

PROJECT REPORT

SP20 – INFO I535: Management, Access and Use of Big Data

OJAAS HAMPIHOLI (ojshampi@iu.edu)

Problem Statement:

Predictive Modelling to determine Market Value of a player based on other features.

Background and Introduction to the Data and Problem:

Football is one of the most followed sports around the world. Globally, association football is played by over 250 million players in over 200 nations, and has the highest television audience in sport, making it the most popular in the world. Football is a family of team sports that involve, to varying degrees, kicking a ball to score a goal. The Fédération Internationale de Football Association (FIFA) is a non-profit organization which describes itself as an international governing body of association football, futsal, beach soccer, and e-football. It is the highest governing body of football. FIFA is the governing body which ensures that all the football related activities are carried on around the globe within the prescribed rules and regulations.

FIFA maintains a record of all the players playing across the world in various leagues and various countries as well as continents. This dataset is maintained to keep a check on the global talent for football, as well to keep an overview on the financial transactions that happen with respect to a player in the football world. A very similar dataset is given to EA Sports which has the legal rights to the video game version of FIFA. One such dataset containing the relevant information has been made open source for all the aspiring data science enthusiasts to use and play around with.

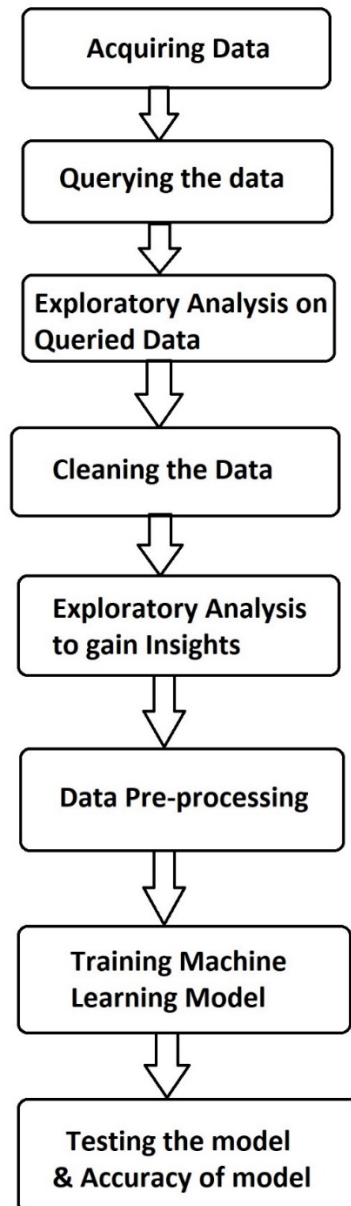
I have been an avid fan of football since I was 10 years old. I have been following the games and the trends in football for a lot of years now. I always wondered how the market value of a player is determined whenever a transfer of player between the club takes place. When I finally found the relevant dataset, I decided to ask myself to come with a model that predicts the market value of a player. This model takes various ratings of the player with respect to skillsets, position wise rating and the wage and release contract amounts into consideration in order to come up with a prediction for the market value of a player.

The dataset contains around 20000 records of various players, with all type of ratings, age and monetary details. The dataset contains 89 features for each player. The dataset has a lot of scope for preprocessing and applying predictive models. The dataset is a dense dataset, with a few NA values which come up in cells. In order to address the problem, I decided to create a pipeline which takes the raw data, performs data cleaning and pre-processing, apply queries and visualization and

eventually applies a machine learning model to the data to find the predictions for the Market Value of a player based on his other attribute ratings.

Methodology:

I got the data for the problem from an open source platform mentioned in the list of attachments. After acquiring the data, I decided to follow the normal predictive analytics pipeline. The pipeline that I followed is as follows:



Acquiring data - The first thing I did was to search for various sources online in order to get the relevant data. The data had to have all the relevant features that could help to build a regression model to predict the market value of a player. I found one such dataset on Kaggle.

Querying the data – Queries are run any dataset in order to get a subset of data which follows a certain constraint. In this case, I had certain assumptions about the relationship between Age, Overall ratings and Potential ratings of any player. I wanted to observe the fluctuations in ratings of the players as their ages increased and decreased. I also wanted to observe the trend of ratings in the three major football playing nations – England, Germany and Spain. The general assumption was that the major football playing nations would follow a similar trend with respect to distribution of player ratings. I also observed the overall average ratings of players in the age range of 20 and 40 to check the hypothesis that potential ratings of players with higher ages is lower. All the above queries and their results and graphs are discussed in detail in the Discussion Section below.

Exploratory Analysis on Queried Data – After running the queries in data.world website, the results from the queries were saved as a csv file. These csv files have been loaded into RStudio in Markdown format, in order to perform exploratory analysis. The graphs drawn in R help us to observe the trends clearly. These graphs have been attached alongside their analysis in the Discussion section below.

Cleaning the data – The next step that was a part of this pipeline was to check the quality of data and to perform the cleaning steps. In order to this, I loaded the data in Jupyter Notebook. Jupyter Notebook is an interactive python interface which is like a markdown file. This notebook can be used to perform predictive analytics on a dataset with ease. After loading the data to a dataframe in python, I checked for the null values that occur in any cells. The option would have been to replace the null cells with either mean or median of the feature or to drop the features having null values. Since the number of nulls were very low, I decided to drop the records which had null values.

Exploratory Analysis on Cleaned Data – I used exploratory analysis on cleaned data in order to observe the trend that is followed in the entire dataset. This analysis helped me to identify the top count of the nations that majority of the players come from, the majority of ratings in the dataset, the most common age group of the players as well as the relation between the market value and release clause, defensive characteristics and midfield characteristics.

Data Pre-processing – Before we can use the data in any predictive model, we must preprocess the data in order to make it ready to the format that is used in models. In order to do this, I had to remove the € symbol from the Wage, Value and Release Contract fields. This was done alongside converting the M for millions and K for thousands to numbers. This ensured that the feature columns held float values and could be treated as a numeric feature by the model. Apart from the above problem, the data has a few fields which use a (number + number) format in order to show the (overall + potential increase) in the ratings of the player with respect to various positions in the field.

Training and Testing Models – The very first step that was taken was to train a simple Linear Regression model. In order to see if the model improves if any other models are applied, SVM Regressor, Decision Tree Regressor and Neural Network Regressor. Then comparison between the models was made based on accuracy scores to determine the better model.

Results and Discussion:

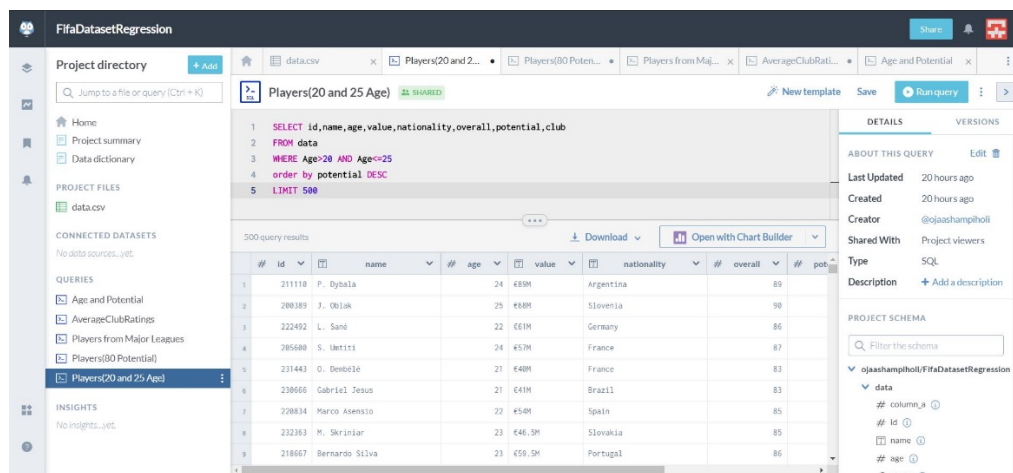
Query and Exploratory Analysis on the data:

Tool Used – data.world website, RStudio(rmd format with libraries- tidyverse, ggplot2, dplyr)

Data.world allows a user to upload his data to a private repository and the allows the user to write SQL Queries on the data in order to gain insights on the data. Data.world also allows a user to create graphs of basic kinds. It is a one stop destination if the user wants to acquire the data and perform exploratory analysis on it.

R is a free software which is used primarily for statistical and graphical processing. R is an interpreted language which is used in the markdown format. The libraries in R such as tidyverse and ggplot2 help greatly to draw meaningful graphs and thus help with exploratory analysis.

Query1 –



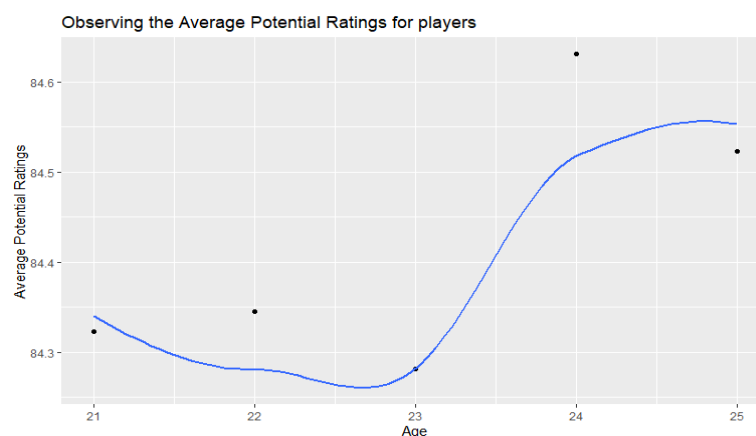
The screenshot shows the Data.world interface for a project named 'FifaDatasetRegression'. The query editor displays the following SQL query:

```
1 SELECT id,name,age,value,nationality,overall,potential,club
2 FROM data
3 WHERE Age>20 AND Age<=25
4 order by potential DESC
5 LIMIT 500
```

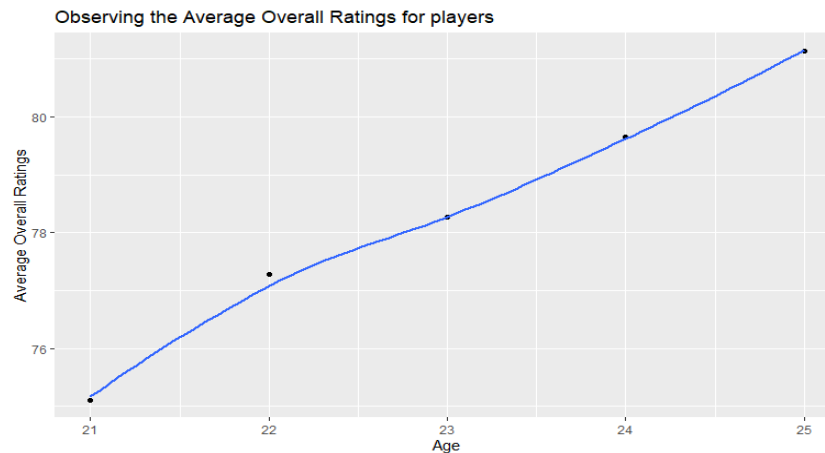
The query results are displayed in a table with 10 columns: #, id, name, age, value, nationality, overall, and potential. The results are sorted by potential in descending order. The first 5 rows are shown:

#	id	name	age	value	nationality	overall	potential
1	211110	P. Dybala	24	€89M	Argentina	89	
2	208389	J. Oblak	25	€68M	Slovenia	90	
3	222492	L. Sané	22	€61M	Germany	86	
4	205680	S. Veretout	24	€57M	France	87	
5	231443	O. Dembele	21	€48M	France	83	

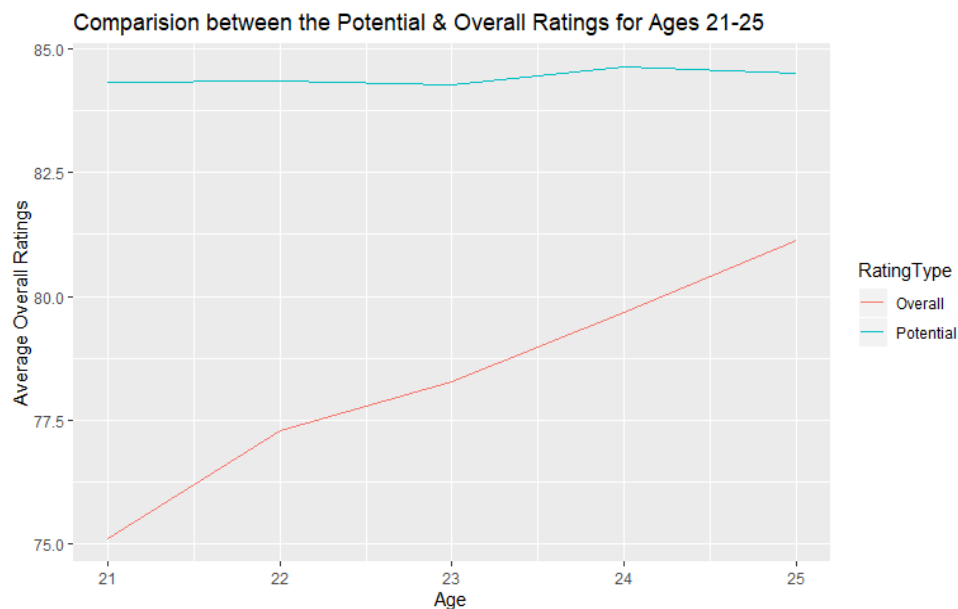
The image above shows the query that I applied to the data. I took the players in the age group of 20 and 25. Arranged the players in a descending order with respect to their potential and took the first 500 records. I saved the queried results to a file named Query1.csv. I loaded the Query1 file into R in order to perform exploratory analysis on the data.



I first decide to study the trend for the potential rating of the players in the age group 20-25. I could not see a very highly different pattern within the same. This implies that ideally all the players within the said age range has expectedly similar average potential ratings.



But having a look at potential ratings only paints half the picture, hence I decide to view the overall ratings of the players in the age group and not only the potential ratings. The overall ratings of the players in the age group varies vastly between 75 and 81. There is a visible trend in the graph, which shows that the average overall ratings of the players is higher for the age group of 24-25 when compared to 21-22. This proves the common assumption in football that as the age of the player increases, his maturity and control in the game increases, which naturally leads to an increase in rating till a certain threshold age.



While observing the graphs for potential ratings as well as overall ratings, I could observe that the scale for both the values was different. In order to compare both on a similar scale, I plotted the values on the same graph. The above graph helps comparison such that the potential rating for all the age groups is between 84 and 85, but the actual overall rating is a bit lower than that in all cases.

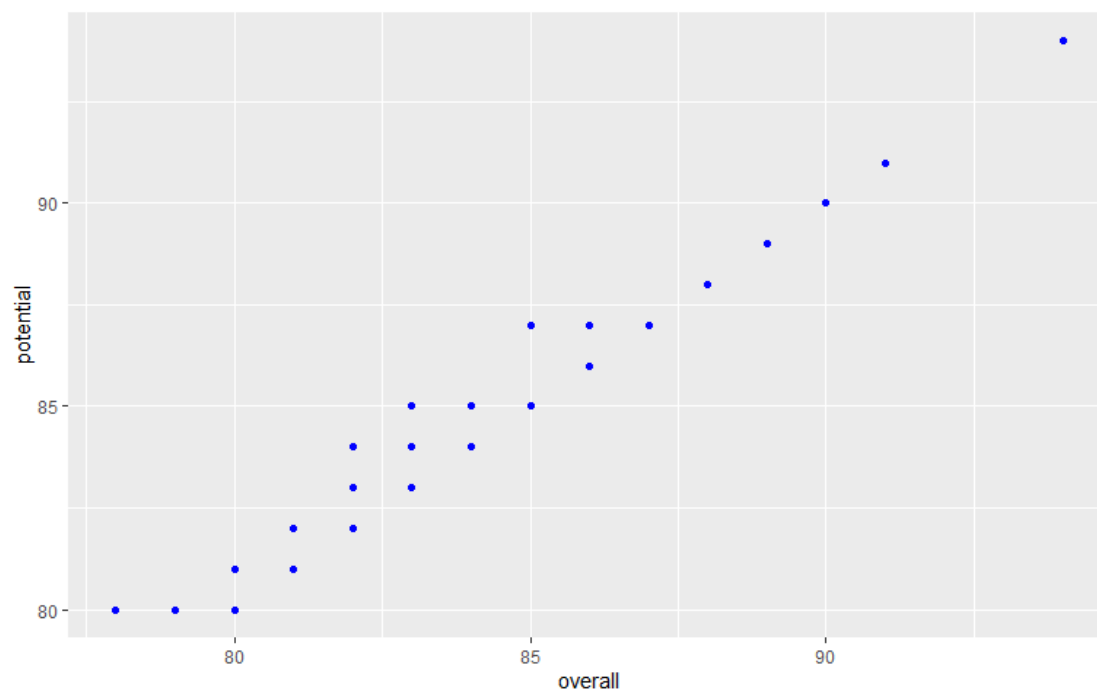
Query2 –

The screenshot shows a web-based data query tool interface. The main area displays a SQL query for 'Query2' which selects player details where potential is greater than or equal to 80, ordered by age in descending order, limited to 250 results. Below the query, a table of 250 query results is shown, listing players like G. Buffon, S. Sorrentino, and A. Barzagli. The interface includes a sidebar with project files and a right-hand panel with query details and schema information.

```
1 SELECT name,age,value,nationality,overall,potential,club
2 FROM data
3 WHERE potential >= 80
4 order by age DESC
5 LIMIT 250
```

	name	#	age	value	nationality	#	overall	#	potential	
1	G. Buffon		40	€4M	Italy		88		88	Paris S
2	S. Sorrentino		39	€1M	Italy		80		80	Chievo
3	Aduriz		37	€8M	Spain		82		82	Athleti
4	A. Barzagli		37	€4.2M	Italy		84		84	Juventa
5	Castillas		37	€1.5M	Spain		82		82	FC Port
6	P. Cech		36	€3M	Czech Republic		82		82	Arsenal
7	Diego López		36	€2M	Spain		80		80	RCD Esp
8	Z. Ibrahimovic		36	€14M	Sweden		85		85	LA Gala
9	Joaquin		36	€8M	Spain		81		81	Real Be

In this query, I selected the top 250 players with respect to age, such that the potential ratings of the players were above 80.



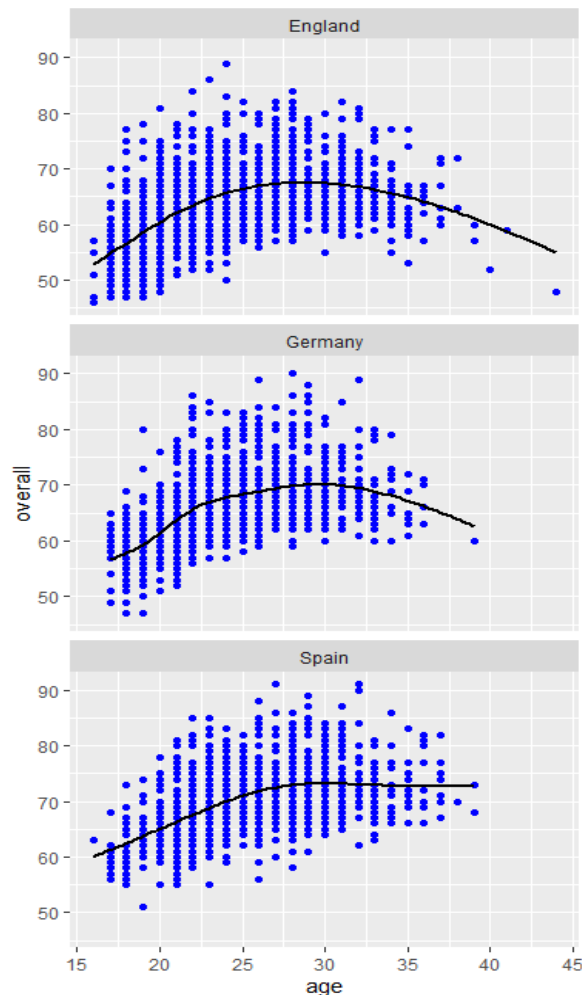
In order to understand the trend between the actual overall rating of the player and the potential ratings of a player, I took the graph of subsets of players, so that the graph could show the trend clearly. From the graph, we can conclude that as the ratings of a player increase, his overall rating and potential ratings are very close, barring a few outliers.

Query3 –

The screenshot shows a data analysis tool interface with a sidebar on the left containing project files and datasets. The main area displays a SQL query for 'Query3' which filters players by nationality (England, Spain, Germany) and orders them by potential in descending order. Below the query, a table shows the first 9 results, including player names, ages, values, nationalities, overall ratings, and potentials. A right-hand panel provides details about the query, such as when it was last updated, the creator's name, and the project schema.

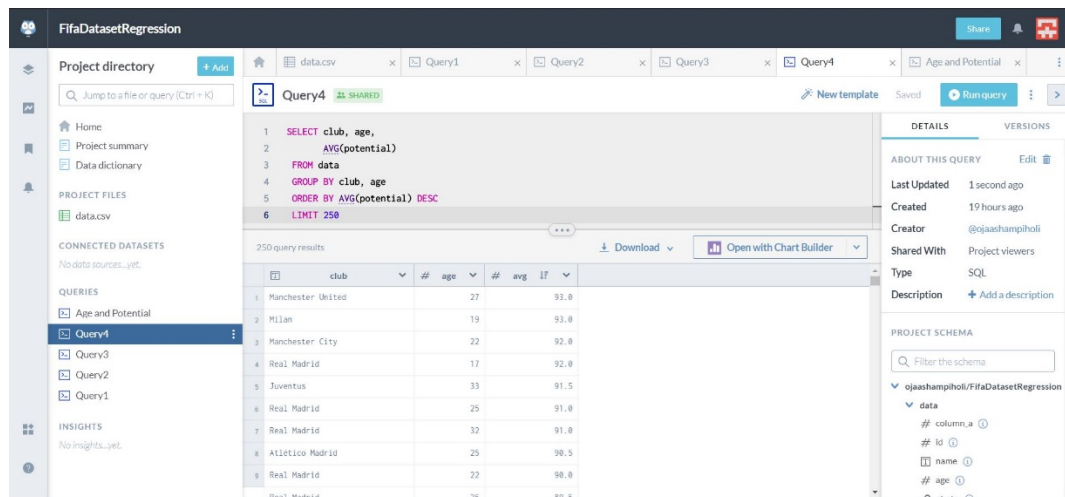
	name	#	age	value	nationality	#	overall	#	potential
1	De Gea		27	€72M	Spain		91	93	Manchester
2	M. ter Stegen		28	€58M	Germany		89	92	FC Barcelona
3	L. Sané		22	€51M	Germany		86	92	Manchester
4	Marco Asensio		22	€54M	Spain		85	92	Real Madrid
5	Isco		26	€73.5M	Spain		88	91	Real Madrid
6	H. Kane		24	€83.5M	England		89	91	Tottenham
7	Sergio Ramos		32	€51M	Spain		91	91	Real Madrid
8	Kopa		23	€28.5M	Spain		83	91	Chelsea
9	T. Kroos		28	€76.5M	Germany		90	90	Real Madrid

In this query, I took players from the three major countries that participate in football events in all professional tiers. I ordered the players in descending order by their potential ratings. This query was intended to study the similarities and differences between the major football playing nations.



We can observe that England has the greatest number of players, closely followed by Germany and then Spain. The overall trend is same for England and Germany if we study overall ratings with respect to age. Both these nations have an increase in trend till 30 years of age and then subsequently decline as the age increases. For Spain, however the case is a little different, the trend has in increase till the ages of 30, however after that age it does not go down, but it stagnates at approximately the same level.

Query4 –



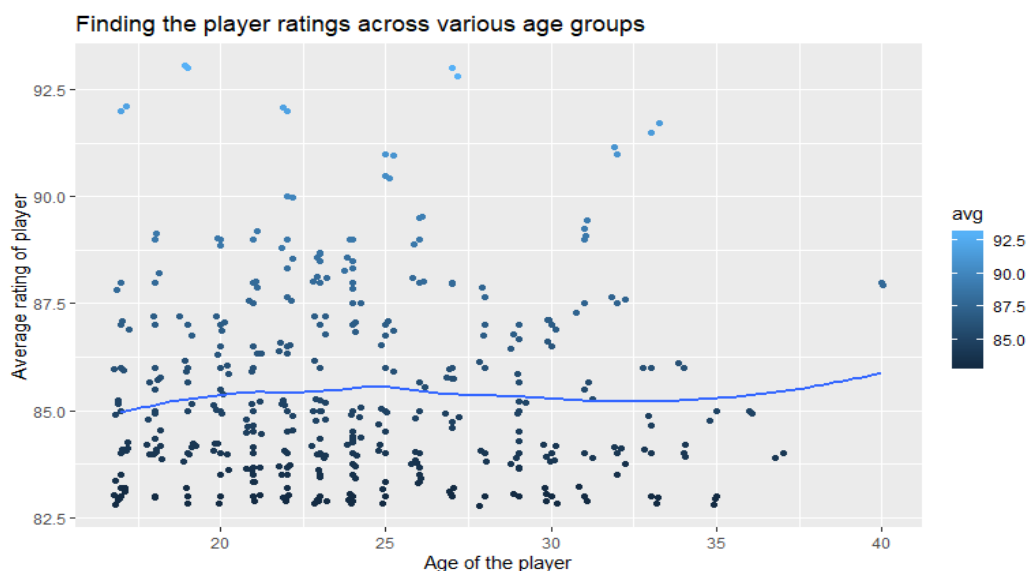
Query4 SQL:

```
1 SELECT club, age,
2     AVG(potential)
3 FROM data
4 GROUP BY club, age
5 ORDER BY AVG(potential) DESC
6 LIMIT 250
```

250 query results:

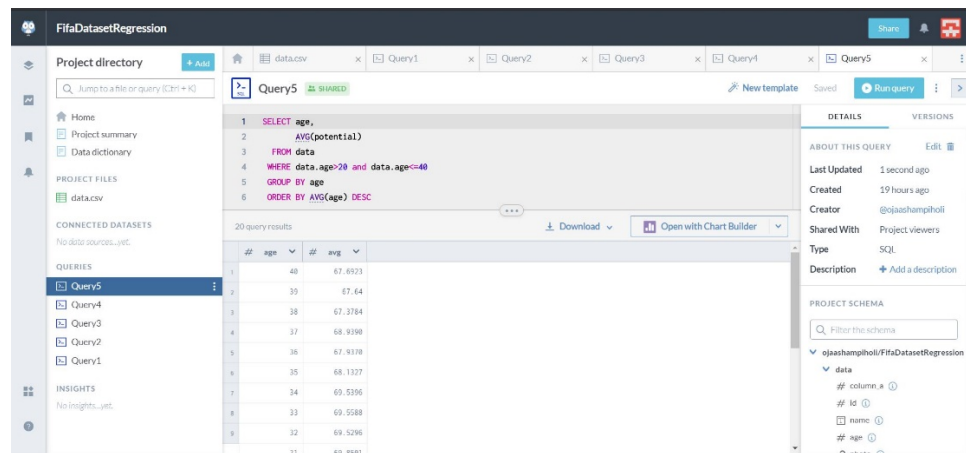
club	#	age	#	avg	if
Manchester United		27		93.0	
Milan		19		93.0	
Manchester City		22		92.0	
Real Madrid		17		92.0	
Juventus		33		91.5	
Real Madrid		25		91.0	
Real Madrid		32		91.0	
Atletico Madrid		25		90.5	
Real Madrid		22		90.0	

In this query, I filtered out the data for players whose potential ratings have been grouped by their clubs and ages. Group By command was used in this query.

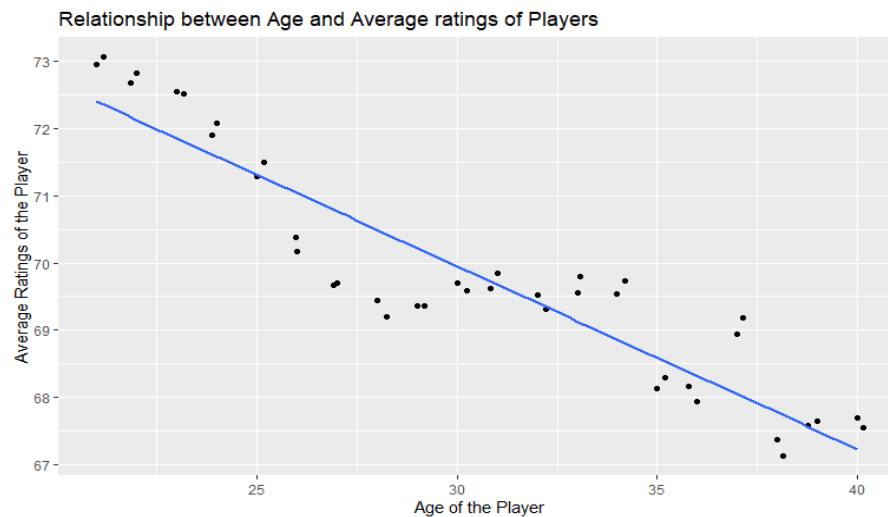


The graph above very clearly shows that if we have a look at the average ratings of the players with respect to their ages, most of the players have a high rating at 25 years. It is a known fact that the ages between 23 and 27 are considered golden years of ages in football. And this can be seen very clearly from the graph above.

Query5 –



In this query, I tried to find the expected potential ratings for the players between 20 and 40 years of age.



I expected the trend to be a negative sloping graph, such that the assumed potential for a player decreases with age. This assumption of mine was confirmed from the graph that is drawn above.

After the analysis on the query data was complete, I decided to move around to actual predictive model designing. The designing and testing of the predictive model was done in jupyter notebook attached alongside. The data was loaded onto the notebook and then I applied exploratory analysis on the data. This showed me the most common ages of players, the nation having most players, most common ratings of the players in the dataset. I then had a look at the null cells in the data. Since the null cells caused more degradation to the model than to keep them, I decided to remove the null cells completely. After this was done, I had to remove the Euro Symbol from the Wage, Value and the Release Contract columns. Further, the columns had the notations K and M for 1000 and 1000000, respectively. This helps us to visualize the monetary aspects of the data very easily. I had an assumption that the market value of a player and his release contract are directly proportional to each other. This assumption of mine was proven from the graph that I have shown in the notebook.

Results from the Model -

Next, some of the cells contain invalid values, which are not null. In order to handle such cases, I decided to fill such numbers with mean of the feature values using the Imputer function. This function calculates the mean of all the cells and replaces the invalid cells with the mean of the feature values. This ensures that the data integrity is maintained, and maximum data is used for the predictive models instead of dropping additional records. This is the data cleaning step that is followed in big data pipeline. This step is extremely important when we deal with any type of data. The better tricks that we use in this step ensure that we get better prediction results from the models.

After this step, the next step that is done is to build the models. This involves deciding the parameters of the model, training the model on a part of the data known as training test and eventually testing the data on testing part of the data. In this step, the models are first defined and the parameters for the models are set to standard models. After this, the data is split into 75:25 portion, where 75% is considered as training data and the rest 25% is considered as testing data. This 75% of the data is used to train the predictive model. In regression, the model predicts the values for the target variables, which is not a fixed set of labels. This means that all the values that are predicted in a regression problem may be unique values, not like classification problem which has fixed labels (usually binary decisions). The training data trainX and the corresponding target variable data called trainY are given to the model in order to train the model. Then, the 25% of the data kept for testing testX is given to the model in order to obtain predicted Y outputs. These predicted Y are compared to the testY samples which are the ground truth labels in order to determine the accuracy of the regressor model.

4 regression models were applied to the data in this problem, in order to determine what models performs the best on this data samples. The names of the models and accuracies for the models is given in the table as below:

REGRESSOR MODEL	R ² SCORE
LINEAR REGRESSOR	0.85
SVM REGRESSOR	-0.1
NEURAL NETWORK REGRESSOR	0.85
DECISION TREE REGRESSOR	0.92

In the analysis of regression models, there are two parameters which can be used to determine the accuracy of the model. The first way is to use the RMSE (Root Mean Squared Error) and the other way is to use the R^2 statistic score. In case of the RMSE, the score is nothing but the sum of squared error between the predicted and actual Y values. The RMSE should be as low as possible for the model. However, in case of the R^2 score, it is nothing but the score of how well the regression model fits the data. R^2 score lies between 0 and 1. 0 signifies that the model does not fit the model at all. While 1 implies that the model perfectly fits the data. Higher score on the R^2 score implies that the model is better. In this problem, I have used the R^2 score to compare the models.

For the linear regression model, the model fits 85% of the data. This is the maximum accuracy that the linear regression model can fit the data. Next, I tried the SVM Regressor model on the data. The SVM Regressor gives a negative R^2 score. This implies that the model does not fit the data at all and performs worse than the linear regression model. The third model is the Neural Network regressor model. This model is run for 500 epochs only, I had to place this restriction on the number of epochs because of the CPU restrictions on my computer. The neural network model gives an accuracy score of about 85% as well. The next and the last model that I tried on the data was Decision Tree Regressor. The decision tree regressor model also uses a cross validation on the training set in order to find the expected accuracies. The cross-validation score comes out to be around 92%. Hence, for the above problem and the dataset, the Decision Tree Regressor performs better than all the other models.

Challenges Faced:

1. The first challenge that I faced was that the data had 89 features for each record. I had to narrow down the number of features in order to ensure that my model only predicts the results based on important features and not on all the available features.
2. Another challenge that I faced was that the data had a lot of irregularities in the features. The wage, value and release contracts had a Euro symbol before the value and the notations for thousands and millions instead of the numerical values.
3. The next big challenge was the handling of missing data. I had to handle the missing data if the model had to be as accurate as possible.
4. The data that was obtained after query was to be used for visualization in R, hence, the copies of the results of the queries had to be saved.
5. The accuracy of the model was not up to par in the first few runs, hence hyperparameter tuning had to be done in order to improve the accuracy of the model.

Conclusion:

In this project, I got a firsthand experience of creating a predictive analysis model pipeline from scratch. In this project, I had to pre-determine the stages of the pipeline and then act accordingly in the execution stages. The tools and technologies that I had to use in this project had to be setup. The major challenges that I faced, took some of my time but in the end, it helped me to gain an integral understanding of the problems that are commonly faced while dealing with big data. This project also helped me to get a hands-on experience with machine learning models and their uses in the real-world problems.

List of Attachments:

- Fifa dataset (18207 rows x 89 columns)
- Pipeline (jpeg file) – Shows the pipeline that was followed during this big data analytics and predictive modelling project.
- Images Query 1-5 (jpeg files) – Show the code for the queries that were run on the data.world website to get desired subset of data.
- Query Result CSV Files (1-5) – Contain the data results obtained after running the queries.
- Fifa_Data_Query RMD File – Contains the R code used for the exploratory analysis of query results csv files.
- Fifa-ValuePrediction Notebook – Contains the code for exploratory analysis, data cleaning and preprocessing and model training, hyperparameter tuning and model testing and accuracies.

References:

- Dataset download link <https://www.kaggle.com/karangadiya/fifa19>
- Predictive Modelling https://www.sas.com/en_us/insights/analytics/predictive-analytics.html
- Big Data Pipeline <https://www.colaberry.com/data-science-predictive-analytics-pipeline/>
- Data.world website to upload data to aa private repository and run queries on the data <https://data.world/>
- Data.world help website <https://help.data.world/hc/en-us/articles/360008853693-Getting-started-guide>
- Tidyverse library in R which is used to handle big data <https://www.tidyverse.org/>
- Ggplot2 in R for exploratory analysis <https://ggplot2.tidyverse.org/>
- RStudio Installation and Guide <https://rstudio.com/>
- Jupyter Notebook Installation and Guide <https://jupyter.org/>
- Simple Imputer to handle missing data <https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html>
- Train test split of data https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
- Regression Models https://scikit-learn.org/stable/supervised_learning.html