Linear regression and application in analysis and predicting sport clubs' scores and rankings

Nguyen Duy Thang/2001040185 2001040185@gmail.com
Hanoi University
Hanoi, Viet Nam

Tutors: Mr.Bui Quoc Khanh

Abstract

In this research paper, I aim to predict the scores as well as the rankings of football clubs based on the collected data including information about the team, scores, rankings, features related to football players such as age, nationality... in the period from 2007 to present in the Turkey league. The data and information about the research subjects in the article are quoted from scorestats.com. The main algorithm used in this study is linear regression, the information about the algorithm is quoted from the book "Pattern Recognition and Machine Learning" and medium.com. Linear regression algorithm will be used to compare the features that can affect the score as well as the ranking of a club, thereby making predictions about the score. the number a club will achieve as well as their ranking in the upcoming season.

Keywords: Linear Regression, Turkey league

1. Introduction

Nowadays with the continuous development of technology and especially machine learning technology, the applications of machine learning are being applied more and more popular in daily life. The biggest strength that machine learning algorithms bring is that they can automatically select the features in the data set that are best and most suitable to solve the problem you are facing [M.Bishop (2006)]. This is called the feature selection process which directly affects the quality of a machine learning model. In fact, machine learning is about determining the most accurate set of features to solve a problem. In large projects there will be teams of dozens of people who improve accuracy to more than 90 percent by trying different feature sets.

In this report, linear regression algorithm will be used as the basic model for analyzing and predicting the scores of football clubs. Before diving into how to apply algorithms in our project, we need to learn a little bit about the data used for the research. The data is taken from the Turkey Super Futball League season from 2007 to 2015 including 2 excell tables. The first table will include club name, player name, player age, foreign, multinational and market value. The second table contains the club names and their scores.

After collecting the above data, we will build a linear regression algorithm to analyze the relationship between features and make predictions. In the predictions session, we will predict the scores and rankings of the teams through the Kendall's Tau method. However, first we need to understand what linear regression is and how it works.

2. Literature Review

2.1 Linear RegressionRegression

Before diving into the coding and application of linear regression algorithm, we need to understand what linear regression is and how it analyzes and predicts data.

First, linear regression is an algorithm based on supervised learning for machine learning. This algorithm is mainly used for analyzing the relationship between different variables, thereby making predictions. To find relationships between different variables, algorithm uses formula for finding a equation of a line in math: y = ax + b. Linear regression will use the equation of the line to analyze the data collected through the coordinate plane Oxy [Gupta (2022)]

We can imagine the output of a problem would be $\hat{y}$ calculated based on a given independent variable x_i with a certain ratio, these are the coefficients w_i. These x_i values can be written as vector $\mathbf{X}$, and w_i can be written as vector $\mathbf{W}$. The optimization of the Linear Regression model is to find the vector $\mathbf{W}$ such that input variables vector $\mathbf{X}$ we can calculate the output $\hat{y}$. The formula algorithm linear regression can be configured as follows:

$$\mathbf{X} = [x_1, x_2, x_3, \ldots, x_n] \tag{1}$$

$$\mathbf{W} = [w_0, w_1, w_2, w_3, \ldots, w_n]^T \tag{2}$$

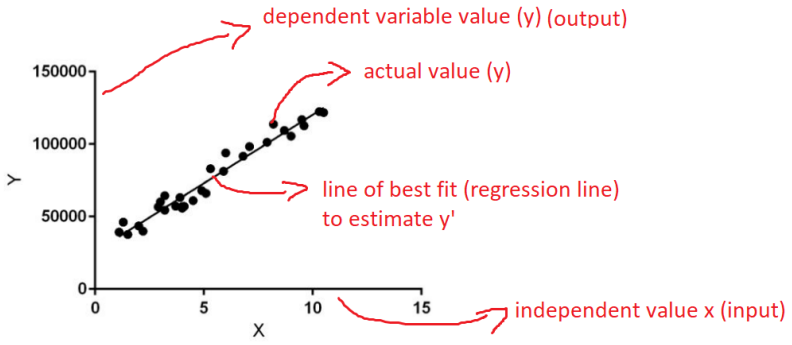
From (1.1) and (1.2) we get:

$$\hat{y} = f(\mathbf{X}) \approx y \tag{3}$$

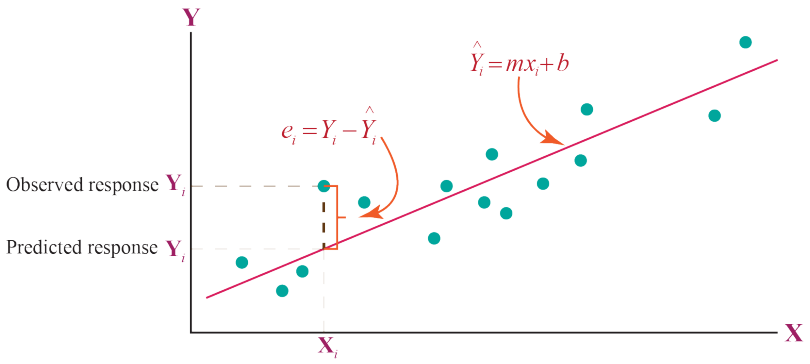
In there,

$$f(\mathbf{X}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots + w_n x_n \tag{4}$$

These two types of variables will be visualized through a graph with "y" as the vertical axis and "x" as the horizontal axis.



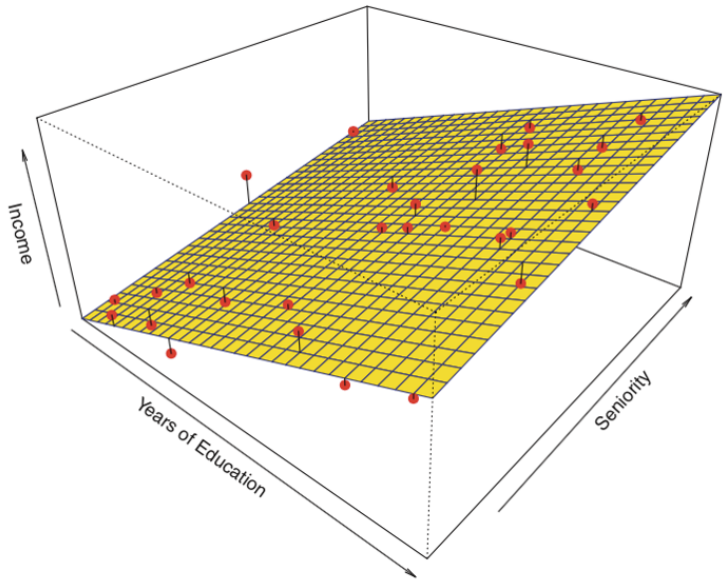
Basically, the result predicts $\hat{y}$ and the actual results "y" will have a certain error. The task of the problem now will be to find the parameters w_0, w_1, w_2,... so that the errors are minimal. Now we will represent the parameter as a column vector $\mathbf{W} = [w_0, w_1, w_2, w_3, \ldots, w_n]^T$, and input $\mathbf{X}$ becomes $\bar{\mathbf{X}} = [1, x_1, x_2, x_3, \ldots, x_n]$. The calculation of the predicted output becomes $\hat{y} = \bar{\mathbf{X}}\mathbf{W}$. The prediction error will be $e = y - \hat{y}$.



We need to calculate so that the error between the predicted result and the actual result is minimal. This calculation will be done through the formula:

$$\mathcal{L}(\mathbf{W}) = \frac{1}{2}\sum_{i=1}^{n}(y_i - \bar{\mathbf{X}}_\mathbf{i}\mathbf{W})^2 = \frac{1}{2}|\mathbf{y} - \bar{\mathbf{X}}\mathbf{W}|_2^2 \tag{5}$$

In addition, linear regression can also be represented in 3D graphs:



2.2 Kendall's Tau method

We need to learn a little bit about Kendall's Tau method which will be applied into predicting session. So what is Kendall's Tau method?

In statistics, correlation refers to the strength and direction of a relationship between two variables. The value of a correlation coefficient can range from -1 to 1, with -1 indicating a perfect negative relationship, 0 indicating no relationship, and 1 indicating a perfect positive

relationship. Kendall's Tau is the which measures the relationship between two columns of ranked data [Glen (2022)].

The formula to calculate Kendall's Tau, often is as follows:

Kendall's Tau = (C-D) / (C+D)

where:

C = the number of concordant pairs; D = the number of discordant pairs

To understand more about this formula, you can see a specific example to understand how the method is calculated through the following link `https://www.statisticshowto.com/kendalls-tau/`

2.3 Review

This research paper was inspired by two research papers on the application of machine learning in football by Peter Mqoaie [Mqoaie (2022)] and Serkan Emre Elci [Elci (2018)], published on the Medium website. In particular, this study would like to cite dataset from Serkan Emre Elci's research as input data.

In the two studies mentioned above, they used a variety of methods to analyze and predict scores and rankings, including linear regression.

3. Implementation

3.1 Reading and Analysis input data

To build the model code, this study will use the Python programming language. First of all, we need to create the necessary Import required libraries for our program:

```
1
2  import time
3  import os
4
5  import pandas as pd # for dataframe operations.
6  import numpy as np #for linear algebra operations.
7  import seaborn as sns # data visualization library
8  import matplotlib.pyplot as plt # for plotting
9
10 from sklearn.model_selection import cross_val_score
11 from sklearn.externals import joblib
12 from sklearn.model_selection import train_test_split
13 from sklearn.linear_model import LinearRegression
14 from sklearn.metrics import mean_squared_error
15 from sklearn.metrics import r2_score
16
17 from scipy.stats import kendalltau
```

- pandas (panel data): open source library for fast data analysis and manipulation. It will be used for reading data from excell tables and analysising them.
- numpy: python's library allows to work with matrices and arrays efficiently. It will be used to read arrays of scores and ranks of clubs.
- seaborn: library will help with data visualization. We will use it to plot linear regression about relationship between features.
- matplotlib.pyplot: matplotlib is a library that will support graphing functions, pyplot is Matplotlib's module used to draw lines for graphs of functions.

Next, we need to install the Scikit-learn library (Sklearn). This is a library for machine learning programs and groups of algorithms for this project:

- cross-val-score: This algorithm evaluates the effectiveness of the supervised learning algorithm using test data about the scores of clubs (validation data) during model training.
- train-test-split: Train-test split is a technique to evaluate the performance of a machine learning algorithm such as regression. It will split the dataset into two parts, one for machine learning training, one for testing.
- LinearRegression: Machine learning algorithm.
- mean-squared-error: The mean squared error is a common way to measure the prediction accuracy of a model.
- r2-score: Regression score function

Finally, in this report we will use Kandall's Tau method for prediction. Kendall's Tau is a method of showing a correlation between two quantities.

Next, we will need to write code for reading the input data. Note, the input data are statistics contained in two excell tables. First sheet includes info about the teams for each season and second sheet includes corresponding points for each team.The Clubs per Season is set as the index to join the dataframes together. Sorting makes them in the same order to avoid a possible .

```
18 points_df = pd.read_excel('Data\\TurkeySuperLeague.xlsx', sheet_name='Points')
19 players_df = pd.read_excel('Data\\TurkeySuperLeague.xlsx', sheet_name='Player')
20
21 # Adding ranking column to the points table
22 points_df['Ranking'] = points_df.groupby('Season')['Points'].rank(ascending=False,method='first')
23
24 points_group_sc = points_df.set_index(['Season','Club']).sort_index()
25 players_group_sc = players_df.set_index(['Season','Club']).sort_index()
```

Now, let's join them together.Because we need to arrange both dataset according to Season and the columns, there will be no mistakes between datapoints.

```
# Joining the dataframes together as a new dataframe
whole_df = pd.DataFrame(players_group_sc)
whole_df['Points'] = points_group_sc['Points'].copy()
```

As mentioned above, we already have data like age, market value, foreign,... But we will need to extract the above data to get new features:

- Mean of age (Average of age) (*Age*)
- Standart deviation of age (Gap between oldest and youngest player) (*age-std*)
- Total foreign player count for each team in each season (*foreign-sum*)
- Total multinational player count for each team in each season (*multi-sum*)
- Total player count for each team in each season (*player-sum*)
- Foreign player count/ Total player count (*foreign-ratio*)
- Total market value for each team for each season (*market-sum*)
- Standart deviation of market value for each team in each season (*market-std*)
- Mean of market value for each team in each season (*Market Value*)

```
whole_df['Ranking'] = points_group_sc['Ranking'].copy()
data_means = players_df.groupby(['Season', 'Club']).mean()[['Market Value', 'Age','Foreign','Multinational']]
data_sums = players_df.groupby(['Season', 'Club']).sum()[['Market Value','Foreign','Multinational']]
total_player= players_df.groupby(['Season', 'Club']).count()['Player']
data_standart_deviations= players_df.groupby(['Season', 'Club']).std()[['Market Value', 'Age']]
feature_df = pd.DataFrame(data_means)
player_counts_per_season = players_df.groupby(['Season', 'Club']).count()[['Player']]
feature_df['Value Ranking'] = data_means.groupby('Season')['Market Value'].rank(ascending=False,method='first')
```

Next, let's combine all the new features into one dataframe:

```
main_df = pd.DataFrame(data_means)
#main_df['Value Ranking'] = data_means.groupby('Season')['Market Value'].rank(ascending=False,method='first')

main_df['Points'] = points_group_sc['Points'].copy()
main_df['Ranking'] = points_group_sc['Ranking'].copy()
main_df['age_std'] = data_standart_deviations['Age'].copy()
main_df['multi_sum'] = data_sums['Multinational'].copy()
main_df['foreign_sum'] = data_sums['Foreign'].copy()
main_df['player_sum'] = total_player
main_df['foreign_ratio'] = (main_df['foreign_sum'] / main_df['player_sum']).copy()
main_df['market_sum'] = data_sums['Market Value'].copy()
main_df['market_std'] = data_standart_deviations['Market Value'].copy()

normalized_means = data_means.groupby(['Season']).transform(lambda x: x/x.mean())
main_df['market_norm'] = normalized_means['Market Value']
main_df['age_norm'] = normalized_means['Age']
```

After having the above data collection steps, we will visualize the output of the data through function graphs.

```
features=main_df.columns

for i in features:
    sns.lmplot(x=i, y="Points", data=main_df,line_kws={'color': 'red'},size=4)
    text="Relation between Points and " + i
    plt.title(text)
    plt.show()
```

Here, we fixed the axis $y =$ Points, while the axis x will be put in a loop of the features we get from reading the data above. Then, the program will display the results on the function graph by one point. When all the points are displayed, we will have the line go through as many points as possible to get the common point of the features.

3.2 Predicting

Once they have built algorithms for data analysis and visualization, they move on to the process of building a program for prediction. The way to do this is to apply the train-test-split method of the Sklearn library and it will divide the feature set into test and training.

It should be noted that our prediction compares the correlation between the previous season's score and the following season's score by applying the Kendall's Tau calculation method.

We will take the results of the 2015 season as a test target. While all the remaining seasons will be used for testing and prediction. Once the predicted results are obtained, we will compare the similarity of the predicted results with the actual results for 2015 using Kendall's Tau method.

Firstly, we will build the code for predict point.

```
def convert_points_to_predictions(predictions, shouldAscend):
    predictions['Real Rank'] = predictions['Real'].rank(ascending=shouldAscend,method='first')
    predictions['Predicted Rank'] = predictions['Predict'].rank(ascending=shouldAscend,method='first')
    return predictions
def train_test_on_points(final_df):
    tau_ = 0
    for season in range(2007,2015):
        train = final_df[final_df['Season']!=season] # train data is not contain target season
        test = final_df[final_df['Season']==season] #test data is all about target season.

        X_train = train.drop(['Points','Season'],axis=1)
        y_train = train['Points'].transform(lambda x: (x - x.mean()) / x.std())
        X_test = test.drop(['Points','Season'],axis=1)
        y_test = test['Points'].transform(lambda x: (x - x.mean()) / x.std())


        final_model = LinearRegression(fit_intercept=False)
        final_model.fit(X_train,y_train)
        score = cross_val_score(final_model,X_train, y_train,cv=10)
        print(score)

        y_predict = final_model.predict(X_test)

        preds = pd.DataFrame({"Predict":y_predict})

        preds['Real']= y_test.reset_index().iloc[:,-1]

        ranks = pd.DataFrame()

        ranks['Real Rank'] = preds['Real'].rank(ascending=False,method='first')
        ranks['Predicted Rank'] = preds['Predict'].rank(ascending=False,method='first')
        tau, _ = kendalltau(ranks['Predicted Rank'], ranks['Real Rank'])
        tau_ += tau

    print('kendalltau for Points estimation: ',tau_)
```

As above, we have built an algorithm for predicting scores. Now, we will also apply the same way but with rankings. As for the rankings, we will also use the 2015 season rankings as the evaluation goal. While the ranking of the previous colors will do the test data to give the prediction results. We then compare the similarity of predictions with the actual result for the 2015 season using the Kendall's Tau method.

```
season = 2015
# to predict the ranking of 2015
final_df=main_df.copy()
final_df.dropna(inplace=True)
final_df = final_df.reset_index()
get_dummy_for_clup_names=pd.get_dummies(final_df['Club'],drop_first=True)
final_df=final_df.join(get_dummy_for_clup_names)
final_df.drop(columns=['Club'],inplace=True)

train = final_df[final_df['Season']!=season]
test = final_df[final_df['Season']==season]
X_train = train.drop(['Ranking','Season'],axis=1)
y_train = train['Ranking']
X_test = test.drop(['Ranking','Season'],axis=1)
y_test = test['Ranking']
# This is the regression model you will use
final_model = LinearRegression(fit_intercept=False)
final_model.fit(X_train,y_train)
score = cross_val_score(final_model,X_train, y_train,cv=10)
# print(score)
y_predict = final_model.predict(X_test)
#a = mean_squared_error(y_test,y_predict)
preds_rank = pd.DataFrame({"Predict":y_predict})
preds_rank['Real']= y_test.reset_index().iloc[:,-1]
# ranks.head()
# show ranks
print(preds_rank.sort_values(by='Real',ascending=True))
tau, _ = kendalltau(preds_rank['Predict'], preds_rank['Real'])
# Print tau both to file and screen
print('\n')
print('kendalltau for rank estimation:',tau)
```

4. Results and Discussion

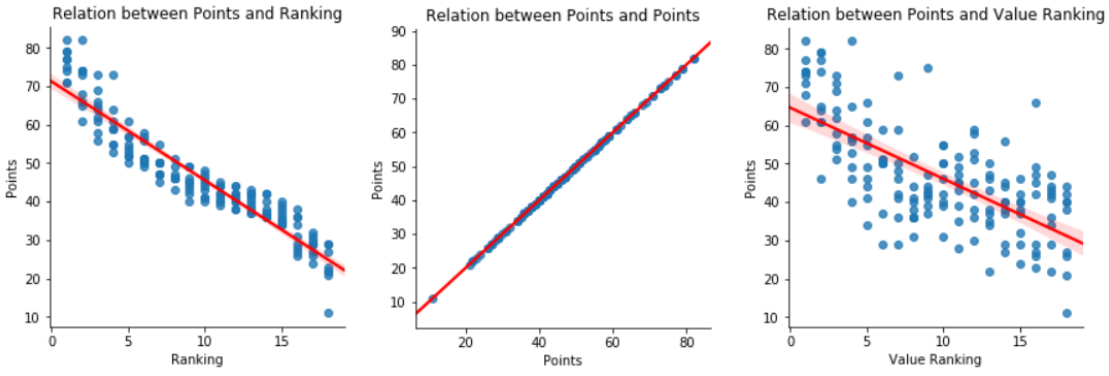
4.1 Results

The results session will be visualized to generalize the relationship between features. We will use two python libraries, Seaborn and Matplotlib.pyplot, to perform this process. In it, Seaborn will play the role of data visualization while Matplotlib.pyplot will plot the

function graph to help support linear regression. First, we'll also use Kendall's Tau method to see how similar features are to scores:

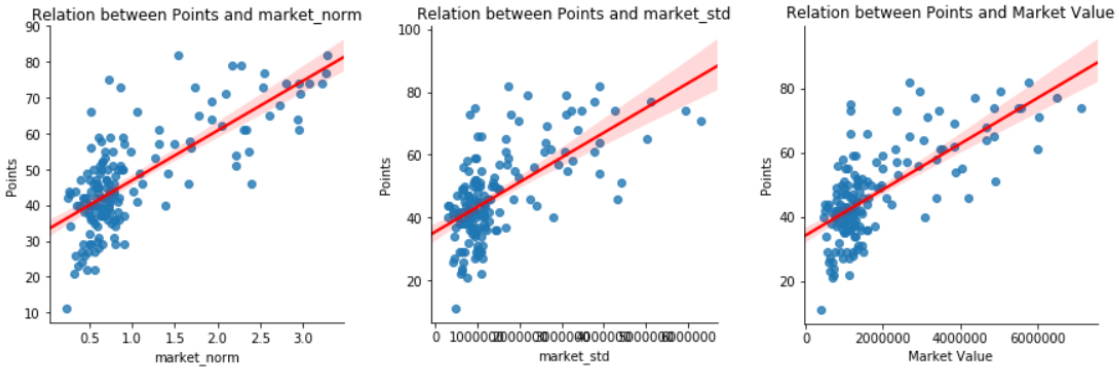
	Correlation
Points	1.000000
market_norm	0.733703
Market Value	0.700236
market_std	0.633554
market_sum	0.623709
age_norm	0.201002
Age	0.187137
Foreign	0.165547
foreign_ratio	0.165547
Multinational	0.162402
age_std	0.001843
multi_sum	-0.023719
foreign_sum	-0.175623
player_sum	-0.570539
Value Ranking	-0.683006
Ranking	-0.940617

Now, to have a better view of the relationship between features that affect the score, let's take a look at the visualization of each feature. First we will see the graph of the three relations with the highest correlation:

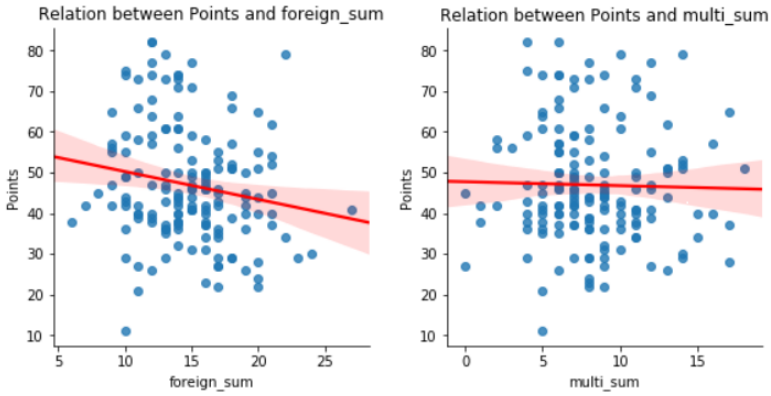


As can be seen on the three graphs, Point and Ranking, Point and Point are the two relationships as predicted with the highest compatibility. Followed by Point and Value Ranking.

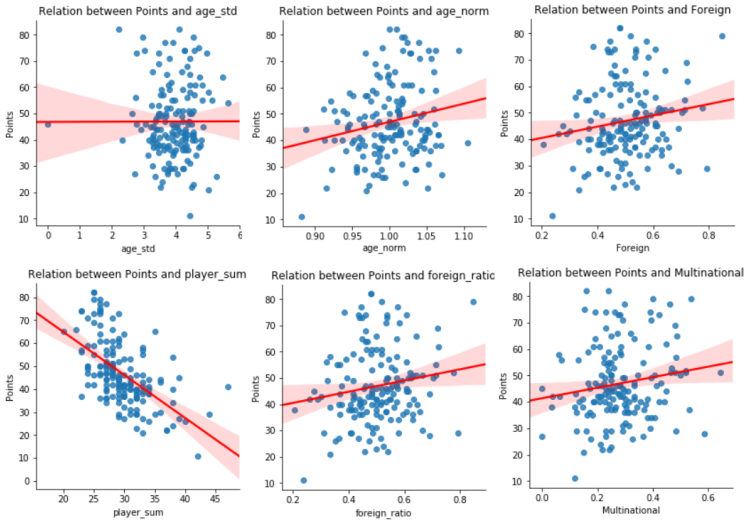
Following are the relationships that have the same compatibility:



Next, we have two features that have the same correlation:



And finally the visualization of features with low correlation with Points:



After calculating as well as applying methods such as Kendall's Tau, cross validation and train test split, we have got ourselves the prediction results for the similarity of the five target scores compared to the previous solutions. Here are the results of predicting scores and rankings:

```
kendalltau for Points estimation:  0.8022875816993467
```

```
kendalltau for rank estimation: 0.8431372549019609
```

4.2 Discussion

In this discussion, we will briefly review the advantages and disadvantages of the Linear regression algorithm and evaluate the efficiency of the algorithm during its application to this project.

First, we will talk about the strengths that this algorithm offers:

- Linear regression is easy to understand and easy to apply.
- Linear regression is easy to visualize.

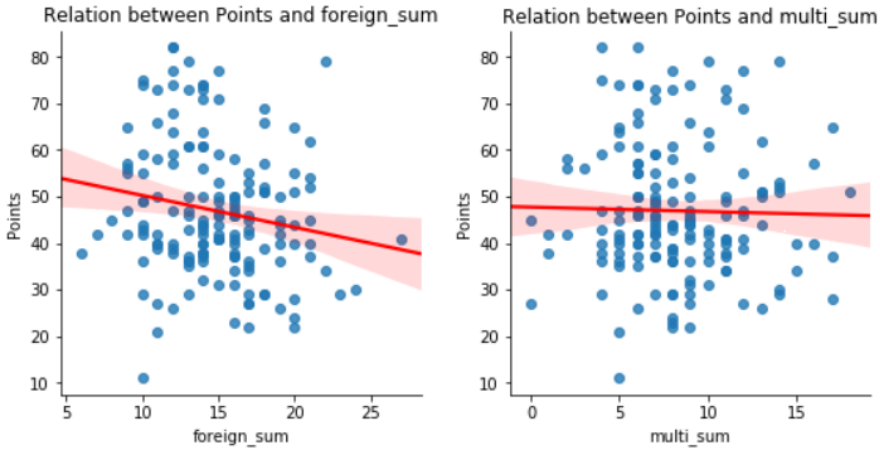
Second, let's talk about the weakness of this algorithm in processing of this project:

- For variables with large units, they need to be normalized before analysis.

  As you can see in the data processing, there are numbers with very large units that are out of range, they need to normalize before proceeding with the calculation. For example,

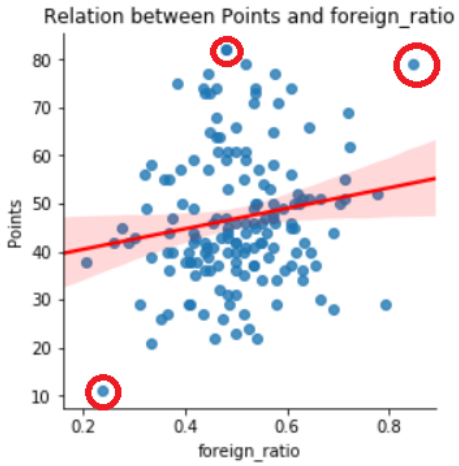
- Prone to underfitting.

  In many cases, linear regression algorithms cannot fit the data. This will reduce the effectiveness of the program. For example:



- Sensitive with the outliners.

  In practice, there will be many exceptions with data that are different from the rest. Linear regression will not be able to fit these outliers, and it will reduce the efficiency of the system. As you can see the graph below:



Eventually, each algorithm will have its particular strengths and weaknesses. Our job is to understand the requirements that the project needs to be able to choose the exact algorithm we need. Linear regression is very easy to understand and apply but it is also very susceptible to inefficiencies, however in this project we have tried to make it as efficient as possible and live up to expectations. has very high accuracy.

5. Conclusion

Linear regression is a basic and statistical method for machine learning training. When choosing linear regression as the algorithm for your model, you need to consider the input data carefully. Because besides the advantages of linear regression, it still has the inherent disadvantage of having to normalize data when the data is too large units. Another downside is that your model is very susceptible to exceptions and when using linear regression users also need to have a lot of libraries with them if they want the project to work properly. This project as an example for the application of machine learning in general as well as linear regression in particular in practice, there will be many algorithms in the field of machine learning that can be applied in different aspects. Finally, machine learning in particular as well as computer science in particular is also increasingly encroaching on life, let's take advantage and fully apply the benefits that the algorithms that we are building to develop and create a stronger science.

References

Serkan Emre Elci. Feature engineering and linear regression. *Feature Engineering and Linear Regression*, 2018.

Stephanie Glen. Kendall's tau (kendall rank correlation coefficient). *Kendall's Tau (Kendall Rank Correlation Coefficient)*, 2022.

Mohit Gupta. Linear regression. *Linear Regression*, IT-14(3):462–467, 2022.

Christopher M.Bishop. *Pattern Recognition and Machine Learning*. University of California, San Mateo, CA, 2006.

Peter Mqoaie. Machine learning can help us predict the fifa world cup 2022. *Machine learning can help us predict the FIFA World Cup 2022*, 2022.