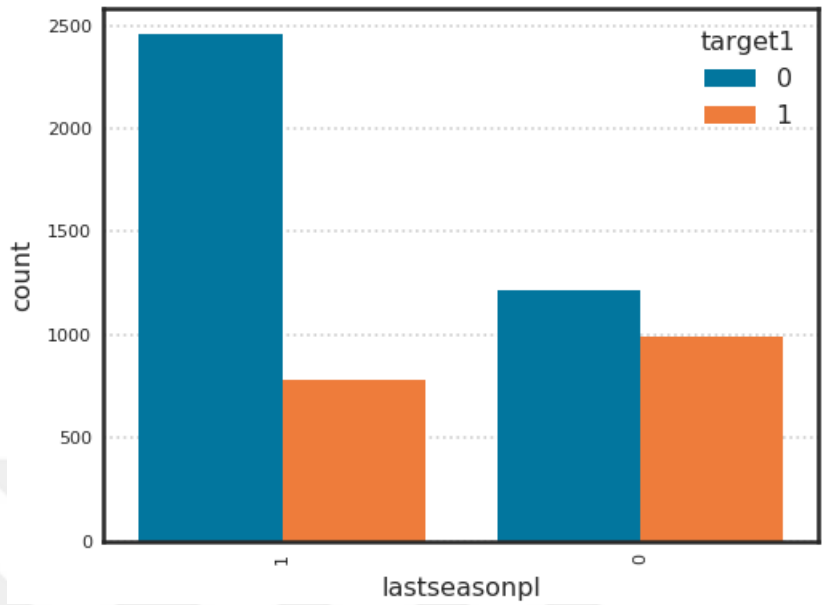

Figure 29: Class Distributions between Target Variable and lastseasonpl

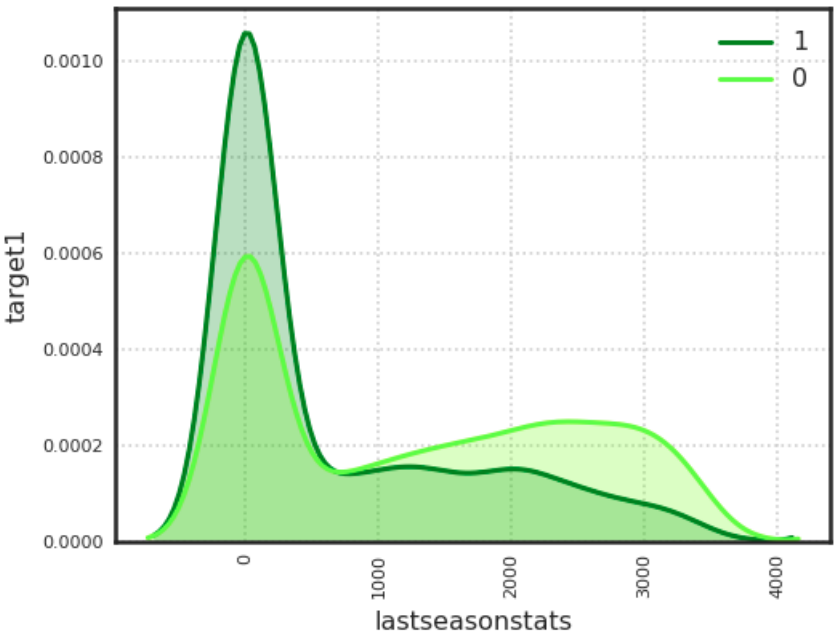

Figure 30: Class Distributions between Target Variable and lastseasonstats

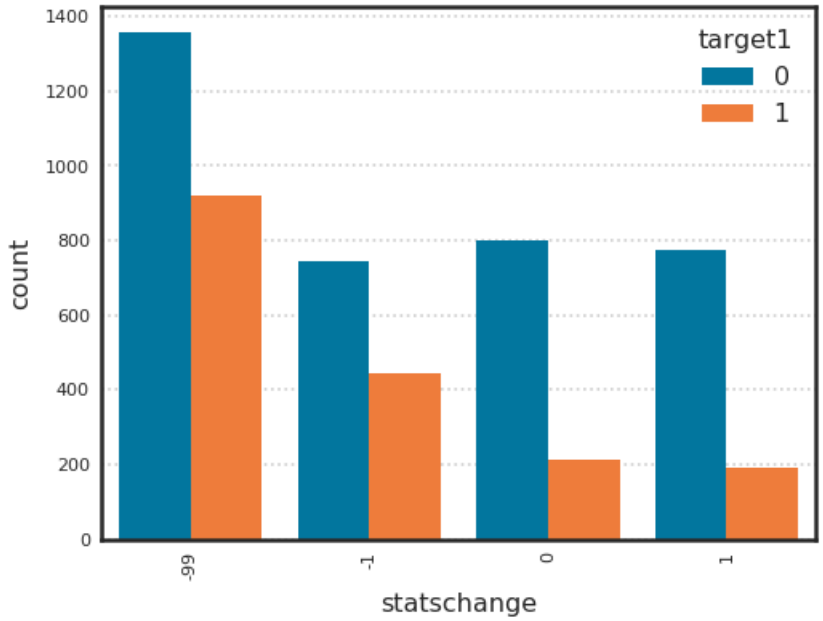

Figure 31: Class Distributions between Target Variable and statschange

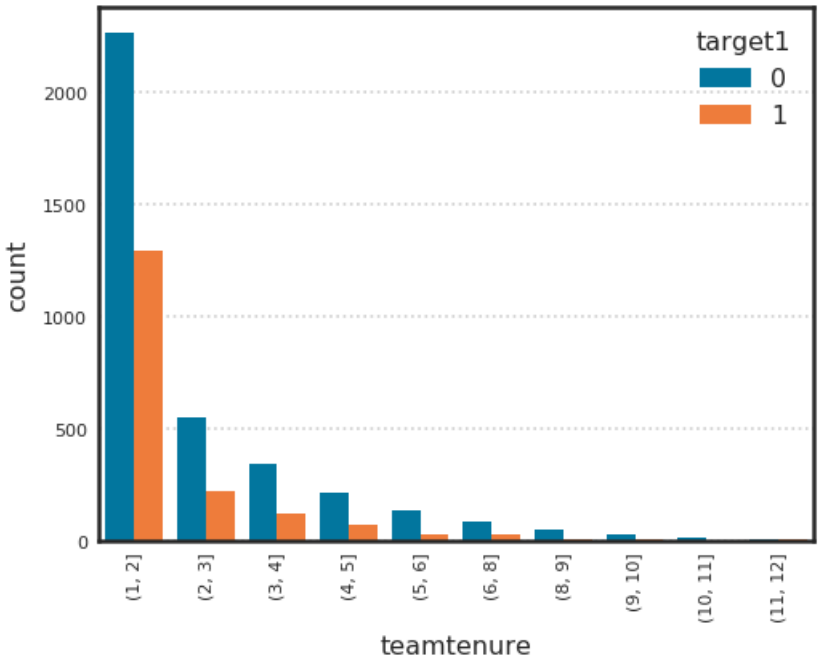

Figure 32: Class Distributions between Target Variable and teamtenure

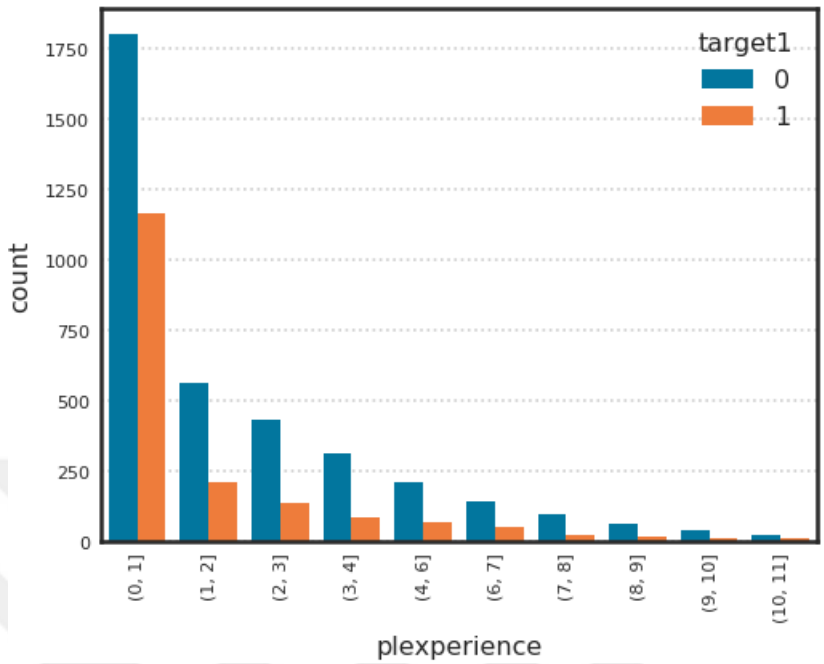

Figure 33: Class Distributions between Target Variable and plexperience

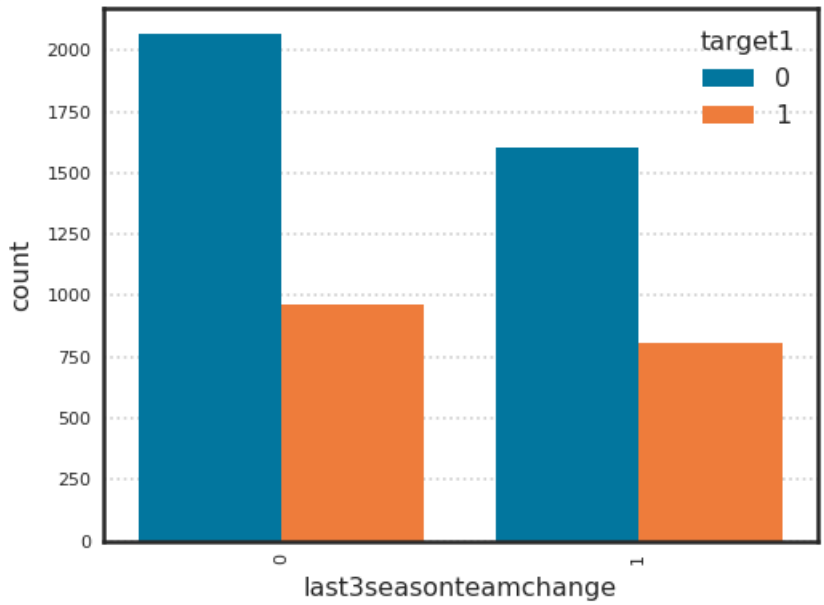

Figure 34: Class Distributions between Target Variable and last3seasonteamchange

When the generated graphics were examined, it was decided to use the variables that generated in feature engineering, considering that they could affect the modeling.

Feature importance models help us better understand the dataset and identify which features are relevant. Thus, it makes our job easier when choosing the features we will use in the modeling phase. At this stage, the feature importance model was developed with a tree-based approach (ExtraTreesClassifier was used as a model) and the importance values of the features were found. Feature importance results are shown in Figure 35.

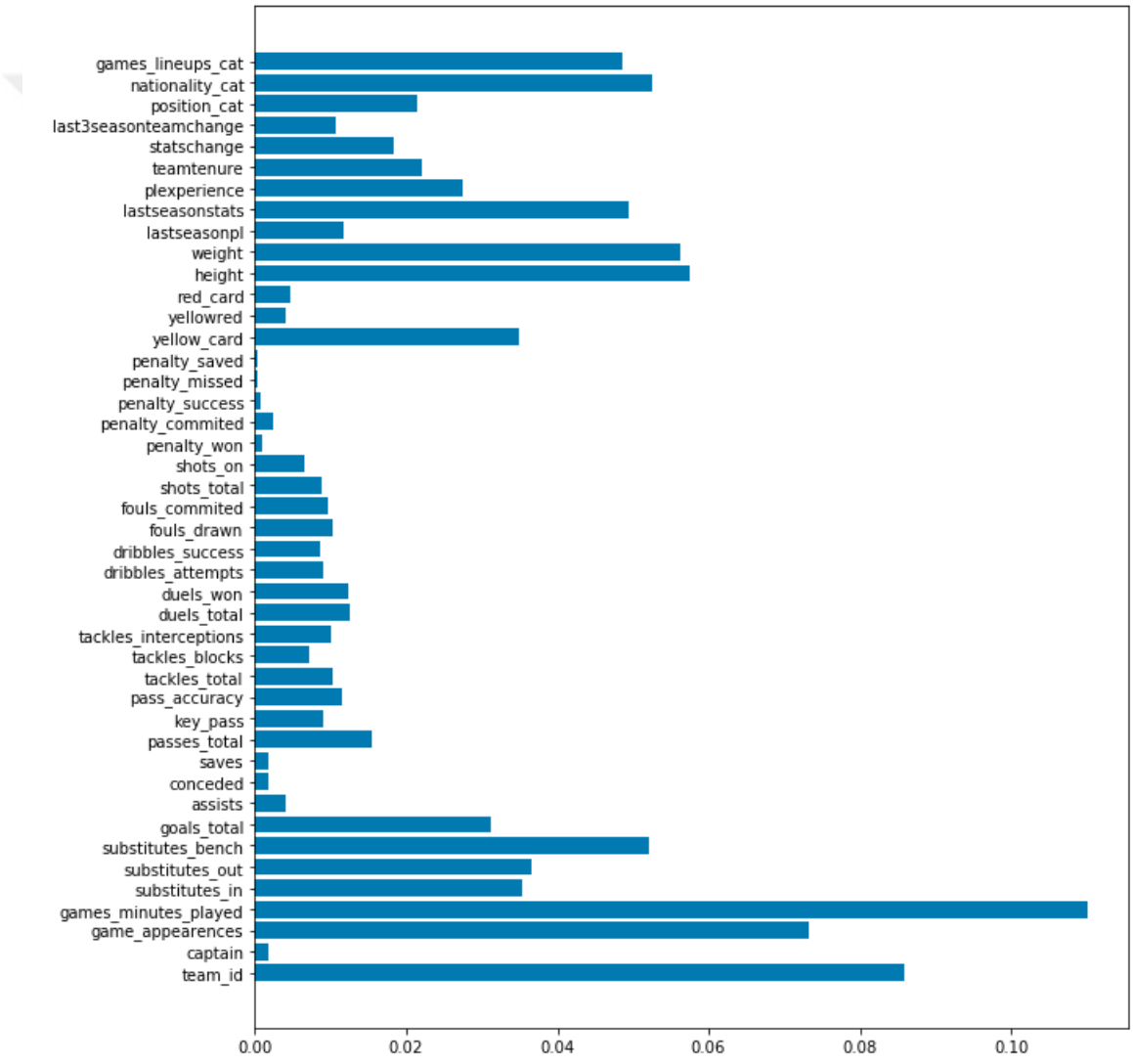

Figure 35: Results of Feature Importance Model

4.5.1 Experiments and Results

In this section, predictive models were created using different feature subsets, different algorithms. Then the results were examined in detail.

4.5.1.1 Experiment - 1

In the first experiment, models were created with seven different algorithms using all the features. The hyper-parameters of the classifiers used in the models were used with their default values. In addition, the data set is divided into 80% training and 20% test. The test results of the models used are as in Table 11. ROC curve is shown in Figure 36.

Table 11: Experiment - 1: Evaluation Metrics with Testing Data

	accuracy	**precision**	**recall**	**f1**	**auc**
XGboost	0.7679	**0.7198**	0.4704	0.569	0.6910
LogisticRegression	0.745	0.6421	0.4901	0.5559	0.6791
SVM	0.6862	0.76	0.05352	0.1	0.5227
LGBMClassifier	**0.7789**	0.7143	**0.535**	**0.6119**	**0.7159**
DecisionTreeClassifier	0.6853	0.517	0.5127	0.5149	0.6407
BaggingClassifier	0.7459	0.6585	0.4563	0.5391	0.671
KNeighborsClassifier	0.6651	0.483	0.4	0.4376	0.5966

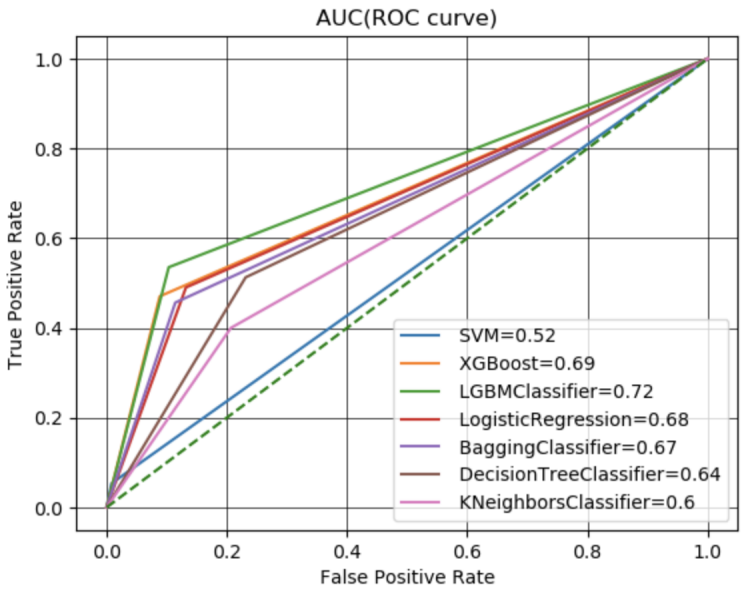

Figure 36: AUC(ROC Curve) of Experiment-1

4.5.1.2 Experiment - 2

In the second experiment, the features to be included in the model were reduced by examining the Feature Importance results (Figure 35). Feature selection was made by deciding on sixteen features in total. With the new feature set, the same models were retrained. The features selected for use in Experiment-2 are listed below.

```
features = ['games_minutes_played','game_appearences','team_id',
'last3seasonteamchange','saves','games_lineups','substitutes_out',
'substitutes_in','teamtenure','lastseasonstats','lastseasonpl',
'nationality','statschange','plexperience','red_card','position']
```

In addition, the data set is divided into 80% training and 20% test. The test results of the models used are as in Table 12. ROC curve is shown in Figure 37.

Table 12: Experiment - 2: Evaluation Metrics with Testing Data

	accuracy	precision	recall	f1	auc
XGboost	0.767	0.7205	0.4648	0.5651	0.6889
LogisticRegression	0.7275	0.6272	0.4028	0.4906	0.6436
SVM	0.6872	**0.7692**	0.0563	0.105	0.5241
LGBMClassifier	**0.7844**	0.729	**0.538**	**0.6191**	**0.7207**
DecisionTreeClassifier	0.6844	0.5155	0.5155	0.5155	0.6407
BaggingClassifier	0.7303	0.6255	0.4282	0.5084	0.6522
KNeighborsClassifier	0.6789	0.5097	0.369	0.4281	0.5988

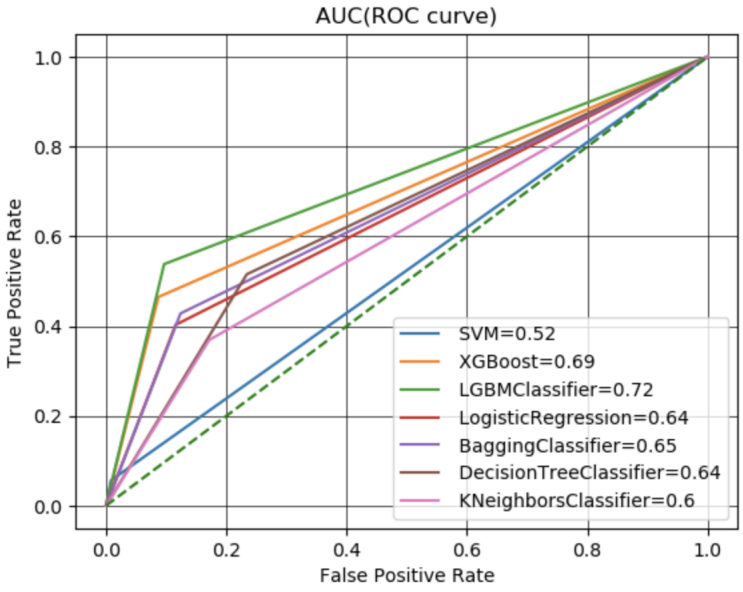

Figure 37: AUC(ROC Curve) of Experiment-2

CHAPTER V

CONCLUSION

Data is becoming the most important asset. Therefore the need for data management platforms is increasing day by day. Every enterprise that manages the data it collects correctly and produces new values over the data will provide an advantage in the competition. At the same time, they will make innovations using these value. However, data diversity and data velocity reaching the upper limits makes data management difficult. At this point, it is very important to analyze modern data management architectures and to choose the right components according to the needs.

In this study, first of all, current data management needs are defined. Then, modern data architectures that meet current needs and the components of these architectures were examined. In addition, it was mentioned what kind of studies should be done in response to changing data formats and data velocity. With this study, we have implemented a prototype of modern data management platforms, which we have explained over reference architectures, through an example use case.

Each layer of the data management platform implemented as a prototype was defined in detail. It is aimed to transfer the know-how about this subject by providing the necessary information about why the technologies used in each stage are selected. The developed data management platform was made by using the services offered by cloud providers. Thus, all the benefits of cloud computing are mentioned in the study. It should not be forgotten that cloud computing is increasing its popularity and accessing modern architectures via cloud platforms is faster and easier.

We conducted an analytical study using the data collected on the platform. With this study, we showed that data management platforms can also address advanced

analytic workloads. The dataset that we collected covered many topics related to football. With this data, we have developed different models that predict whether the players in the English Premier League will be in the league in the next season.

We recorded the results by running the analytical models we developed with different parameters. When we examined the model results, we observed that the best results were obtained with ensemble algorithms. LGBM and XGBoost algorithms produced better results than other methods. The limited data set have caused overfitting problems in methods such as SVM. The k-Nearest Neighbor algorithm, one of the item based learning methods, also failed to produce a successful result. Also, the feature selection study conducted before model training did not have a major effect on the results. As a result, feeding the models with more data will not only improve the results, but also allow the testing of algorithms with higher learning capacity. Therefore, finding more data will improve the work here.

APPENDIX A

SOME ANCILLARY STUFF

All the codes and designs that we have developed in this study can be accessed via the GitHub repository below.

`https://github.com/emrahmete/data_management_platform.git`

Bibliography

[1] P. Strengholt, *Data Management at Scale*. Sebastopol, California: O'Reilly, 2020.

[2] S. Gupta and V. Giri, *Practical Enterprise Data Lake Insights*. New York, NY: Apress, 2018.

[3] "What is Data Management." [Online] Available: `https://www.oracle.com/database/what-is-data-management/`. Accessed: 2020-08-15.

[4] "The top 5 data management challenges and how to overcome them." [Online] Available: `https://www.bba.org.uk/news/insight/the-top-5-data-management-challenges-and-how-to-overcome-them`. Accessed: 2020-08-20.

[5] F. Almeida and C. Calistru, "The main challenges and issues of big data management," *International Jounal of Research Studies in Computing*, vol. 2, no. 1, pp. 11–20, 2013.

[6] Oracle, "Information management and big data," tech. rep., Oracle, Redwood Shores, California, September 2014.

[7] W. H. Inmon, *Building the Data Warehouse*. Wiley, 2005.

[8] "What is Data Lake." [Online] Available: `https://aws.amazon.com/big-data/datalakes-and-analytics/what-is-a-data-lake/`. Accessed: 2020-09-20.

[9] "Data Lake." [Online] Available: `https://shortbread.io/experiences/datalake.html`. Accessed: 2020-09-25.

[10] "The Data Reservoir: Architecture, Best Practices and Governance." [Online] Available: `https://www.ibmbigdatahub.com/blog/building-data-reservoir-use-big-data-confidence`. Accessed: 2020-09-12.

[11] "What is Data Warehouse." [Online] Available: `https://www.oracle.com/tr/database/what-is-a-data-warehouse/`. Accessed: 2020-09-11.

[12] "Database Data Warehousing Guide." [Online] Available: `https://docs.oracle.com/database/121/DWHSG/concept.htm`. Accessed: 2020-09-11.

[13] "Data Warehousing Concepts and Design Guide." [Online] Available: `https://docs.oracle.com/database/121/TDPDW/tdpdw_arch.htm`. Accessed: 2020-09-15.

[14] "Data Warehouse Guide." [Online] Available: `https://panoply.io/data-warehouse-guide/data-warehouse-vs-data-lake/`. Accessed: 2020-10-02.

[15] "Definitive Guide to Cloud Data Warehouses and Cloud Data Lakes." [Online] Available: https://www.talend.com/resources/data-lake-vs-data-warehouse. Accessed: 2020-10-03.

[16] "What is Machine Learning? A definition." [Online] Available: https://expertsystem.com/machine-learning-definition/. Accessed: 2020-10-05.

[17] E. Alpaydin, *Introduction to Machine Learning*. Adaptive Computation and Machine Learning, Cambridge, MA: MIT Press, 3 ed., 2014.

[18] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–503, 2016.

[19] X. HE, N. Chaney, M. Schleiss, and J. Sheffield, "Spatial downscaling of precipitation using adaptable random forests," *Water Resources Research*, vol. 52, 10 2016.

[20] "Enterprise data warehousing - an integrated data lake example." [Online] Available: https://docs.oracle.com/en/solutions/oci-curated-analysis/index.html. Accessed: 2020-10-10.

[21] "Data Lake." [Online] Available: https://azure.microsoft.com/en-us/solutions/data-lake/. Accessed: 2020-10-10.

[22] "Data lake on AWS." [Online] Available: https://aws.amazon.com/solutions/implementations/data-lake-solution. Accessed: 2020-10-11.

[23] "Cloud Storage as a data lake." [Online] Available: https://cloud.google.com/solutions/build-a-data-lake-on-gcp. Accessed: 2020-10-11.

[24] "Autonomous Data Warehouse." [Online] Available: https://www.oracle.com/autonomous-database/autonomous-data-warehouse/. Accessed: 2020-10-15.

[25] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "Crisp-dm 1.0 step-by-step data mining guide," tech. rep., The CRISP-DM consortium, August 2000.

[26] P. Barlas, I. Lanning, and C. Heavey, "A survey of open source data science tools," *International Journal of Intelligent Computing and Cybernetics, http://www.emeraldinsight.com/doi/abs/10.1108/IJICC-07-2014-0031*, vol. 8, 06 2015.

[27] "Top Programming Languages for Data Science in 2020." [Online] Available: https://www.geeksforgeeks.org/top-programming-languages-for-data-science-in-2020/. Accessed: 2020-10-31.

[28] "Top 10 Most Valuable Football Leagues in 2019." [Online] Available: https://football-tribe.com/malaysia/2020/01/25/top-10-most-valuable-football-leagues-in-2019/. Accessed: 2020-11-05.

[29] W. Xu, L. Ning, and Y. Luo, "Wind speed forecast based on post-processing of numerical weather predictions using a gradient boosting decision tree algorithm," *Atmosphere*, vol. 11, p. 738, 07 2020.

VITA

Emrah Mete received his B.Sc. degree in Computer Engineering from Yildiz Techinal University in 2011. He has been working more than ten years and prior role was Senior Data Analytic Developer at Turkcell Technology Research and Development A.S,. Emrah is working as an Enterprise Cloud Architect at Oracle. His research interests include data science, cloud platforms, data management, machine learning and deep learning.