

# **CSP-554 – BIG DATA TECHNOLOGIES PROJECT REPORT**

---

## **FIFA 18 PLAYER Performance Prediction**

### **Submitted by:**

Shubham Modi - A20494476

Sai Reshma Guntimadugu - A20521853

Praveen Gorikapudi - A20522179

### **Submitted to:**

Joseph Rosen

## **ABSTRACT:**

The FIFA 18 Player Performance Prediction dataset is a comprehensive dataset containing various statistics related to football player performance in the FIFA 18 game. With the exponential increase in the size of the dataset, the traditional methods of analyzing data have become insufficient. The need for efficient and scalable big data technologies has increased tremendously in recent years. Apache Hadoop and Apache Spark are some of the most widely used big data technologies for processing large volumes of data. In this abstract, we propose the use of these technologies for analyzing the FIFA 18 Player Performance Prediction dataset. We discuss the challenges associated with the dataset and demonstrate how big data technologies can be used to preprocess, analyze, and model the data. We also explore the potential applications of this dataset, such as player performance prediction, team building, and player scouting, which can greatly benefit from big data analytics. Finally, we discuss the importance of utilizing big data technologies for analyzing large datasets, and how it can lead to more accurate predictions and better decision-making in the sports industry. This abstract highlights the significance of big data technologies in analyzing the FIFA 18 Player Performance Prediction dataset, and how it can provide valuable insights into football player performance.

## **1. INTRODUCTION:**

The FIFA 18 Player Performance Prediction dataset is a collection of data related to the performance of professional football players during the 2017-2018 season. The dataset includes information on players physical attributes, playing position, team, and individual statistics such as goals scored, assists, and successful passes. In this project, big data technologies will be used to analyze the dataset and develop predictive models to forecast player performance based on their attributes and past performances. The project aims to leverage the power of big data technologies such as PySpark to handle large volumes of data and perform efficient and scalable data processing, data analysis, and machine learning.

The project will involve data cleaning and preprocessing, exploratory data analysis, feature engineering, model training, and model evaluation. Various machine learning algorithms such as logistic regression, Naive Bayes, etc. will be implemented to develop accurate and reliable prediction models. The results of the project can be used by football clubs, coaches, and fans to gain insights into player performance and make informed decisions regarding player selection and team strategy.

### **Project Tools:**

- Apache Spark
- Jupyter
- Python

## **2. DATA OPERATIONS AND PROCESS FOR ANALYSIS:**

- **ETL pipeline:** Extract data from the above-mentioned datasets and pass them through the ETL pipeline using steps to clean any anomalies in the text, setting all values in a number format, and fixing all incorrect values and missing values.
- **Data Analytics:** We performed different operations on the fields to get the required analysis. We studied the data and transformed the data into graphs. We studied the graphs and made an analysis of all the data.
- **Create a Model:** Once the data has been collected, and transformed and the features have been designed, the next step is to create a Machine learning model using techniques like Random Forest, Logistic regression, Gradient Boosting, etc. where this machine learning model is trained by fitting the model to the data using the collected data and features and tested to predict the player preferences and understand the working of Machine learning models and their accuracy/ F-1 Score.
- **Data Visualization:** Visualize the players performance, the relationship between player attributes such as speed, agility, and shooting accuracy with their overall performance rating in the game. Compare the performance of different players over time or across different leagues and tournaments. Using machine learning algorithms to predict future performance and visualize the predictions along with the actual results.
- **Making Predictions:**  
Pick out the best players based on the overall rating.  
Which type of offensive players tend to get paid the most.  
Top 5 players for different positions in terms of overall as well as potential points.

## **3. COMPARISON OF TOOLS:**

### ***Hive:***

Hive<sup>[41]</sup> is a data warehousing tool that allows users to query and analyze large datasets stored in Hadoop's distributed file system using SQL-like queries. Hive provides a higher-level abstraction over Hadoop's MapReduce framework and allows users with SQL knowledge to work with big data.

### **Features:**

- SQL-like query language (HiveQL) for querying and analyzing data.
- Schema-on-read for handling unstructured data.
- Integration with the Hadoop ecosystem including HDFS, MapReduce, and YARN.
- User-defined functions and custom input/output formats for data processing.
- Supports ACID transactions for the update, insert, and delete operations (in Hive 0.14+).

---

1 "Spark vs Hive - What's the Difference - ProjectPro." <https://www.projectpro.io/article/spark-vs-hive/480>. Accessed 3 May. 2023.

**Benefits:**

- Provides a SQL-like interface to Hadoop's distributed file system, making it easier for users familiar with SQL to work with big data.
- Supports a wide variety of data formats and can handle both structured and unstructured data.
- High-level abstraction over Hadoop's MapReduce framework, making it easier to create data flows and pipelines for processing data.
- Provides an easy way to summarize, query, and analyze large datasets.
- Supports ACID transactions for some operations.

**Drawbacks:**

- Not suitable for real-time processing of data.
- Hive's SQL-like queries may not be as efficient as traditional SQL for some use cases.
- Limited support for transactional processing compared to traditional relational databases.

***Hadoop:***

Hadoop<sup>[52]</sup> is an open-source distributed processing framework that provides scalable and fault-tolerant storage and processing of large datasets. It is a batch-processing system designed to handle massive amounts of structured and unstructured data.

**Features:**

- Distributed file system (HDFS) for storing large datasets across multiple nodes.
- MapReduce processing framework for batch processing of large datasets.
- YARN for resource management and job scheduling.
- Hadoop Common for providing libraries and utilities for Hadoop modules.
- Hadoop Distributed Data Store (HBase) for real-time read/write access to big data.
- Pig for creating data flows and pipelines for processing data.
- Zookeeper for distributed coordination and synchronization of the Hadoop cluster.

**Benefits:**

- Scalable and fault-tolerant storage and processing of large datasets.
- Cost-effective compared to traditional data warehousing solutions.
- Support for a wide variety of data formats including structured and unstructured data.
- Flexibility to handle various use cases such as batch processing, data warehousing, and real-time data processing.
- Highly customizable and can integrate with other tools and technologies.

---

<sup>2</sup> "What is Hadoop? - Amazon AWS." <https://aws.amazon.com/emr/details/hadoop/what-is-hadoop/>. Accessed 3 May, 2023.

**Drawbacks:**

- Requires significant hardware resources and technical expertise to set up and maintain.
- Not suitable for real-time processing of data.
- High overhead for small datasets.
- Complex infrastructure to manage and monitor.

***Pyspark:***

Pyspark<sup>[33]</sup> is a Python-based API for Spark, a distributed computing framework that provides fast data processing, machine learning, and graph processing capabilities. Pyspark provides an easy-to-use interface for writing data processing workflows in Python.

**Features:**

- Distributed computing framework for processing large datasets in parallel across multiple nodes.
- In-memory processing using Spark's Resilient Distributed Datasets (RDDs) for faster processing.
- Machine learning and graph processing libraries for advanced data analysis.
- Python-based API for writing data processing workflows.
- Supports real-time and batch processing of data.

**Benefits:**

- Provides fast data processing and analysis using distributed computing and in-memory processing.
- Supports various data sources and formats, including structured and unstructured data.
- Easy-to-use Python-based API that is familiar to many data scientists and developers.
- Supports real-time and batch processing of data, making it suitable for various use cases.
- Provides machine learning and graph processing libraries for advanced data analysis.

**Drawbacks:**

- May require significant hardware resources and expertise to set up and maintain.
- Not as mature as Hadoop or Hive in terms of enterprise support and ecosystem.
- Can have a steep learning curve for users new to distributed computing.
- Performance can be impacted by network latency and other factors.

---

<sup>3</sup> "Scala vs. Python for Apache Spark - ProjectPro." 24 Apr. 2023, <https://www.projectpro.io/article/scala-vs-python-for-apache-spark/213>. Accessed 3 May. 2023.

#### 4. DATASET EXPLANATION:

This dataset<sup>4</sup> contains a list of all soccer players from around the world along with their attributes which depict the skills of every individual player during the Fifa 2017 world cup. The first column shows the name of the player. The next few columns provide us with the demographic information for the player namely - Nationality, the club the player plays for, etc.

The “Overall” column provides us with a value between 0 - 100 and gives us a fair idea of how the player's performance is on average in all his skills with 100 being the highest and 0 being the lowest.

The next 39 columns like aggression, dribbling, agility, Ball Control, passing, etc. give us a numerical value representation between 0 - 100 for each player which allows us to understand the skill of the player for one particular attribute.

The Position column gives us details about what position the player plays in a team. Look at the image below to understand more about the position of the players.



<sup>4</sup> "FIFA 18 PLAYER Performance PREDICTION - Kaggle." <https://www.kaggle.com/datasets/edith2021/fifa-18-player-prediction>. Accessed 3 May. 2023.

## **5. OBSERVATION AND ANALYSIS:**

- **Data Preparation and Analysis**

Before working on the data we worked on analyzing the data and preparing the data for the next steps. After evaluating the dataset we concluded this step with the approach to perform data cleaning before analyzing the data and making predictions. Changing the Excel file to CSV for better read / write speeds and maximum compatibility was our initial step. More details about cleaning are discussed in the Data Cleaning section of this report.

- **Data Cleaning**

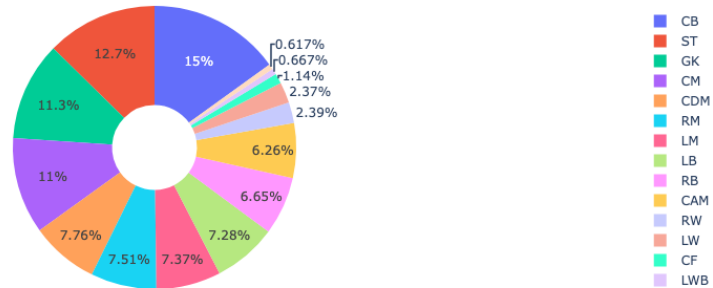
In this step, we performed data cleaning on the entire dataset. Few numerical columns contained ASCII characters, for example, the “wage” column contains values as “ 900 M” so as a part of the cleaning process we had to eliminate the non-numerical characters and changed the column name to “Wage in Millions” to match the column attribute value. Similarly, we cleaned the data for the “Value” column in the dataset and changed it to “Value in Thousands”.

As we know, in the world of football, the performance of a player keeps on changing dynamically throughout the season and that has to be kept track of for better predictions. The values in the columns containing attributes of individual players were modified to show the numerical change in the skill. This resulted in some values being displayed as an expression. For example, the passing skill of some players which was improved is noted as “80+5” or for some players, if the shooting accuracy has decreased then the same was shown as “90 - 4”. We made use of UDF functions in the spark Dataframes to evaluate the expressions and calculate the updated attribute value.

The columns in the spark dataframe were set to StringType when the read csv function was used. We updated the data type of columns to appropriate data types to represent numerical values.

While training our models with this dataset to predict the position of the player based on its attributes, we understood that the model was performing poorly due to a lack of data points to train the model. The visualization below gives us a better understanding of the data distribution based on the labels.

Plotting Label Distribution of Complete Dataset



As we can see there are very few data points in the dataset when split by labels with the lowest being 111 rows for the Right Wing Back (RWB) position. This resulted in the training data being skewed and biased towards the label Center Back (CB) which has the highest number of rows with 2705 players.

In order to increase the accuracy of the models, we decided to merge similar labels and decided on the following mapping.

defenders = LWB, RWB, CB, LB, RB, GK

midfielders = LM, CM, RW, CDM

strikers = LW, CAM, CF, ST

The logic behind this mapping is that all the defenders have the same set of attributes i.e. higher defending skills and lower shooting accuracy, lower pace except wing backs, etc.

This logic fits perfectly for midfielders and strikers with higher passing accuracy, higher chance creation, dribbling, vision, etc., and higher agility, finishing, etc respectively.

After merging the labels according to the mapping above the dataset was fairly balanced as displayed before and the machine learning models were performing relatively well.

Plotting Label Distribution of Updated Dataset



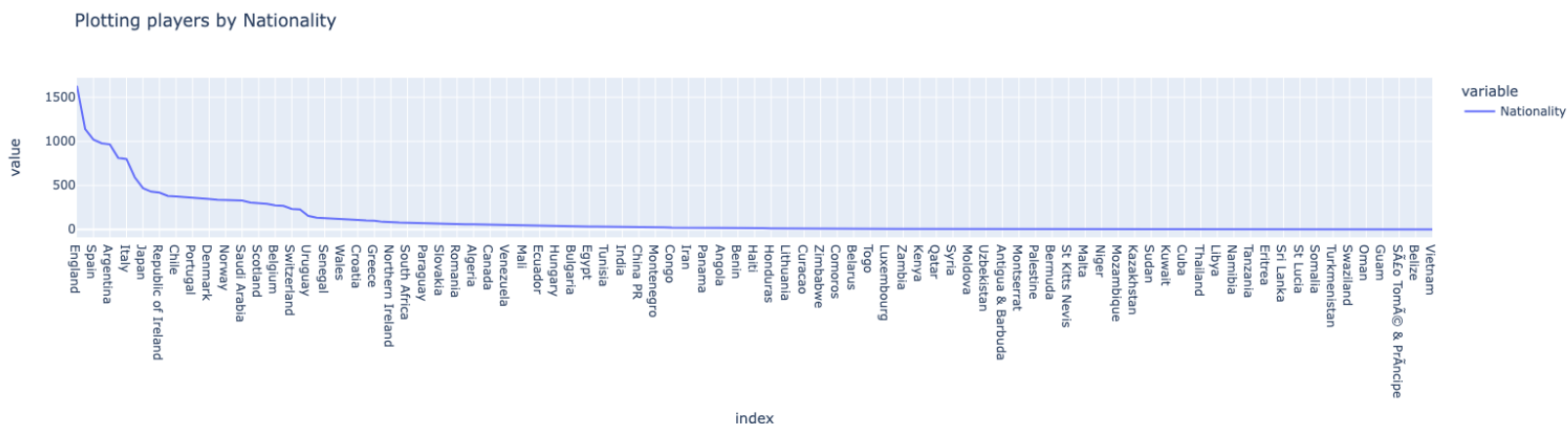


- **Data Visualization**

To Understand the dataset better we used visualization techniques and implemented a few plots as shown below.

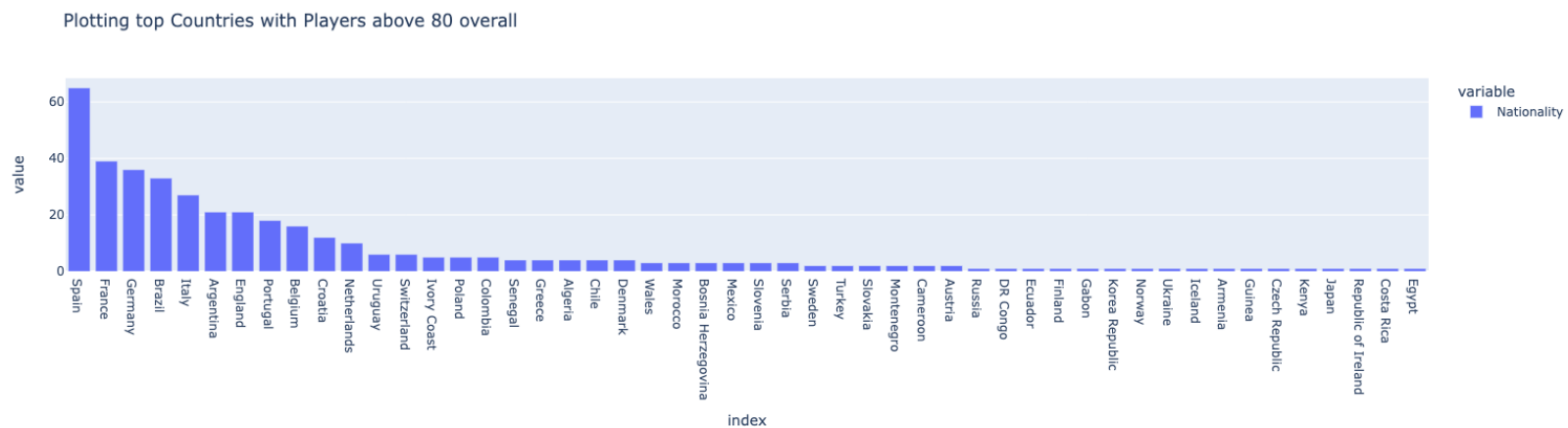
## 1. Plotting the Number of players by Nationality

This plot allows us to understand the number of players with nationality as England has exponentially more professional players than countries like Barbados, Puerto Rico, etc.



## 2. Plotting top Countries with Players above 80 overall

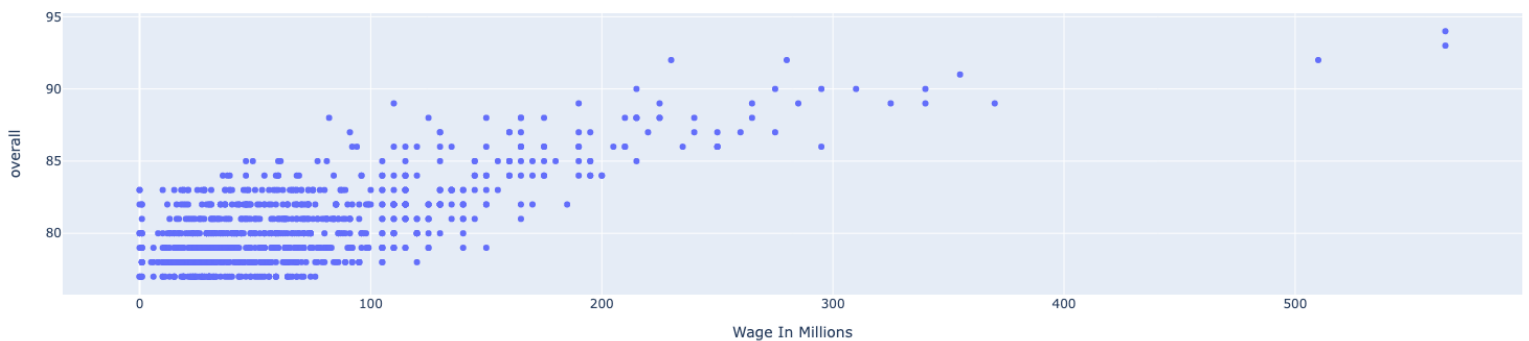
This plot tells us a different story altogether than the earlier graph. It is interesting to see that even though England has the highest number of players in Fifa, Spain comes out on top to have the highest number of valuable players with 60+ players with overall attributes over 80.



### 3. Plotting Player Wage in Millions vs Ability

The scatter plot shown below visualizes how the Wage for players increases as their skill level increases. The scatter plot can be used to identify trends and patterns in the data, as well as to identify outliers and unusual data points. It can also be used to make predictions about future player wages based on their ability level. provides a powerful visual tool for understanding the relationship between player wages and ability in professional sports.

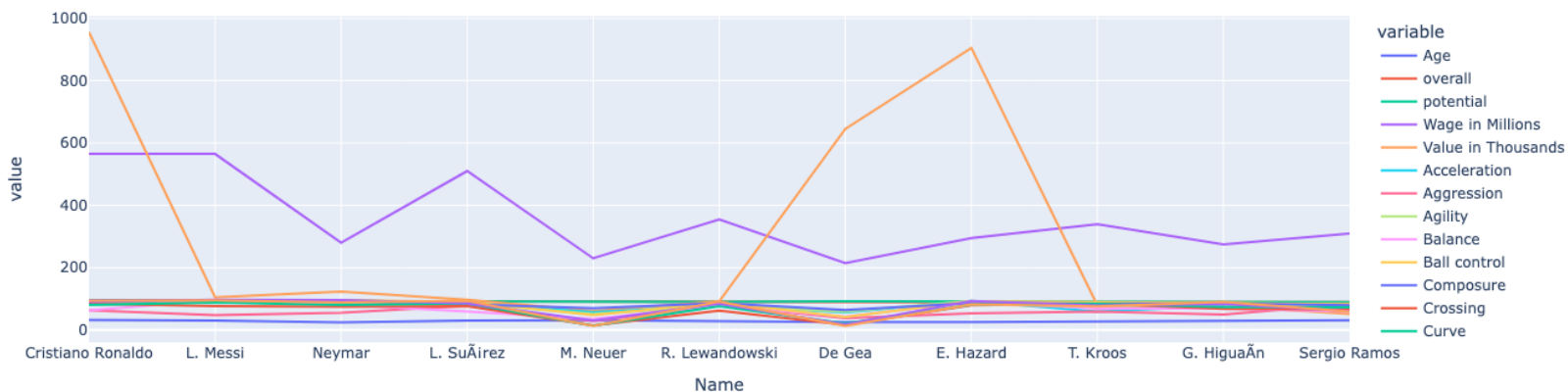
Plotting Player Wage in Millions vs Ability



### 4. Plotting Players with 90+ Overall and their Abilities

The line plot below describes the skill level of the top players and their attributes. For Example, We can see that Cristiano Ronaldo and Eden Hazard have comparatively higher Wages in the range of 800 to 1000 million dollars with all other skills being at par with other professional players like Messi, Neymar, etc.

Plotting Players with 90+ Overall and their Abilities

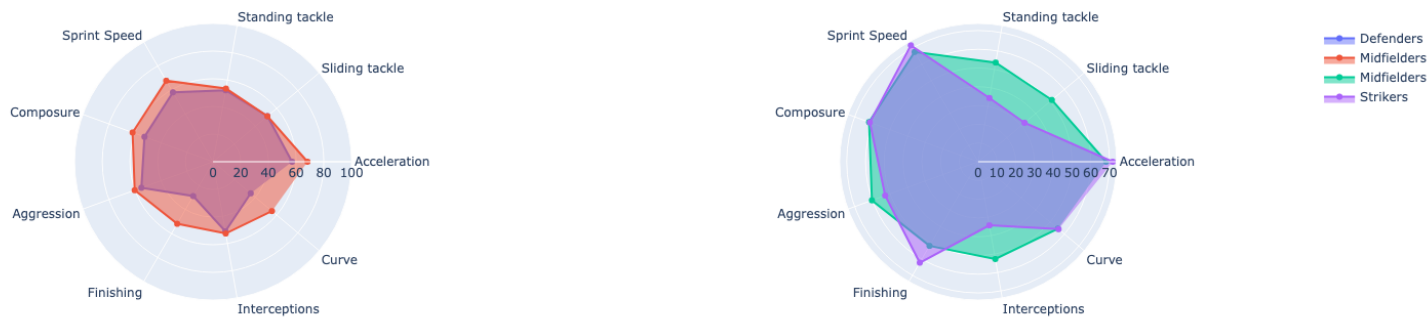


## 5. Plotting the Average skill level of different Positioning Players to understand key skill differences

The left subplot is a subplot of Defenders vs Midfielders, where we can see that the defending abilities like tackling Interceptions and aggression are at par for both however, the midfielders have higher sprint, finishing, and curve compared to defenders.

The right subplot is a plot of the abilities of a Striker vs the abilities of a midfielder. We can deduce that midfielders have a higher sprint rate, finishing, acceleration, and Curve as compared to defenders but when compared to Strikers, the natural abilities of a Striker come out on top.

Plotting similarities Between Defenders, Strikers and Midfielders



## 6. Plotting the Average skill level of Defenders vs Strikers to see major positioning differences

This subplot is a plot of the abilities of a defender vs the abilities of a Striker. We can analyze that the set of skills is different for players in different positions. For example, a defender needs to intercept and tackle very well as compared to a striker who need not have a good interception and tackling skills. Whereas a striker needs to have more sprint speed, Acceleration, exponentially higher finishing, and special skills like a curve.

Plotting similarities Between Different Defenders and Strikers

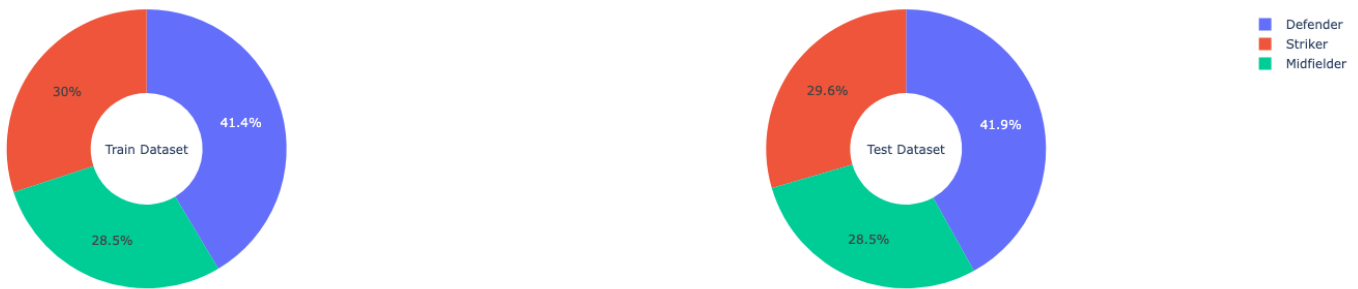


- **Data Modeling:**

In this step, we first split the dataset into train and test datasets using an 80-20 split by using the random split method of Pyspark pandas.

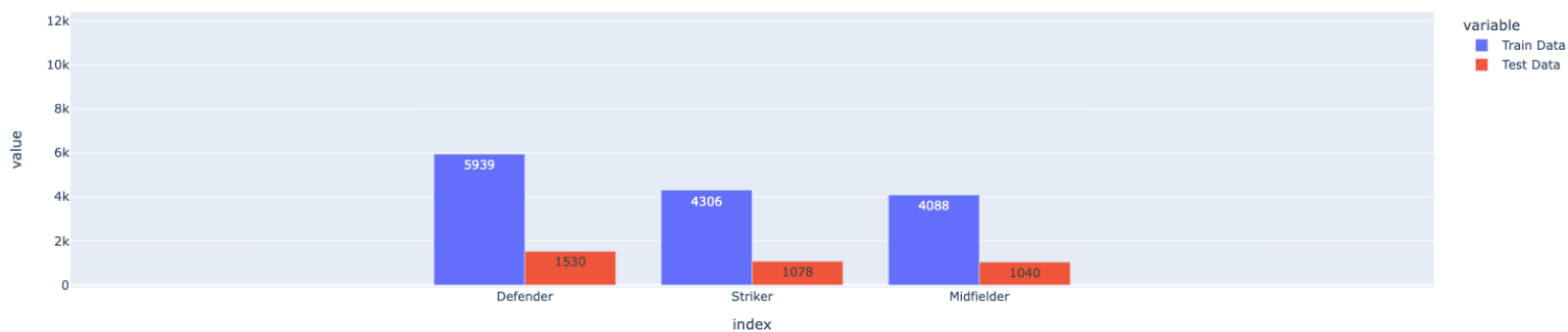
The datasets are balanced and equally distributed reducing bias.

Plotting Label Distribution for Train Test Split



To better understand the split we visualize the train and test splits side by side in a bar chart which also gives us a fair idea of the label distribution.

Plotting Label Distribution for Train Test Split

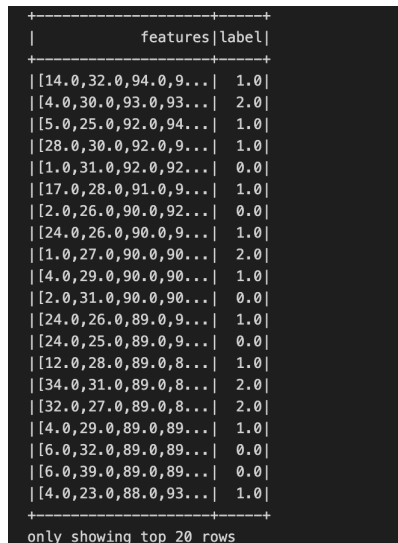


The train and test datasets are balanced however the number of players with the defender's label is slightly on the higher end. Since the training data is larger for Defenders, the models should be able to predict Defenders more accurately.

- **Data Pipeline**

Machine learning models require the data to be in a specific format and can only accept numerical values. In order to make the dataset compatible<sup>[95]</sup> to pass through machine learning models and deep learning neural networks we create a data pipeline and pass our dataset through the same.

The data pipeline makes use of StringIndexers and VectorAssemblers to convert String values in the dataset to Indexes and then model the data in the form of features and labels.



features	label
[14.0, 32.0, 94.0, 9.0, ...]	1.0
[4.0, 30.0, 93.0, 93.0, ...]	2.0
[5.0, 25.0, 92.0, 94.0, ...]	1.0
[28.0, 30.0, 92.0, 9.0, ...]	1.0
[1.0, 31.0, 92.0, 92.0, ...]	0.0
[17.0, 28.0, 91.0, 9.0, ...]	1.0
[2.0, 26.0, 90.0, 92.0, ...]	0.0
[24.0, 26.0, 90.0, 9.0, ...]	1.0
[1.0, 27.0, 90.0, 90.0, ...]	2.0
[4.0, 29.0, 90.0, 90.0, ...]	1.0
[2.0, 31.0, 90.0, 90.0, ...]	0.0
[24.0, 26.0, 89.0, 9.0, ...]	1.0
[24.0, 25.0, 89.0, 9.0, ...]	0.0
[12.0, 28.0, 89.0, 8.0, ...]	1.0
[34.0, 31.0, 89.0, 8.0, ...]	2.0
[32.0, 27.0, 89.0, 8.0, ...]	2.0
[4.0, 29.0, 89.0, 89.0, ...]	1.0
[6.0, 32.0, 89.0, 89.0, ...]	0.0
[6.0, 39.0, 89.0, 89.0, ...]	0.0
[4.0, 23.0, 88.0, 93.0, ...]	1.0

only showing top 20 rows

- **Making Predictions**

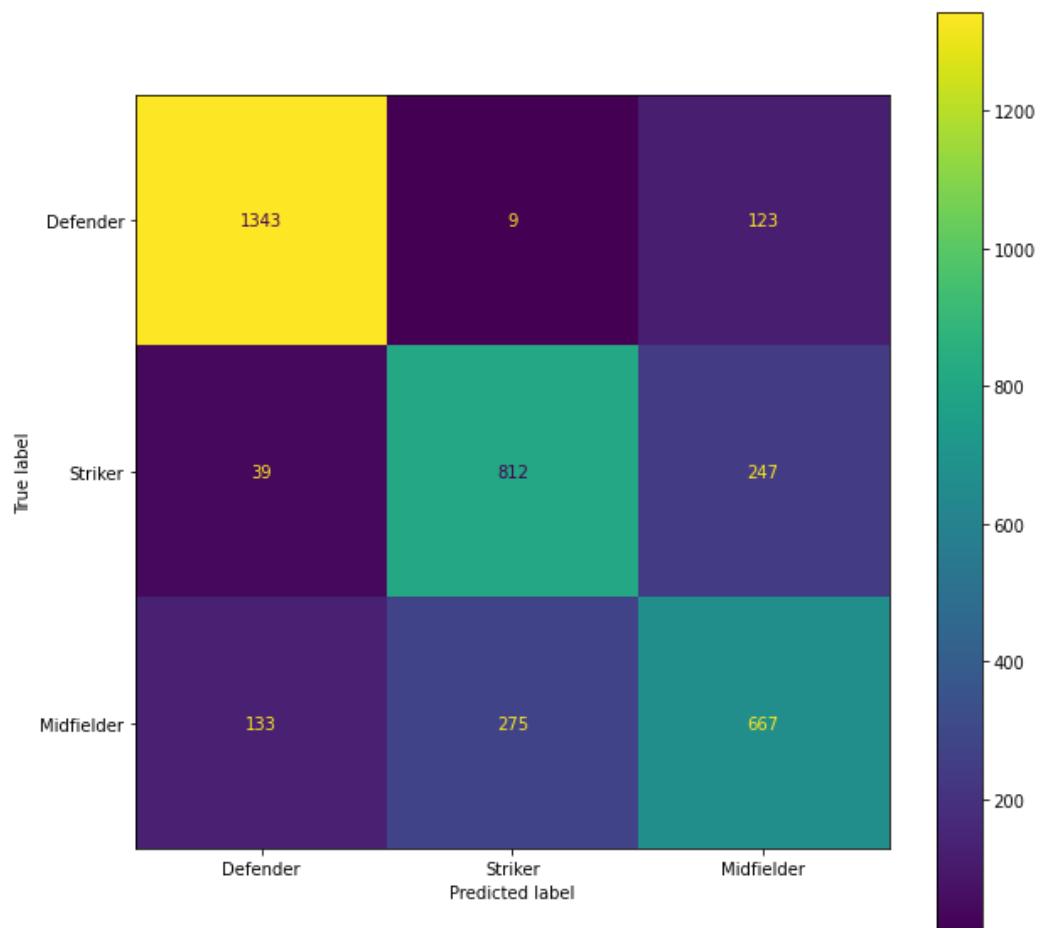
Now, once we have the data preprocessing, we start training the machine learning models<sup>[66]</sup> on the training dataset and eventually test the accuracy of the same on the test dataset.

For the purpose of this project, we are making use of 3 machine learning models.

### 1. Logistic Regression

In statistics, the logistic model is a statistical model that models the probability of an event taking place by having the log odds for the event be a linear combination of one or more independent variables. In regression analysis, logistic regression is estimating the parameters of a logistic model.

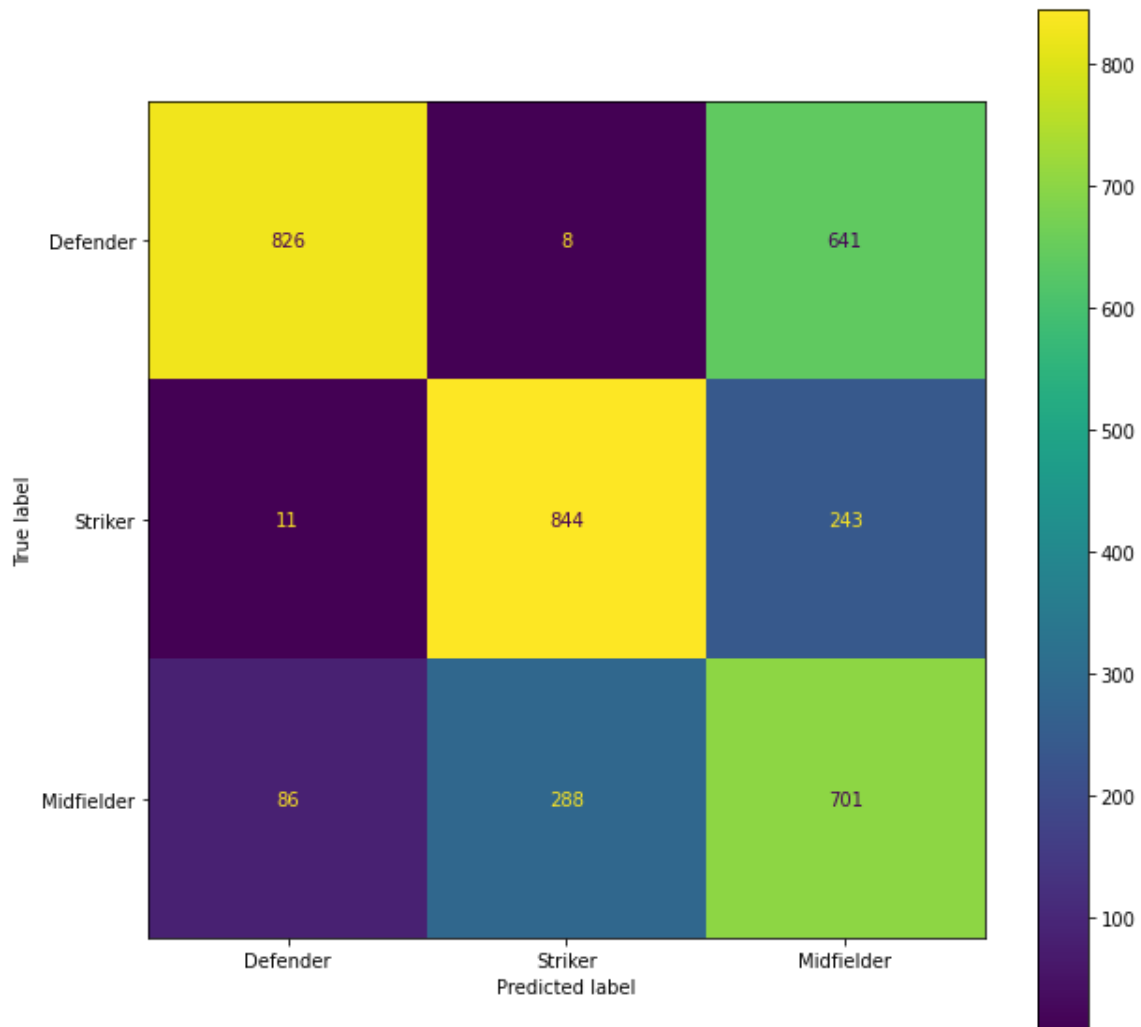
The Logistic regression model produces the following results and as we predicted earlier the model is able to classify defenders better than the rest of the labels because of more data points available in the training dataset. The average accuracy produced by the logistic regression model is **0.77**



## 2. Naive Bayes

The Naïve Bayes classifier is a supervised machine learning algorithm, which is used for classification tasks, like text classification. It is also part of a family of generative learning algorithms, meaning that it seeks to model the distribution of inputs of a given class or category. Unlike discriminative classifiers, like logistic regression, it does not learn which features are most important to differentiate between classes

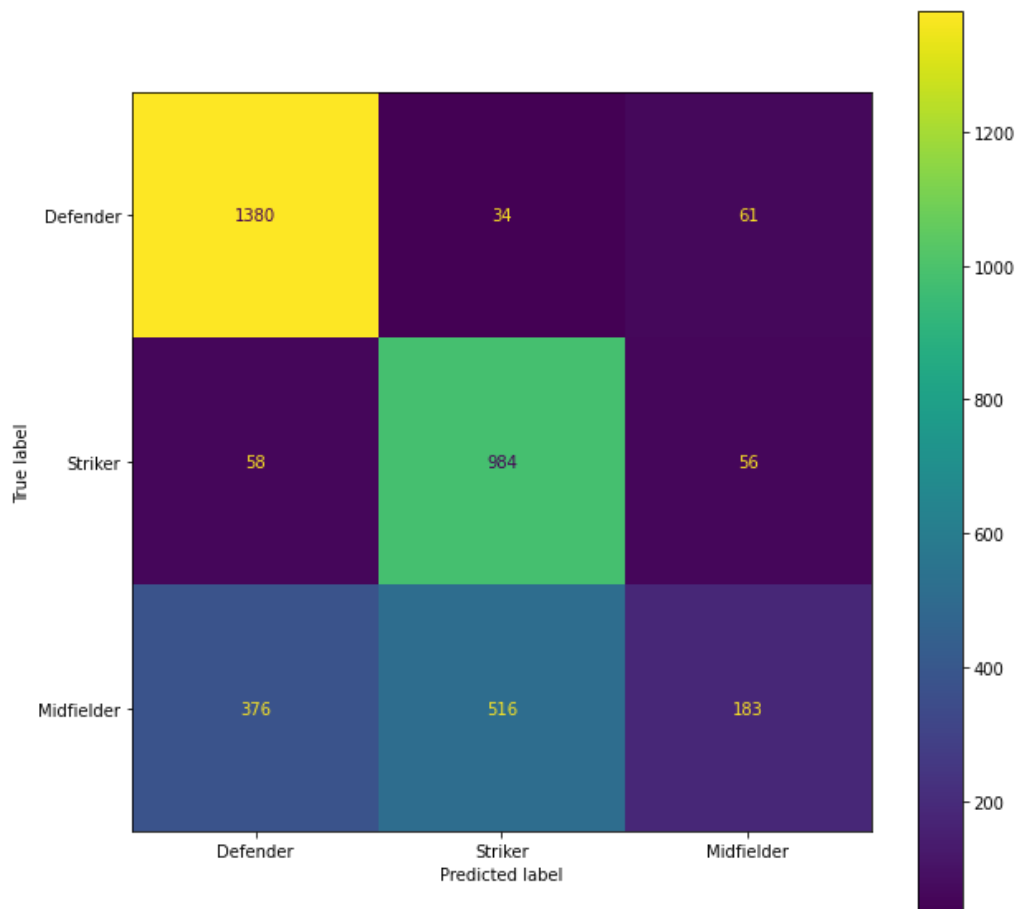
The Naive Bayes model is able to classify labels with a shade better than random accuracy with an average accuracy of **0.65**



### 3. Neural Networks

A neural network is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain. It creates an adaptive system that computers use to learn from their mistakes and improve continuously.

The neural network being the most advanced technique of the 3 is able to predict more defenders and Strikers as compared to any other model used but looking at the classification report, we observed that this deep learning network model is biased towards the Defenders and the Strikers label more and tries to predict almost all labels as Defender or Striker and thus performs poorly for the Midfielder label.





- **Classification reports for Machine Learning Models:**

- **Logistic Regression**

Logistic Regression Model Results				
	precision	recall	f1-score	support
Defender	0.89	0.91	0.90	1475
Striker	0.74	0.74	0.74	1098
Midfielder	0.64	0.62	0.63	1075
accuracy			0.77	3648
macro avg	0.76	0.76	0.76	3648
weighted avg	0.77	0.77	0.77	3648

- **Naive Bayes**

Naive Bayes Model Results				
	precision	recall	f1-score	support
Defender	0.89	0.56	0.69	1475
Striker	0.74	0.77	0.75	1098
Midfielder	0.44	0.65	0.53	1075
accuracy			0.65	3648
macro avg	0.69	0.66	0.66	3648
weighted avg	0.72	0.65	0.66	3648

- **Neural Network**

Neural Network Model Results				
	precision	recall	f1-score	support
Defender	0.76	0.94	0.84	1475
Striker	0.64	0.90	0.75	1098
Midfielder	0.61	0.17	0.27	1075
accuracy			0.70	3648
macro avg	0.67	0.67	0.62	3648
weighted avg	0.68	0.70	0.64	3648

Now, this is really an interesting result and even though it may look like the model is performing poorly we need to understand that the neural network tries to think as a human brain. To analyze this machine learning model we need to dive deeper into the world of soccer. Let's take a step back and understand how the positions in soccer are placed strategically. Talking about Midfielders, there are 2 sub-types in midfielders called as **Attacking Midfielders** and **Defending Midfielder**.

The defending midfielders have higher defending skills along with improved passing and ball control accuracy. Now, defenders also have the same set of skills and all defenders who are actually wing backs (LWB and RWB) also have increased pace and great long ball and crossing skills which we would normally think to see in a midfielder. For a human to think, the players who play as a defender and players who play as defending midfielders should have the same defending skills with little difference. When the Neural network tries to understand the weights for defenders and midfielders, it understands this data in a similar way and since we have more data for defenders it goes on to predict the midfielders as defenders.

The attacking midfielders have higher passing and shooting accuracy and dribbling skills as they are required to help the strikers and provide the ball for the strikers to score. Sometimes, the midfielders are also in a better position and end up taking the shot to score a goal. This tells us that the abilities of strikers and attacking midfielders are similar. Talking about generative machine learning models which give equal importance to all features, neural networks attempts to understand the relationship between the features and aggregate an output based on statistical calculations. Thus, the neural network ends up predicting the attacking midfielders as strikers.

This change can be carefully observed in plots 5 and 6 of the data visualization paragraph. The abovementioned plots skillfully depict the differences between players in different positions and how it affects their performance.

## **6. FINAL CONCLUSION:**

After the exploratory analysis of the dataset and taking a closer look at the predictions to analyze the results we conclude that for this dataset the Logistic Regression would be the best type of machine learning model as it is a discriminative machine learning model and it is able to accurately understand the importance of different features for a different type of players however talking about the big data approach and exploiting the features of big data to provide real-time and accurate predictions we believe that the neural network solution would be much beneficial as the dataset keeps on increasing the neural network keeps learning. We could also split the dataset into more labels as defending midfielders and attacking midfielders and the neural network will perform better than the rest of the classifiers.

To assist us in controlling Big Data, PySpark API combines the practicality of Python with the strength of Apache Spark. Spark is a complete project that has produced an entire ecosystem. It fits really well with the project we were trying to create. PySpark offers thorough and top-notch methods for processing zettabytes and petabytes of data on parallel clusters significantly faster, making it superior to traditional Python machine-learning applications. PySpark additionally features a machine learning pipeline and statistical analysis methods. Python is a simple language to pick up and use. It offers a straightforward but comprehensive API. Python offers significantly better maintenance, familiarity, and legible code. In contrast to Java or Scala, it provides a wide range of data visualization choices that we may use by utilizing PySpark. In addition to its simplicity and its ability to handle errors, its ease of use makes Pyspark the best technology for FIFA 18 player prediction dataset analysis.

## **7. Project Code:**

Please find the entire codebase on the below mentioned github repository

[https://github.com/shubham11-07/Fifa\\_Player\\_Prediction](https://github.com/shubham11-07/Fifa_Player_Prediction)

## **7. REFERENCES:**

1. FIFA 18 player prediction dataset:  
<https://www.kaggle.com/datasets/edith2021/fifa-18-player-prediction>
2. Python: <https://docs.python.org/3/library/>
3. ApacheSpark: <https://www.projectpro.io/article/scala-vs-python-for-apache-spark/213>
4. Hive: <https://www.projectpro.io/article/spark-vs-hive/480>
5. Hadoop: <https://aws.amazon.com/emr/details/hadoop/what-is-hadoop/>
6. Gianluca Morciano; Andrea Zingoni; Andrea Morachioli; Giuseppe Calabrò Machine Learning prediction of the expected performance of football player during training (2022): IEEE
7. McCabe, A., & Trevathan, J. (2008). Artificial Intelligence in Sports Prediction. Fifth International Conference on Information Technology: New Generations.
8. Pantzalis, V., & Tjortjis, C. (2020). Sports Analytics for Football League Table and Player
9. Performance Prediction. 2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA).
10. Apostolou, K., & Tjortjis, C. (2019). Sports Analytics algorithms for performance prediction. 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA). Patras, Greece: IEEE.
11. Breiman, L. (2001). Random Forests. Machine Learning, 5-32.
12. Plotly Visualization library for python : <https://plotly.com/graphing-libraries/>