**Predicting Player Ratings of European Soccer Players**

**by: Mehul Goel**



**Introduction and Background**

I have been interested in soccer from a young age and love watching the sports. I even tried to pursue soccer at a higher, collegiate level and have had ratings associated with my level of play. Due to my interest in this data, I worked on building this project of predicting ratings for European Soccer Players.

**Datasets Used**

For the purpose of this project, I will be only using the Player\_Attributes table extracted [from this SQL dataset on Kaggle.](https://www.kaggle.com/hugomathien/soccer) The Player\_Attributes table contains the following 39 relevant columns: Date of row entry, Overall\_rating (what I was trying to predict), Potential (their potential rating they can achieve), Preferred Foot, Attacking Rating, Defensive Rating, Crossing, Finishing, Heading Accuracy, Short Passing, Volleys, Dribbling, Curve, Free Kick Accuracy, Long passing, Ball Control, Acceleration, Sprint Speed, Agility, Reactions, Balance, Shot Power, Jumping, Stamina, Strength, Long Shots, Aggression, Interceptions, Positioning, Vision, Penalties, Marking, Standing Tackle, Sliding Tackle, Diving, Handling, Kicking, Positioning, Reflexes.

I found the data particularly interesting because of how many statistics there were available for each player and I wanted to understand how each of these attributes affects or reflects on predicting a player’s overall rating and understanding what makes them a good soccer player.

**Statement of Purpose**

The purpose of this project is to understand which factors are most influential in determining how good someone is at soccer and also to predict which professional player will be good at soccer. This can be accomplished by creating a model that can accurately predict the overall quality of a player given any of the relevant data columns that I have defined above. This type of model would be very valuable to managers of professional soccer teams to help teams assemble the best team by picking players who are predicted to be good. Additionally, it could help coaches understand how much attention each player needs by understanding their rating ahead of time.

**Main Objectives**

There are two main objectives for this project.

1. Regression Objective: correctly predict 85% of the player's overall rating score within a plus or minus 3 rating interval.
2. Classification Objective: correctly classify 85% or more of the players into tiers based on their overall quality, which I will define in the Creating Overall Quality Column section.

**Data Cleaning and Preparation**

**Multiple entries per player**

I realized that since there are multiple entries for each player due to the date column, it would be better to group the data frame for each player id and take the row with the player’s highest overall rating.

**Creating Overall Quality Column**

I created an ‘overall quality’ column that classifies the players into five total tiers of players. The highest tier will be denoted by a 1, while the lowest tier will be denoted by a 5. The tiers are defined within the following overall rating ranges:

Overall Quality Scores:

* 1 → Overall Rating >= 82
* 2 → Overall Rating >= 73, < 82
* 3 → Overall Rating >= 67, < 73
* 4 → Overall Rating >= 60, < 67
* 5 → Overall Rating < 60

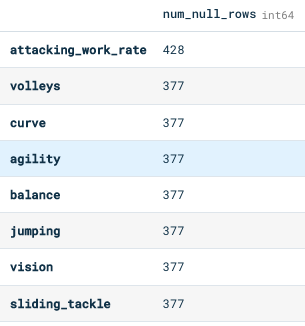
These overall quality ranges were arbitrarily chosen to somewhat match the distribution of the ‘overall\_rating’ column. I used the value\_counts() method to accurately break down the data into 5 evenly classification rows.

**Splitting up data into testing and training**

I split the data using an 80-20 train-test split and determined the project’s accuracy by predicting it on the test data.

**Data Pipeline Creation**

The player\_df dataframe that I used throughout the project had multiple null values in the rows which would not allow me to run any proper classification or regression algorithms.



**Figure 1:** Null values in multiple data columns

In order to address this, pipelines were made to handle all columns within the dataset. This allowed for flexibility in the project in case I wanted to add more data columns in the future. Two different pipelines were used for the classification and regression portions of the project. For null values in numerical data columns, a Simple Imputation Transformer was used with a strategy of “mean.” This means that it would take the mean value of all other scores in that column and set the null value to that mean.

**One Hot Encoder**

One Hot Encoder was used to change all categorical data columns into numerical values which allowed the machine learning models to run and interpret the data correctly.

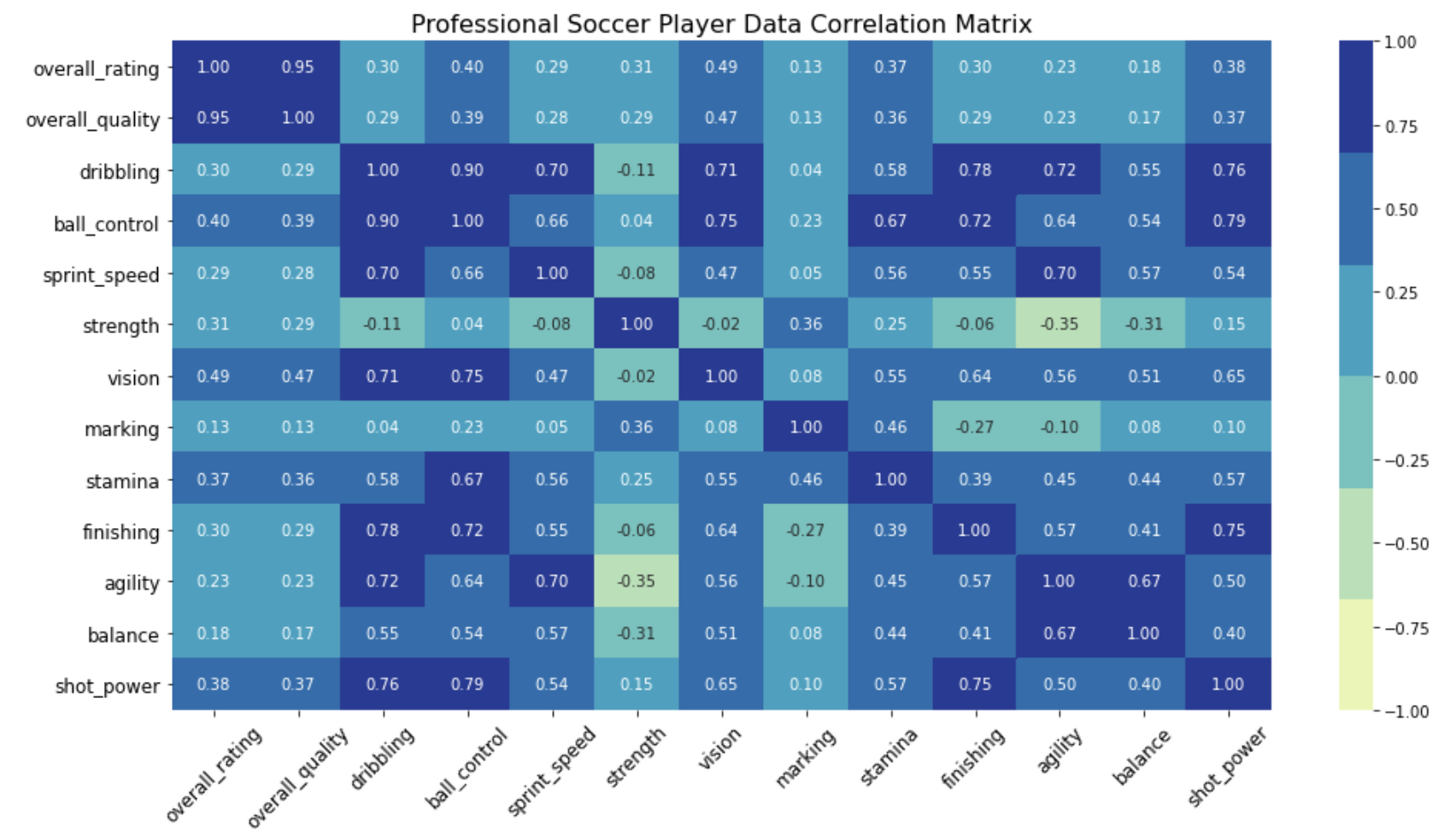
**Standard Scaler**

A standard scaler was used to make sure all numerical data columns had means of 0 and standard deviations of 1. This allowed my machine learning models to run and interpret the data correctly.

**Exploratory Data Analysis**

**Correlation Matrix**

After dropping the unnecessary columns (such as Date) which I would not be using for this project and creating the ‘over\_quality’ categorical column, there were 39 columns total in the player\_df dataframe. Initially, I tried to make a correlation matrix with all 39 columns, but it was too much information and could not be interpreted in one visualization. So, I decided to only focus on the key 15 of the 39 variables for the correlation matrix to make it easier to process. I chose these 15 variables based on what I researched about FIFA Ratings and found were the most important factors from my own personal experience as a soccer player. The 15 variables I chose were overall\_rating, attacking\_work\_rate, defensive\_work\_rate, heading\_accuracy, dribbling, ball\_control, sprint\_speed, strength, vision, marking, stamina, finishing, agility, balance, and shot\_power.

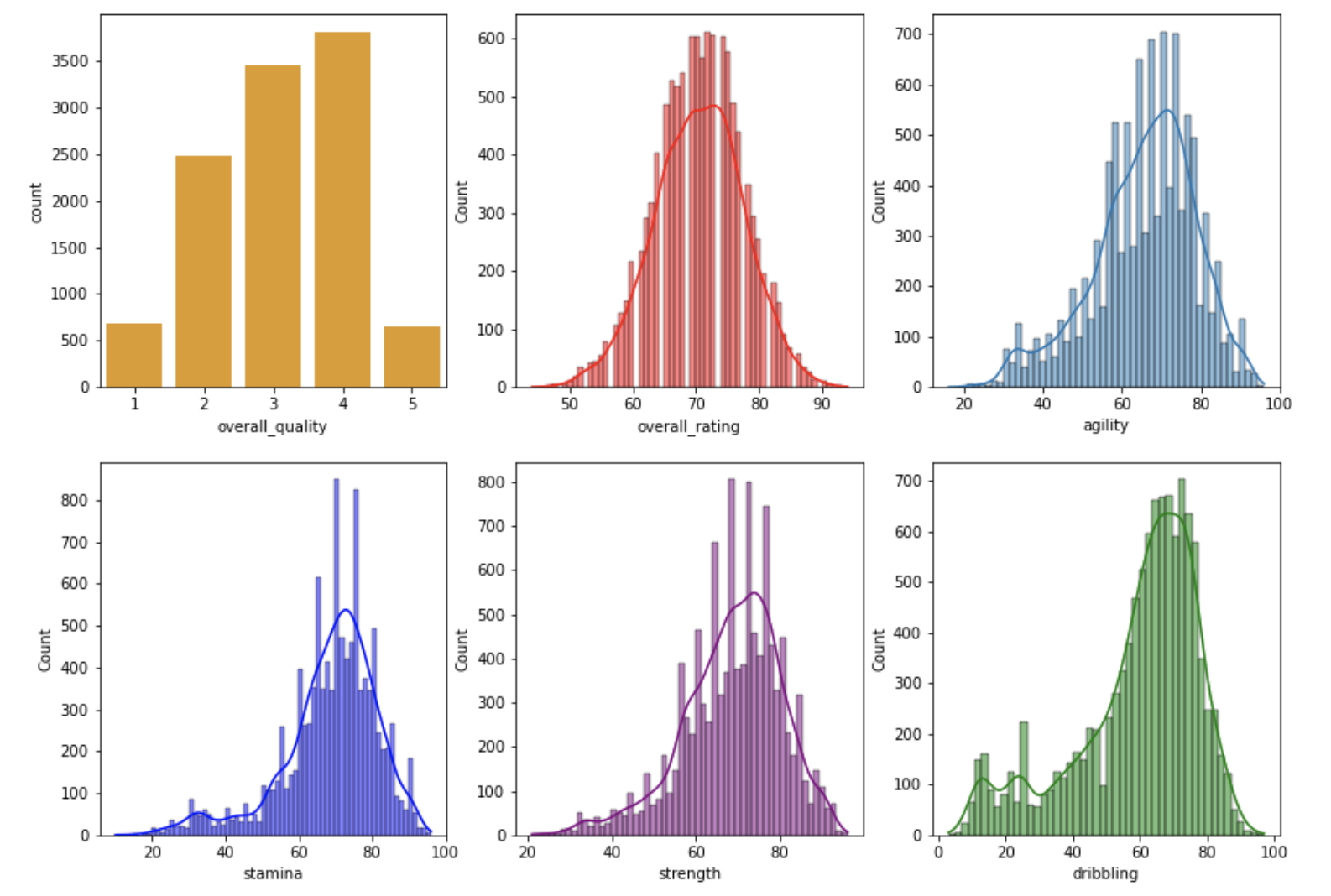
****

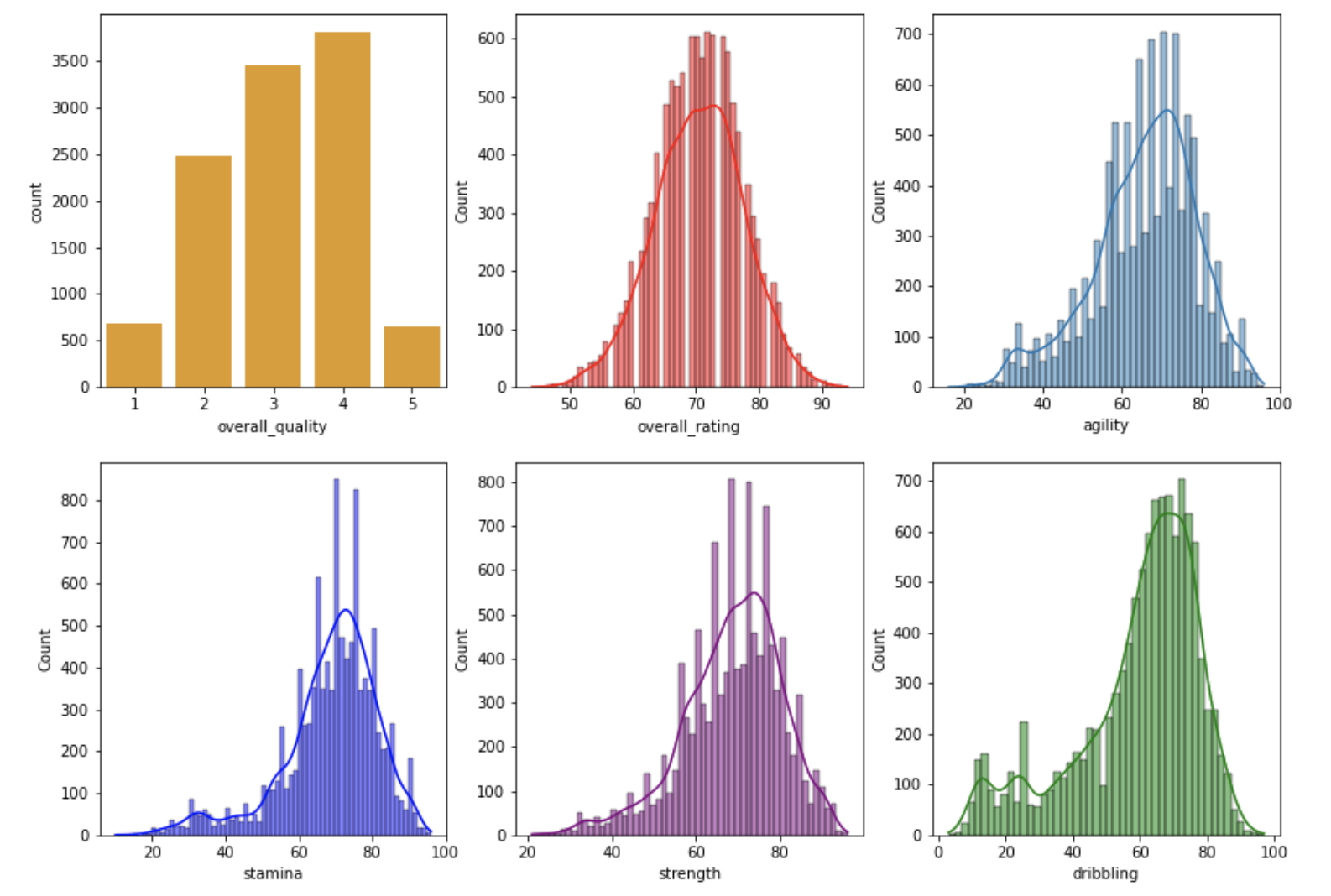
**Figure 2:** Data Correlation Matrix for Chosen Variables

From the correlation matrix, I found that the variables dribbling and ball\_control have the strongest correlation of 0.9. The variable pairs with the next strongest correlations are ball\_control with shot\_power, dribbling with finishing, and dribbling with shot\_power. The strongest negative correlation was -0.35 between strength and agility, then -0.31 between strength and balance, and then -0.27 between marking and finishing.

I was interested in the overall\_quality of a player so I focused on which variables were most strongly correlated to it. The overall\_rating column was used to create the overall\_quality column so there was a strong correlation between the two, but I ignored it because I was more interested in how the other variables impacted overall\_quality. I found that the variables with the strongest correlation to overall\_quality, excluding overall\_rating, in order are vision, ball\_control, shot\_power, and stamina.

**Visualizations of Variables of Interest**

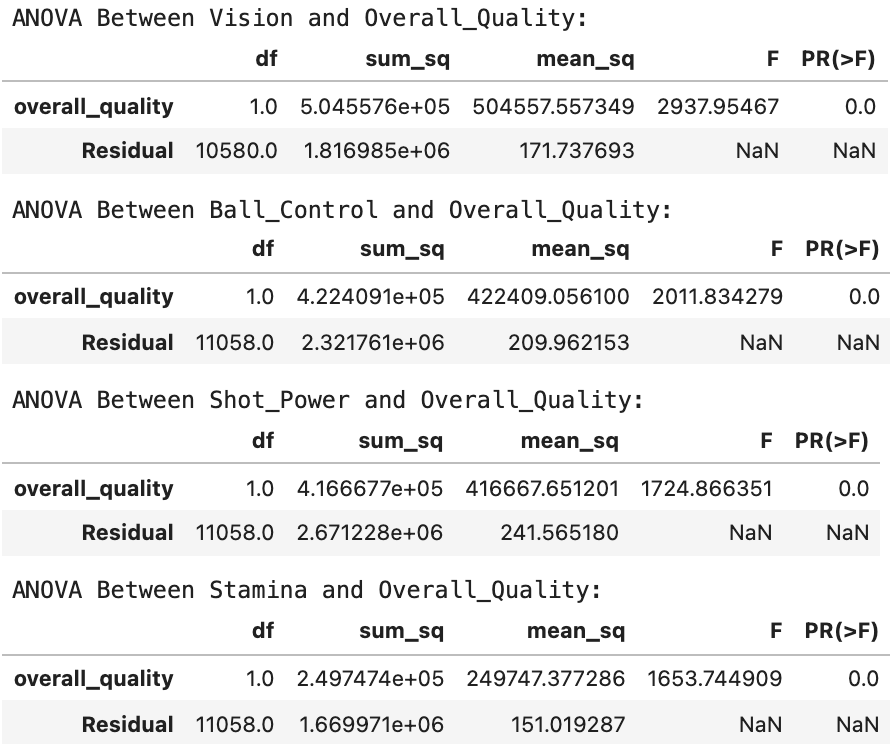


**Figure 3:** Histogram Plots of Main Data Variables

The visualizations above show the distribution of values for the variables overall\_quality, overall\_rating, agility, stamina, strength, and dribbling. I created the overall\_quality variable to be a categorical version of the overall\_rating variable, so it makes sense that they have similar distributions. The four other plots are included because they are the variables of interest related to overall\_quality that were found from the correlation matrix. The distribution of the agility, stamina, strength, and dribbling variables are all slightly skewed to the left. The majority of the values for all four of those variables fall within the range of 60-80.

**Analysis of Variance (ANOVA)**

ANOVA is a statistical technique that is used to check if the means of two or more groups are significantly different from each other. I wanted to use ANOVA to investigate if there is a significant difference in scores of my variables of interest between the five groups of different players based on their overall\_quality score. I conducted four ANOVA tests between overall\_quality and each of the variables of interest (vision, ball\_control, shot\_power, stamina).

****

**Figure 4:** ANOVA Table to Test Differences in Tiers of Players

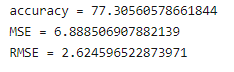
The null hypothesis for each of the ANOVA tests was that there is no difference between the mean of a player’s vision, ball\_control, shot\_power, or stamina for the five groups of players based on their overall\_quality. For all of the four ANOVAs, the p-value is less than the generally accepted significance level of 0.05. Therefore, in all four cases, I rejected the null hypothesis and can conclude that there is a statistically significant difference in the means of a player’s vision, ball\_control, shot\_power, and stamina based on which group of overall\_quality they are in. In fact, all the p-values are either zero or extremely close to zero, which means the results are highly significant.

**Main Findings**

**Regression**

Linear regression is an approach for modeling the relationship between a dependent variable and one or more independent variables. I used multiple linear regressions where the dependent variable was overall\_rating to understand how the variable depends on other factors. The pipeline I created was used to prepare the data and fit a linear regression model to formulate predictions. I also worked towards creating an accuracy function to see whether the predicted overall rating was within plus or minus 3 of the actual overall rating.

From my analysis, I found that there was only 77.3% accuracy. Furthermore, the MSE score was around 6.9, implying that the average distance between the observed overall rating and the predicted ratings was around 6.9. Additionally, the RMSE score was 2.6, which implies that this model will predict a player’s overall rating wrong around 26% of the time. That is a fairly high frequency to have the model be incorrect.

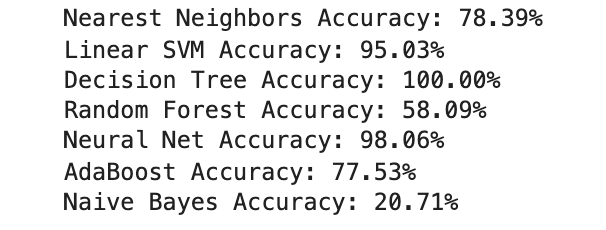


**Figure 5:** Regression Accuracy Scores

My findings suggest that the model is moderately good at predicting a player’s rating, given I have access to all the other data columns in the Player Attributes column.

**Classification**

Classification is a machine learning technique used to predict something (Y) given an input (X). In this project, I was interested in predicting the overall\_quality of professional European soccer players using the other variables within the dataset. To find which classification methods produced the most accurate results, I evaluated the accuracy of seven different kinds of algorithms: Nearest Neighbors, Linear SVM, Decision Tree, Random Forest, Neural Net, AdaBoost, and Naive Bayes. The Linear SVM, Decision Tree, and Neural Net classifiers were the most accurate at predicting a player's overall\_quality. The results are shown below in Figure 4.



**Figure 6:** Classification Accuracy Scores

* Linear SVM - it is a supervised machine learning technique that analyzes data for classification. The algorithm creates a line that separates the data into different classes. It relatively works well when there is a clear margin of separation between classes. In this model, all classes were separated from each other, and there was no overlapping, which may be a reason for its high accuracy with the data.
* Decision Tree - it is a supervised machine learning model where the data is continuously split according to a certain parameter and classified into nodes. It works well when there are certain features that contribute more to the predictions than others. In this case, data columns like “potential” may have had a high impact on calculating ratings. This may be the reason why it has such high accuracy with the data.
* Neural Networks - it is a set of algorithms that are modeled loosely after the human brain and they are designed to recognize patterns within the dataset. It iterates for a predetermined number of iterations, called epochs. After each epoch, the cost function is analyzed to see where the model could be improved. This technique works well with large amounts of data so that might be why it performed so well with the dataset.

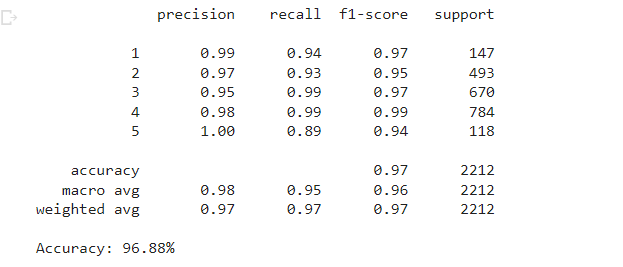
I then built a VotingClassifier, to get better predictions for player\_ratings. A VotingClassifier is a machine learning estimator that trains from various base models and then builds up predictions on the basis of aggregating the findings of each base estimator. For my VotingClassifier, I used the above-mentioned seven classifiers to get the final prediction accuracy, recall, precision, and f1-scores.

Final accuracy of 96.88%

Precision of 0.97

Recall of 0.97

F1-score of 0.97



**Overall Results**

Although the regression model’s accuracy only reached around 75% prediction accuracy, which was less than my 85% objective, the classification portion of this project was successful. I completed the objective and had higher than 85% accuracy.

**Statement of Limitations**

While working on this project, there were multiple limitations that I had to place to get the dataset to work effectively:

* There were nearly ~40 relevant data columns used with hundreds of null values. This meant that I had to use imputation to fill in the null values, leading to a much lower accuracy score. By sourcing more full data, the models’ accuracy for classification and regression would most likely improve.
* Each player had about seven rows of data attributed to them. This made the dataset very large. Thus, as a limitation, I decided to only include one row of data per player. My method for choosing which row to include was to find the row with the highest overall rating for each player. Doing this decreased the size of the data frame, making it easier for me to conduct my analysis, however, it may have resulted in lower model accuracy.
* I wanted to create a pairplot for all variables, however, it took too long to execute. Instead, I had to create hist or count plots for the variables I determined to be most relevant.
* The data was only limited for European professional soccer players for a set number of years. Thus I could not examine the models on data from professional soccer players in other parts of the world.

**Challenges Encountered**

* It took a lot of time to figure out how I should handle the data since there were multiple rows of data per player. Once I decided that I was going to take the record with the maximum overall rating per player, it was difficult to implement due to problems with null data types.
* The large number of variables in the dataset created challenges throughout the data exploration phases of the project. I couldn’t create visualizations that encapsulated all the variables because it would be too confusing. Thus, I had to spend time thinking of the best way to remove some of the less relevant variables, while still creating effective visualizations showing important relationships and information.
* The large number of variables also made it hard to understand which variables were influencing the predictions I created.

**Recommendations for Future Work**

* The main next step for the project would be to use the model I have trained with the data to predict the ratings and quality of other professional soccer players. It would be interesting to see if the model could predict a player’s potential rating right before a season has ended and see if it matches the actual rating agencies within a reasonable margin of error.
* Additionally, it would be good to understand the 5 least influential variables. After identifying those least important variables, it may be interesting to remove them to tune and increase the accuracy of the model.
* While the project only used the Player Attributes table, it would have been interesting to tune the classification and regression models using more information from other tables within the European Soccer database I accessed, such as the “Team Attributes” or “Match” tables.
* Lastly, using more complex imputation methods to fill in null values could be explored. For numerical values, I decided to use the mean of the specific column to fill its null values. However, maybe there are other strategies such as using median or creating my own apply functions that do a better job of accurately filling the null values.