Cairo University

Team 7
Big Data & Cloud Computing
FIFA 22 Analysis

Name	Section	Bench Number
Ahmed Ihab	1	2
Ahmed Gamal	1	3
Weaam Bassem	2	36
Yousef Ahmed	2	38

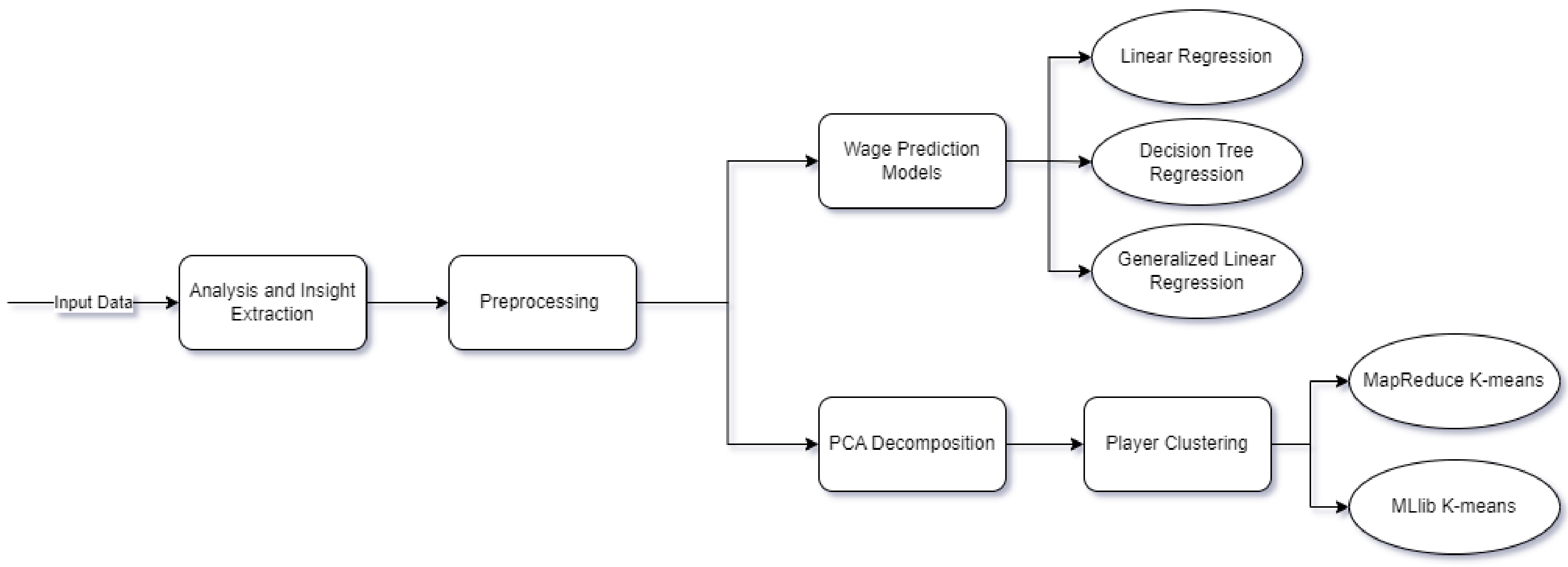
Spring 2023

Problem Description

Football is one of the most popular and profitable sports in the world, with millions of fans and billions of dollars in revenue. However, football is also a highly competitive and dynamic industry, where clubs and managers must constantly scout, recruit, train, and retain the best players for their team.

Throughout this project we have analyzed the FIFA 22 complete player dataset | Kaggle, extracted valuable insights about the players, predicted the player wages, and clustered the players using MapReduce

Project Pipeline



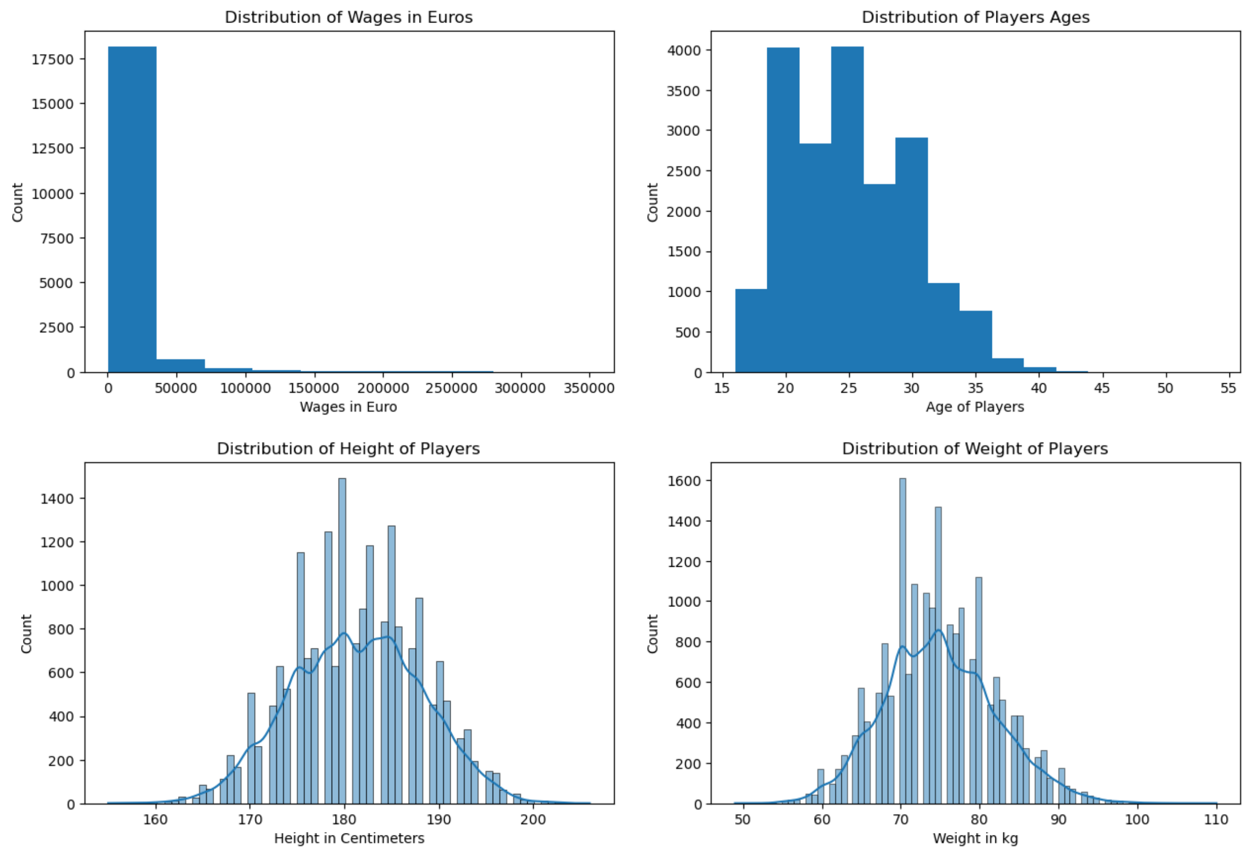
Preprocessing

Insert preprocessing steps here:

- Drop any unneeded column from the data such as ids and any column with urls.
- Checking for nullity in all columns
  - No null values found in all columns.
- Define all categorical columns and perform the following
  - Label Encoder to encode the categorical column to number.
  - One hot Encoder to represent categorical variables as numerical values in a machine learning model.

Exploratory Data Analysis

Q1. What are the height, wage, weight and age distribution of players?

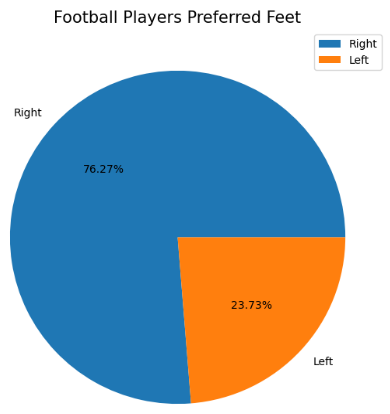


Insights:

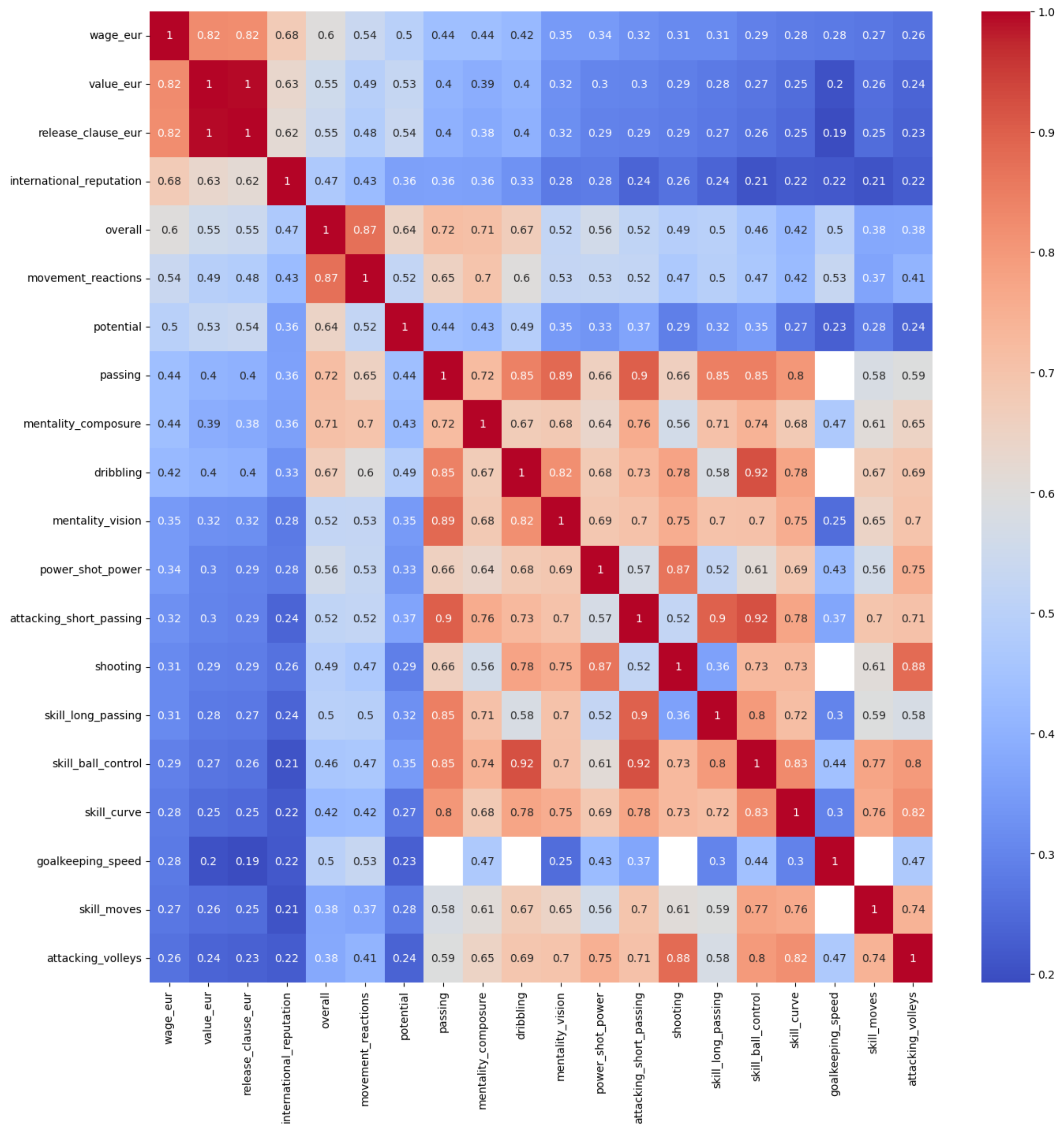
- Most of players height are around 180CM
- Most of players have weights between 70-80KG
- There is a higher portion of players aged less than 25
- Wages are most distributed from 0 to 40000 Euro

Q2. What is the percentage of players whose preferred foot is right?

Around 78% of players prefer right foot



Q3. What attributes have the max effect on the wage of players? (correlation)



We faced some issues as the data have **110 columns** which will make the heatmap impossible to be readable

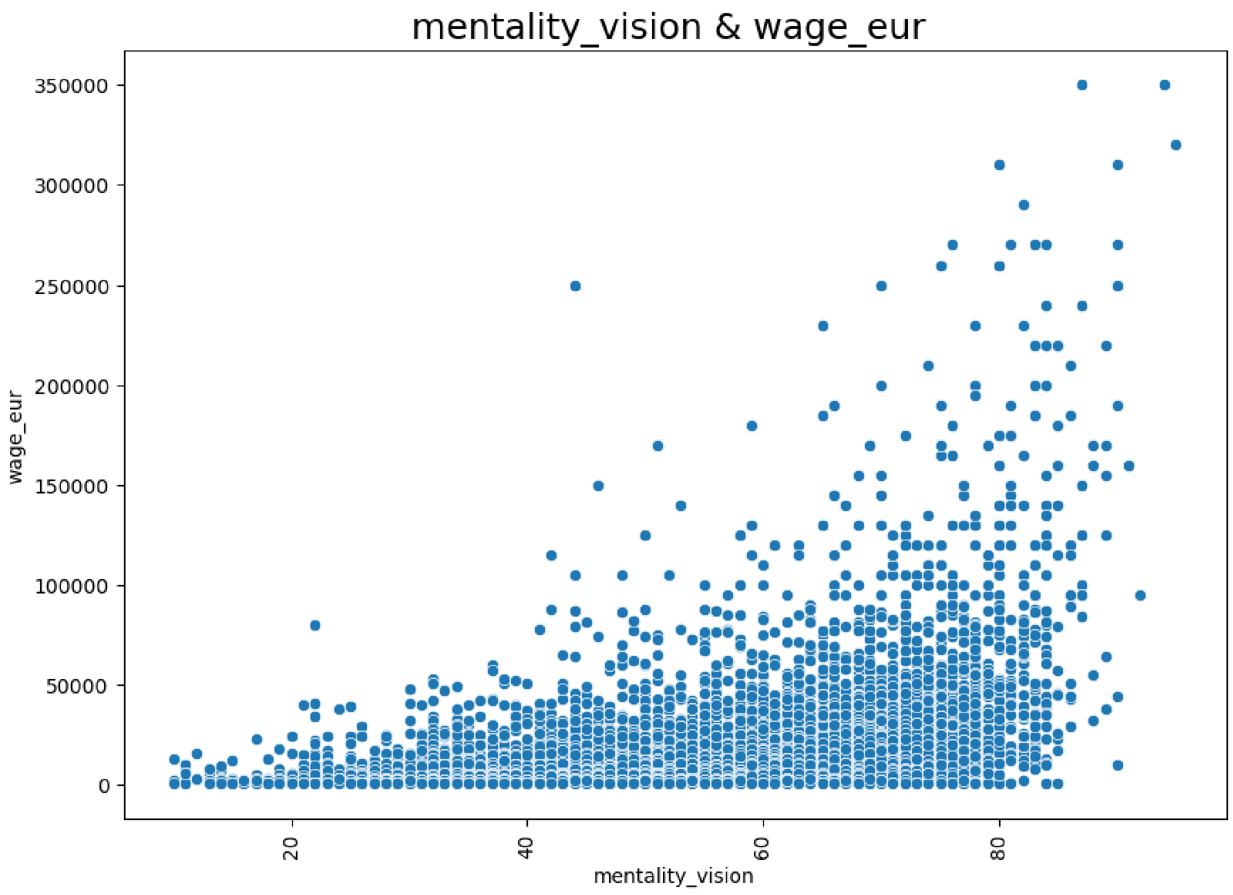
Mitigation:

- We chose only the columns with type **number** which resulted in only 60 column
- Also we decided to choose the top 20 correlated columns and now the heatmap makes more sense

Insights:

- From the previous graph we can see that the top 20 correlated columns to price which will help us in machine learning model for predicting the wage for the players

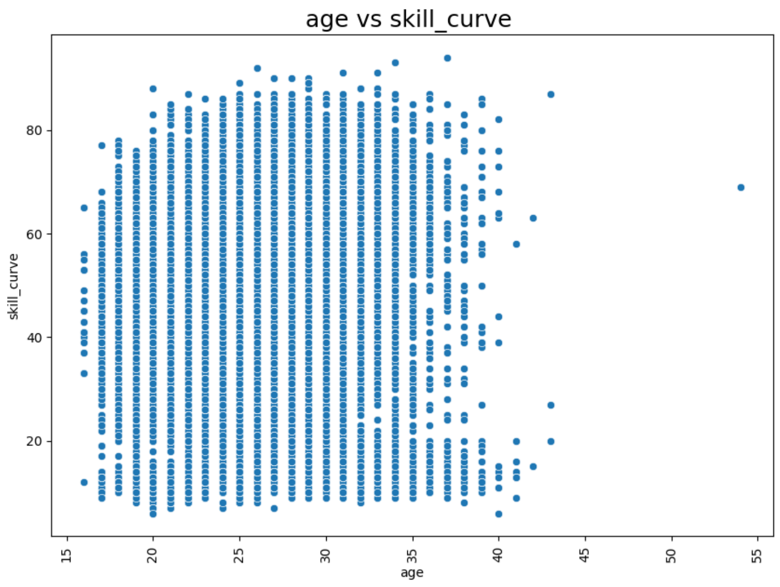
Q4. What is the relation between mentality_vision of the player and his wage?



Insights:

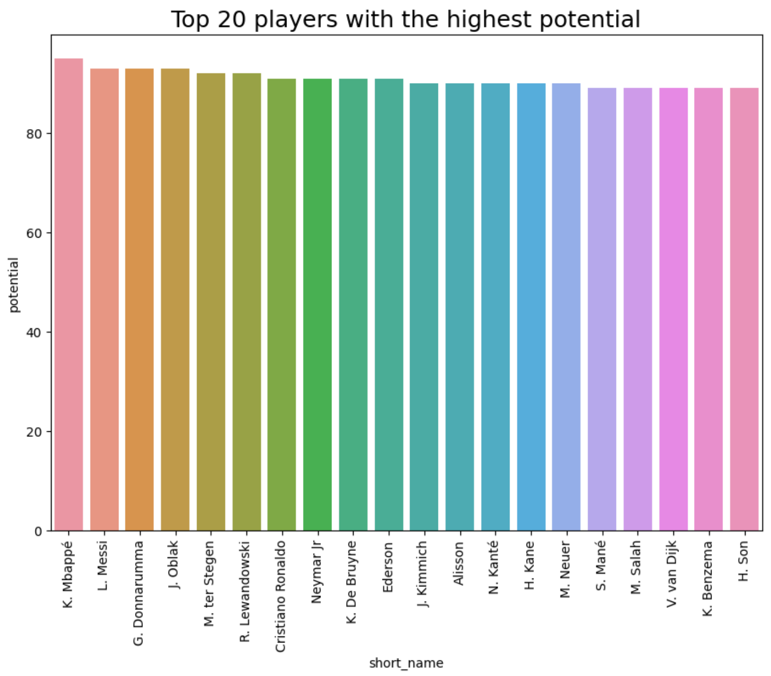
- The more the mentality vision of the player the more wage he gain. And this proves **Mohamed Salah** Hypothesis 😂

Q5. Is the skill curve of players affected by their age?



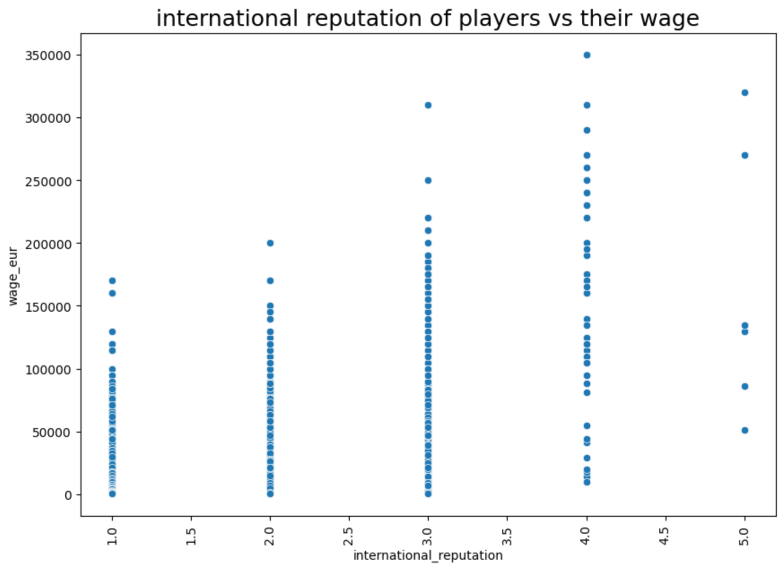
- **As shown in the figure between player age and skill curve :**
    - Surprisingly 😯 , the player's age and skill curve almost not correlated with each other, then skill curve of a player doesn't affected by his age.

Q6. Who are the top 20 potential players in the data set?



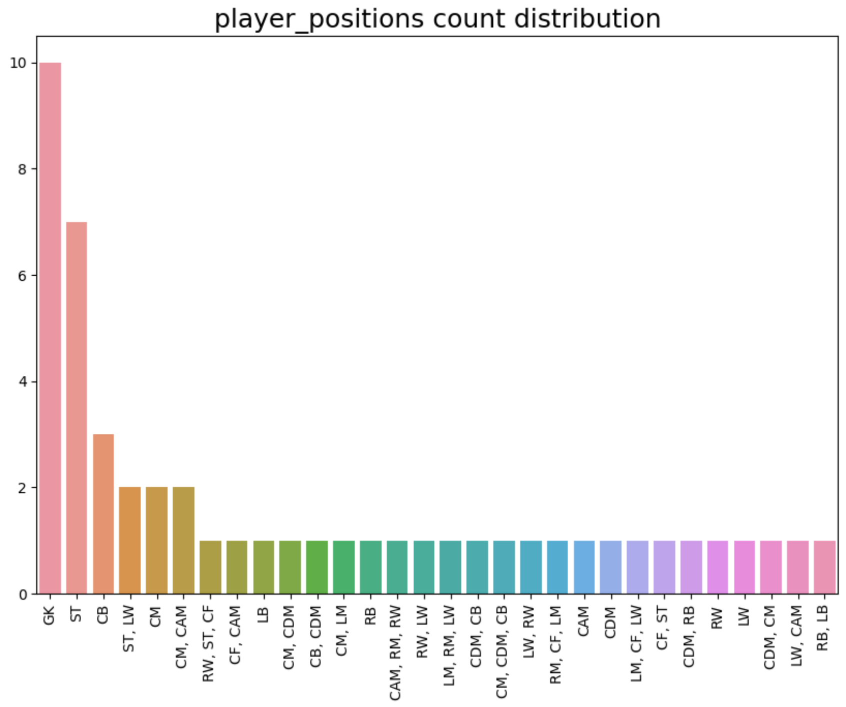
- **As shown in figure for top 20 players in potential value 💪:**
    - K.Mbappe is the highest player with potential 95 🥇.
    - L.Messi is the second highest player with potential 93 🥈.
    - H.son is the last player with potential 89.

Q7. How does international reputation affect the wage?



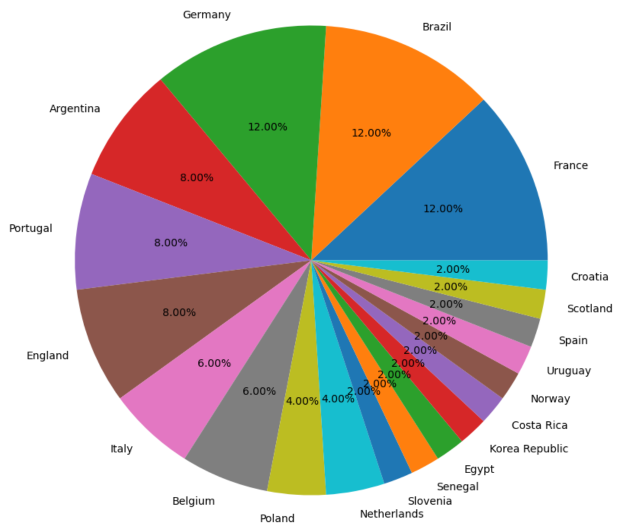
- **As shown in figure the correlation between international reputation and player wage : 👌**
    - There is a high positive correlation between the international reputation of the player and his wage.
    - As the player wage increases with the international reputation.

Q8. What is the distribution of player_positions in the data set for the top 50?



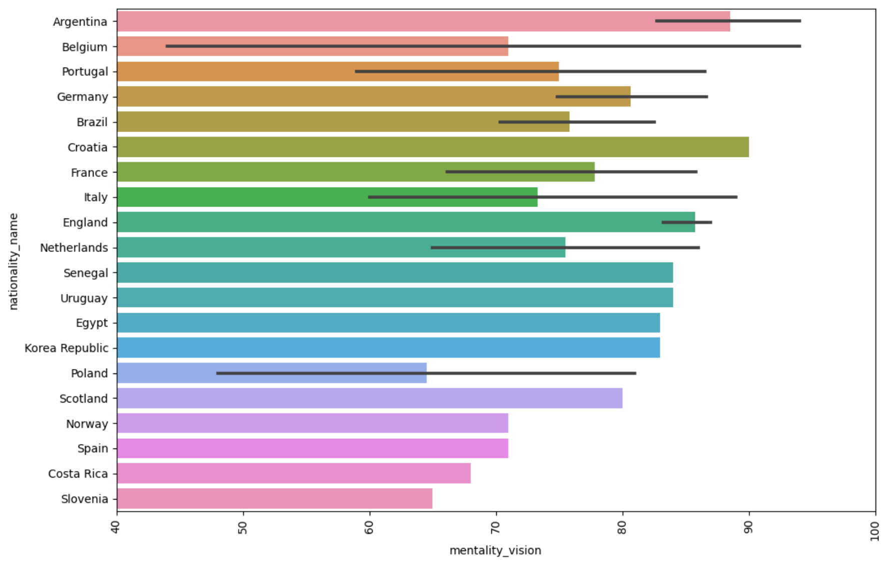
- **As shown in the bar chart for the top 50 players in overall figure that**
    - Surprisingly 😯 , that the most freq position in the top 50 players is GK.
    - The second freq position in the top 50 playersis ST.

Q9. What is the nationality percentage for the top 50 players?



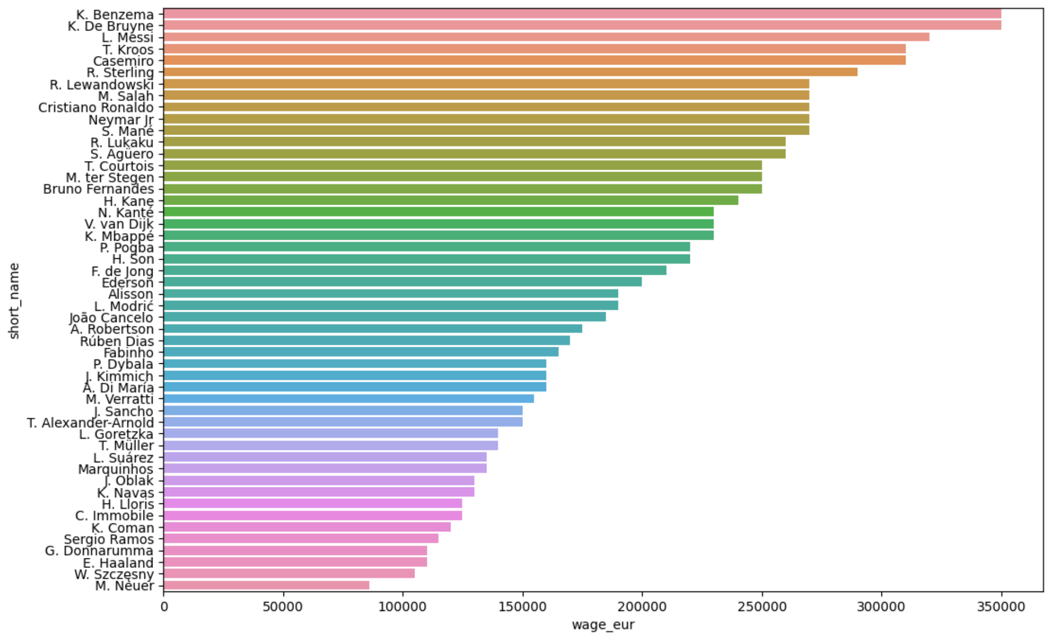
- **As shown from the pie chart the top 5 countries with the highest number of players for top 50 player are :**
    1. France 12.00 %
    2. Germany 12.00%
    3. Brazil 12.00%
    4. Argentina 8.00%
    5. Portugal 8.00%

Q10. What is the nationality of the highest mentality of the top 50 players?



- **As shown from bar graph for the nationality of the highest mentality vision players in the top 50 players** 🥇
    - The top nationality for mentality is Argentina
    - The lowest nationality for mentality is Slovenia

Q11. who are the highest wage players in the top 50 players ?



- **As shown from bar graph for the players of the highest wage for players in the top 50 players 💰**
    - K.Benzema and K. De Bruyne are the players with the highest wage is about 350k eur 🥇.
    - L.Messi is in the second position with about 320k eur 🥈.
    - M.Neuer is in the last position with about 86k euro.

Feature Engineering

Before feeding the dataset to the clustering models, we have applied PCA to bring down the number of features from 110 to the number of features that would achieve 95% variance, this number happened to be 30 features.

We have then used only the first three components to visualize clustering. Note that the first three components achieve 51.87% variance.

Used Models

Regression Models

The main column we have been trying to predict was the wage_eur column, which depicts the players' wage in euros. We have tried 3 regression models to predict the column using classifiers provided by PySpark's MLlib. The columns we have used were the 20 that had the highest correlation with the wage column.

Using MLlib's regression classes requires the data to first be transformed into a PySpark dataframe. We have then dealt with numerical and categorical features separately.

The numerical features were scaled using StandardScaler, then assembled into a feature vector using VectorAssembler. The categorical features were one-hot encoded, then assembled into a feature vector. Finally the numerical and categorical features vectors were assembled into one feature vector

The models we used were:

- Linear Regression:

The most basic regression model derived from a least-squares fit

- Decision Tree Regression:

A decision tree learning algorithm for regression

- Generalized Linear Regression:

Fits a Generalized Linear Model specified by giving a symbolic description of the linear predictor (link function) and a description of the error distribution. The reason we sought out this model is that linear regression produced a few negative predictions for a column of positive values. This led us to using a generalized linear regression model to enforce sampling data from a Poisson distribution and prevent negative values. This has decreased the loss, and increased the R2 measure (a measure that shows how well the data fit the regression model-the goodness of fit) greatly.

Clustering Models

We used Kmeans Clustering to cluster the similar players based on the top 3 PCA components. We used PCA as it was mentioned above and we picked the first 30 PCA components but to make it easy to visualize the clusters of data we picked the first 3 components

We applied Kmeans with two ways and compared their performance:

- Our implementation using MapReduce
- Ready made one from PySpark MLib

Steps:

1. Calculated WCSS to determine the best number of clusters
2. Initialize the centroids by picking random k points from the data and mark them as centroids
3. The map is responsible for calculating the best cluster for a given point and return (bestcluster , (point , 1))
4. The reducer sums the points and their count in order to compute the new centroid
5. The mapper compute the new centroids for each cluster
6. Use KMeans from pyspark.ml with same number of clusters and compare the results

(View the notebook for interactive graphs)

Results

The regression models have achieved the following test metrics:

	Linear Regression	Decision Tree Regression	Generalized Linear Regression
Mean Squared Error	47274300.81	87134811.76	**19240546.21**
Root Mean Squared Error	6875.63	9334.60	**4386.40**
Mean Absolute Error	3380.68	4068.31	**1539.21**
R2	0.88	0.795	**0.96**

As for the clustering algorithms, there is no measure for the accuracy since this is an unsupervised learning model, however we have used Within Cluster Sum of Squares to determine the optimal number of clusters. Using this number, the clustering done by our MapReduce Model and MLlib's model is quite similar.

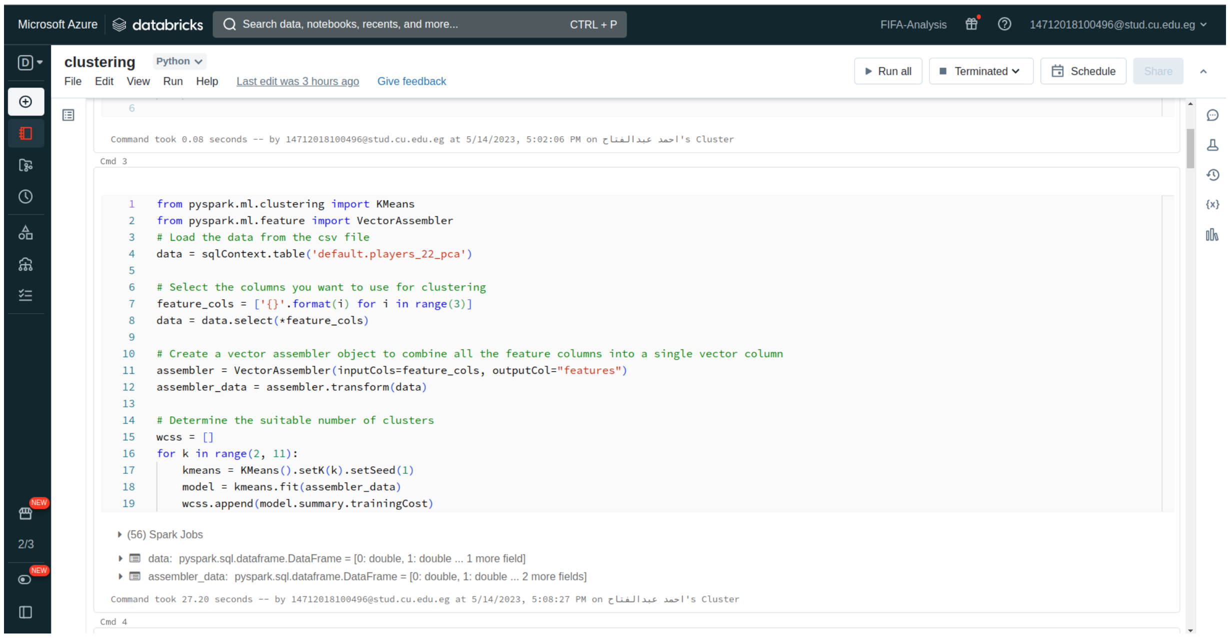
Azure Databricks (Bouns)

Apache Spark is at the heart of the Azure Databricks Lakehouse Platform and is the technology powering compute clusters and SQL warehouses on the platform. Azure Databricks is an optimized platform for Apache Spark, providing an efficient and simple platform for running Apache Spark workloads.
The Databricks company was founded by the original creators of **Apache Spark**. As an open source software project

Steps to configure the cluster:

- Open Azure Portal and search for Azure Databricks service
- Create an Azure Databricks workspace
- After creation, go to Workspace. It will open a fully integrated Databricks workspace
- From the side menu choose compute and configure your cluster specs (according to Students Subscription we are only allowed to run the cluster in single node mode)
- Click create notebook and import the one you need
- Start working with PySpark

Screenshots:

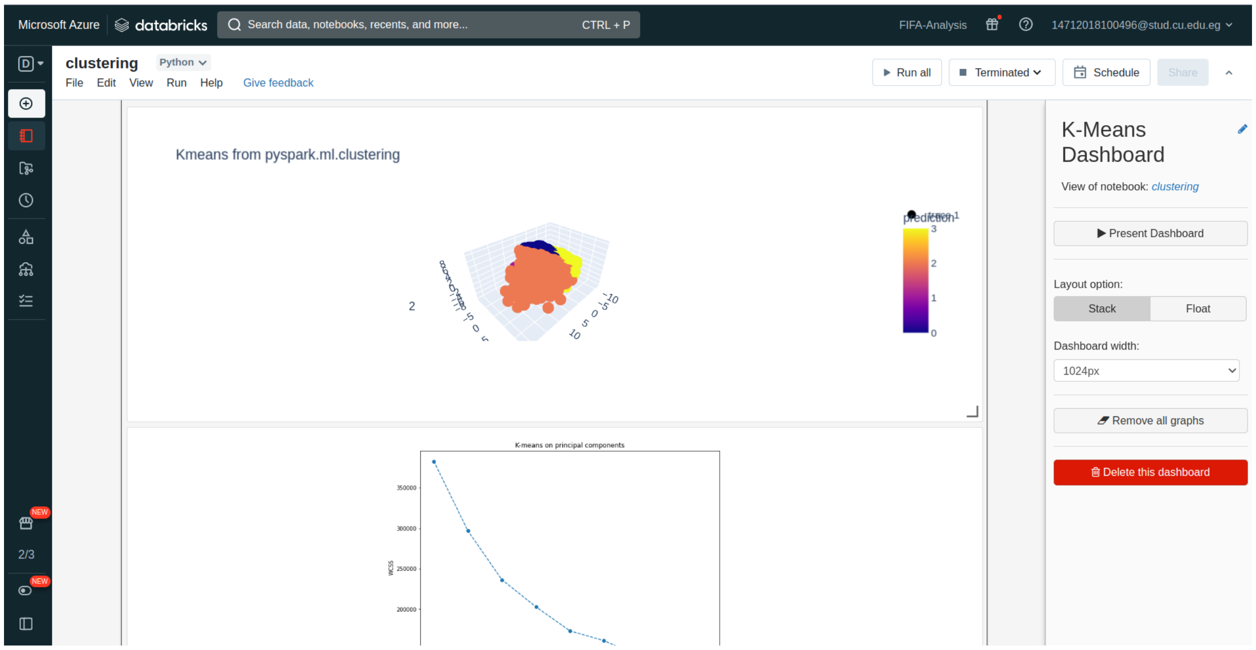


```
Command took 0.08 seconds -- by 14712018100496@stud.cu.edu.eg at 5/14/2023, 5:02:06 PM on احمد عبدالفتاح's Cluster
Cmd 3
```

```python
from pyspark.ml.clustering import KMeans
from pyspark.ml.feature import VectorAssembler
# Load the data from the csv file
data = sqlContext.table('default.players_22_pca')

# Select the columns you want to use for clustering
feature_cols = ['{}'.format(i) for i in range(3)]
data = data.select(*feature_cols)

# Create a vector assembler object to combine all the feature columns into a single vector column
assembler = VectorAssembler(inputCols=feature_cols, outputCol="features")
assembler_data = assembler.transform(data)

# Determine the suitable number of clusters
wcss = []
for k in range(2, 11):
    kmeans = KMeans().setK(k).setSeed(1)
    model = kmeans.fit(assembler_data)
    wcss.append(model.summary.trainingCost)
```

```
(56) Spark Jobs
data:  pyspark.sql.dataframe.DataFrame = [0: double, 1: double ... 1 more field]
assembler_data:  pyspark.sql.dataframe.DataFrame = [0: double, 1: double ... 2 more fields]
Command took 27.20 seconds -- by 14712018100496@stud.cu.edu.eg at 5/14/2023, 5:08:27 PM on احمد عبدالفتاح's Cluster
Cmd 4
```

Working Notebook with PySpark



K-Means Dashboard

View of notebook: *clustering*

Layout option: Stack

Dashboard width: 1024px

Dashboard Containing All Graphs In The Notebook