Predicting Market Value of a Player

Data Mining CA-2

Abstract

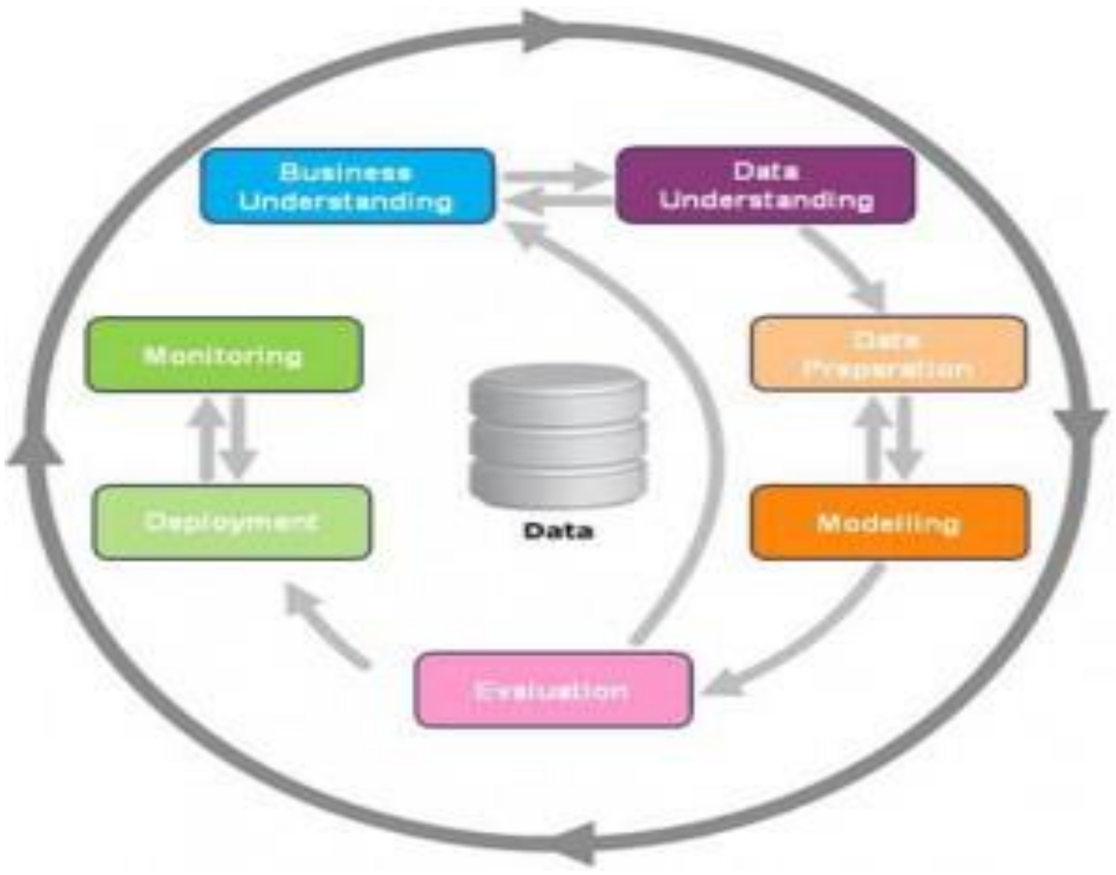
As we have seen in recent times the value of football players goes to such extent that it seems that there is no upper limit to this transaction. However, there are a lot of teams that cannot afford to spend on such a level. There are many teams that hire sports analytics professional to get a gauge of any new player that enters the fray. So, that they can make proper assumptions about the value and can make smart buys rather than the most expensive one that they usually ends up getting signed. Therefore, for this assignment, we have assumed that we work for a sports analytics company and our company has been hired by a club to make a model to predict the market value of any new player that comes through. Hence, in this assignment, we have made a model to predict the value of players and we have used the FIFA 19 complete players dataset for that.

1. Introduction

There are millions of professional players in hundreds of countries and gauging their value can be a pain if you do not have the proper data. Hence, we have used the FIFA 19 complete player dataset as it is the most comprehensive and well-researched dataset found in the industry. In order to predict that, we have implemented a ML model and applied the principles to process the data for it.

Specifically, we have taken this Dataset available on Kaggle:
https://www.kaggle.com/karangadiya/fifa19

This process was implemented using CRISP-DM (CRoss-Industry Standard Process for Data Mining) method. This analysis follows the phases of CRISP-DM reference model



2. Business Understanding

In this section we understand the Business problems associated with the task and to decide if they can be extinguished using ML techniques.

2.1 Business Objective

- To determine the market value of a player.
- Which are the main factors that affect the players market value?

2.2 Stakeholders

The club that has hired our company are the stakeholder. For this assignment, we have assumed that this club is a mid-level club that is looking for ‘smart buys’ rather than ‘hard buys’.

2.4 Business Constraints

Although, our model is highly accurate, we have to realise that there are a lot of other factors that can affect the market value of the player that are not necessarily sports related like the agent commission, the tax rate of the particularly country etc.

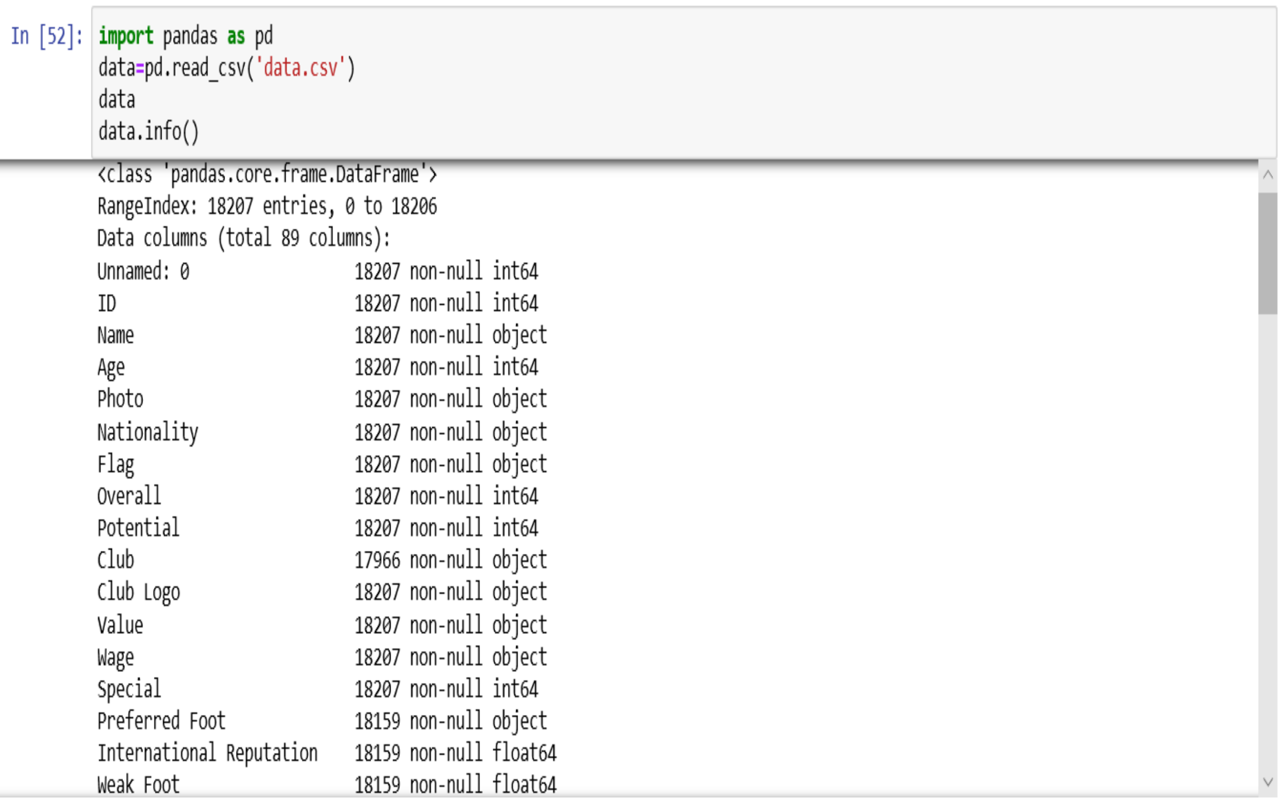
3. Data Understanding

3.1 Data Source

we have taken this Dataset available on Kaggle: https://www.kaggle.com/karangadiya/fifa19

3.2 Data Exploration

We have used Python to get details about this data.

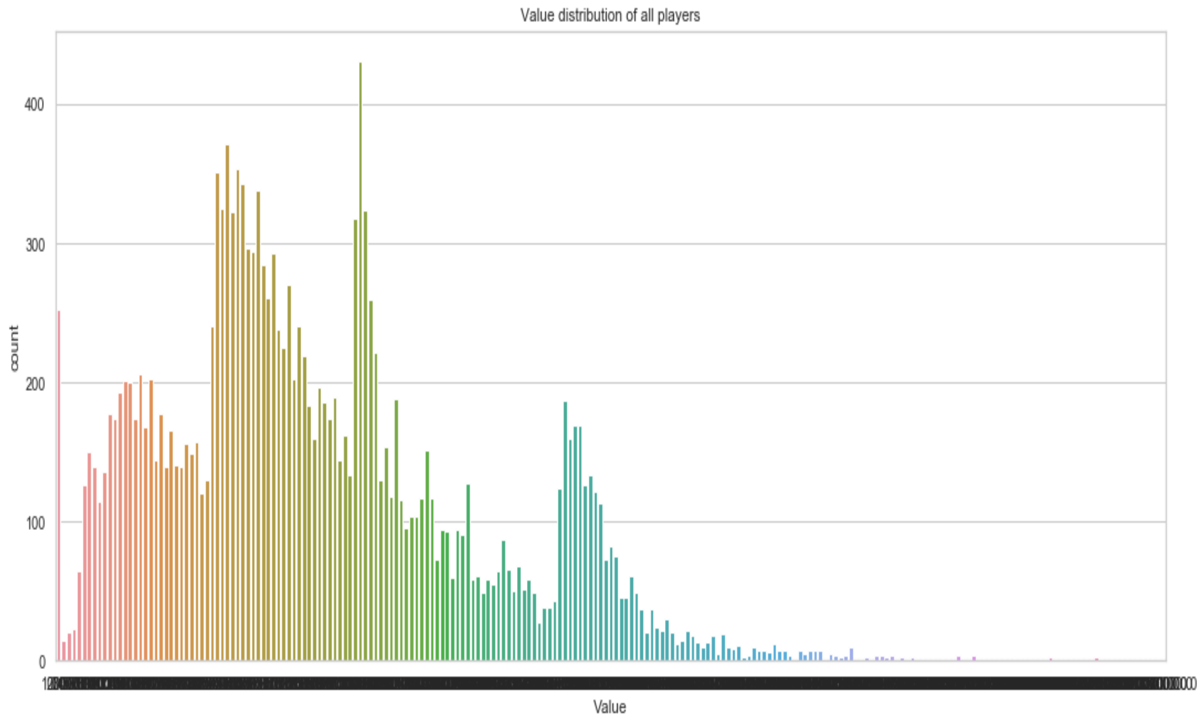


```
In [52]: import pandas as pd
         data=pd.read_csv('data.csv')
         data
         data.info()
```

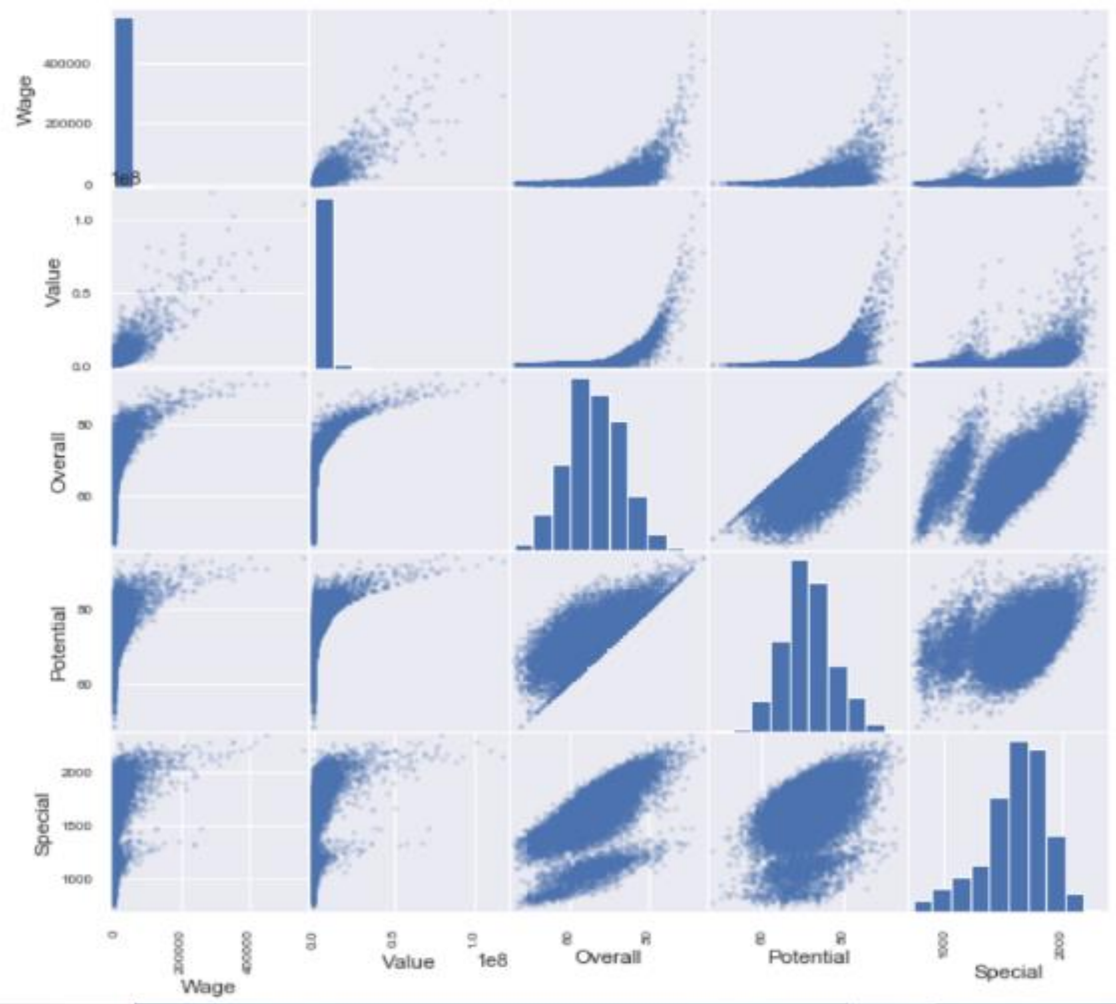
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18207 entries, 0 to 18206
Data columns (total 89 columns):
Unnamed: 0                  18207 non-null int64
ID                          18207 non-null int64
Name                        18207 non-null object
Age                         18207 non-null int64
Photo                       18207 non-null object
Nationality                 18207 non-null object
Flag                        18207 non-null object
Overall                     18207 non-null int64
Potential                   18207 non-null int64
Club                        17966 non-null object
Club Logo                   18207 non-null object
Value                       18207 non-null object
Wage                        18207 non-null object
Special                     18207 non-null int64
Preferred Foot              18159 non-null object
International Reputation    18159 non-null float64
Weak Foot                   18159 non-null float64
```

As per results, we can see that Data has 18000+ observations and 89 feature which describes the dataset, which are of various data types and has to be worked on further.

To get the range of market Values that we have to go through we also plotted a graph to get that



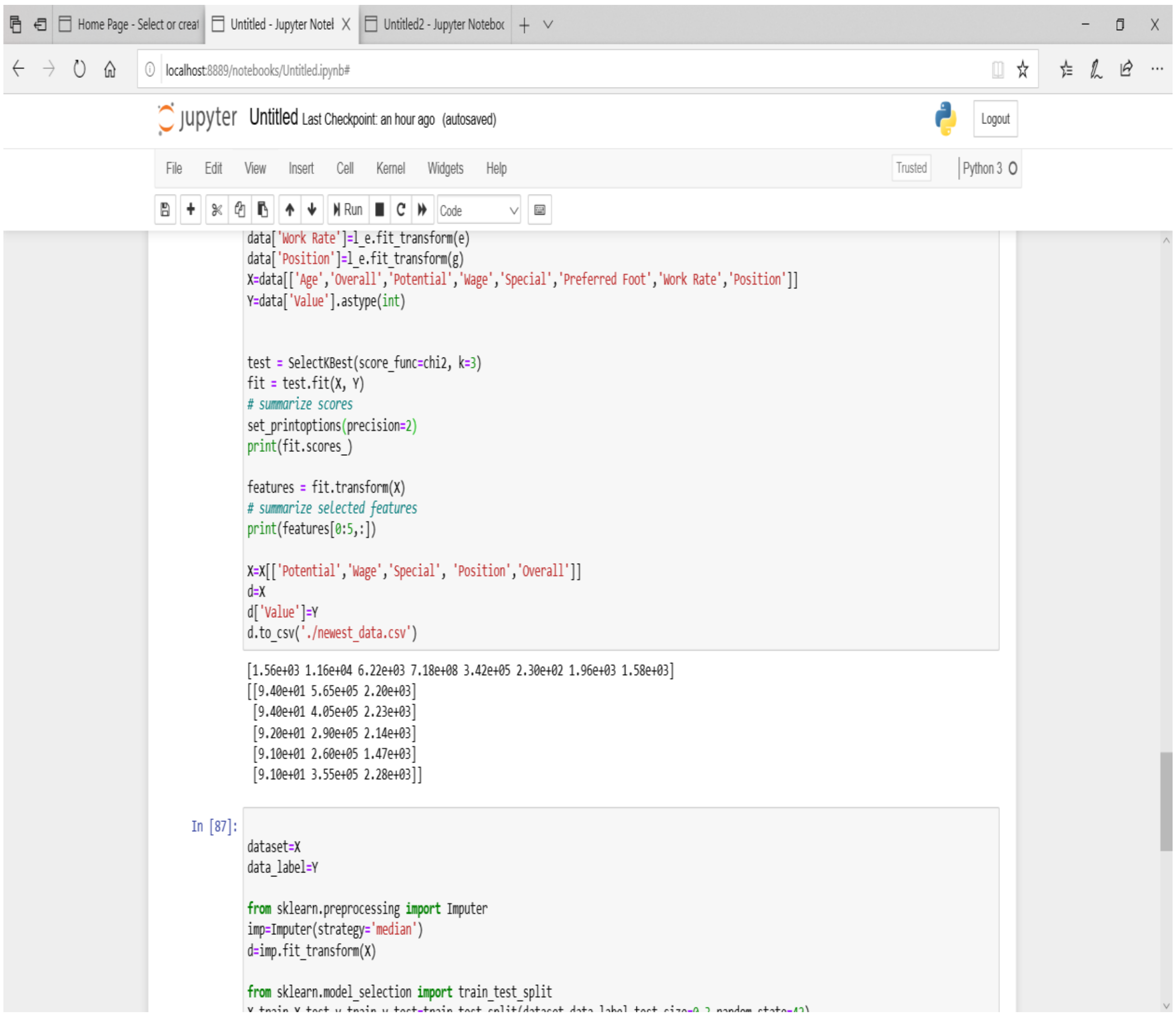
Moreover, to the get the correlation among the feature, we used scatter matrix to help us select the best features of this dataset to get the market value of the Player.



3.3 Data Cleaning

To clean the dataset, we used the label binarizer and label encoder to get the categorical values in the dataset to numericals. We also ignored the features that deal with the players abilities on the field as we know from our football knowledge that the Overall feature represents those features to as we calculate Overall by combining those attributes of those players and was bound to have a lot of missing values.

Then from the remaining 8 features, we used SelectKBest for feature selection on the advice of our professor and found that we ignored an important feature when we just ran the model for features we thought were the most important and that feature was the position at which the player plays. This made our accuracy score higher and were able to get the proper model.


Home Page - Select or creat
Untitled - Jupyter Notel
Untitled2 - Jupyter Noteboc
localhost:8889/notebooks/Untitled.ipynb#
jupyter Untitled Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
data['Work Rate']=l_e.fit_transform(e)
data['Position']=l_e.fit_transform(g)
X=data[['Age','Overall','Potential','Wage','Special','Preferred Foot','Work Rate','Position']]
Y=data['Value'].astype(int)


test = SelectKBest(score_func=chi2, k=3)
fit = test.fit(X, Y)
# summarize scores
set_printoptions(precision=2)
print(fit.scores_)

features = fit.transform(X)
# summarize selected features
print(features[0:5,:])

X=X[['Potential','Wage','Special', 'Position','Overall']]
d=X
d['Value']=Y
d.to_csv('./newest_data.csv')
```

```
[1.56e+03 1.16e+04 6.22e+03 7.18e+08 3.42e+05 2.30e+02 1.96e+03 1.58e+03]
[[9.40e+01 5.65e+05 2.20e+03]
 [9.40e+01 4.05e+05 2.23e+03]
 [9.20e+01 2.90e+05 2.14e+03]
 [9.10e+01 2.60e+05 1.47e+03]
 [9.10e+01 3.55e+05 2.28e+03]]
```

```
In [87]:
dataset=X
data_label=Y

from sklearn.preprocessing import Imputer
imp=Imputer(strategy='median')
d=imp.fit_transform(X)

from sklearn.model_selection import train_test_split
```

After that we divide the dataset into train and test data and then move on to making the model.

4. Methodology/Approach Selection

As mentioned before we used CRISP DM process to analyse the data, and use machine learning algorithms to make the right model

4.1 Software/Tools Selection

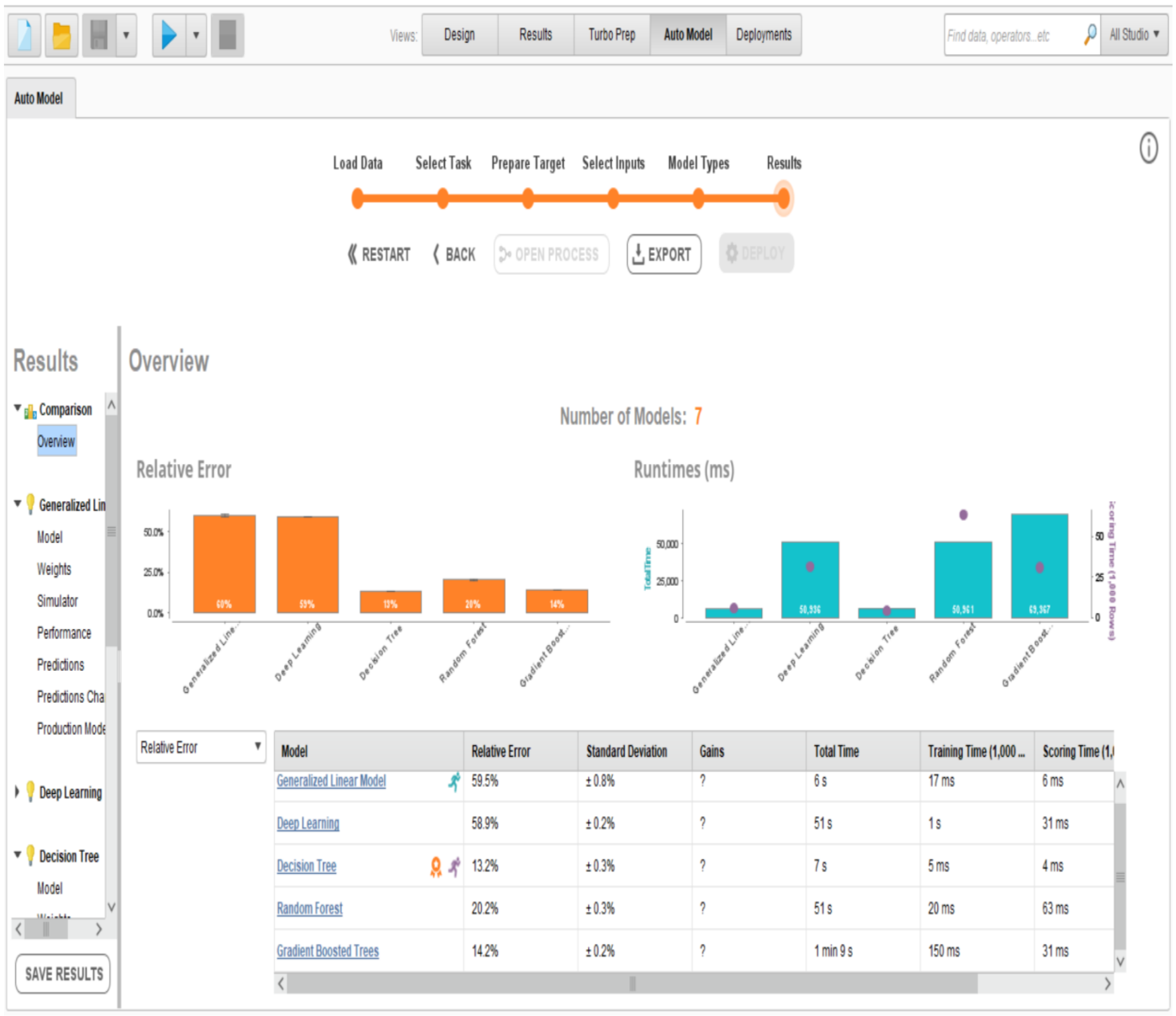
The softwares and tools used in this assignment are:

- Rapid Miner Studio
- Jupyter Python

These are the best approaches for this type of task. Here, Rapid Miner helped us a lot as we were able to find the right model by using the “Auto-Model” feature of Rapid Miner to get the best Algo for the dataset.

4.2 Testing & Selection of Model/Algorithm Approach

For this task, we used the “Auto-Model” feature of Rapid Miner the right model for it.

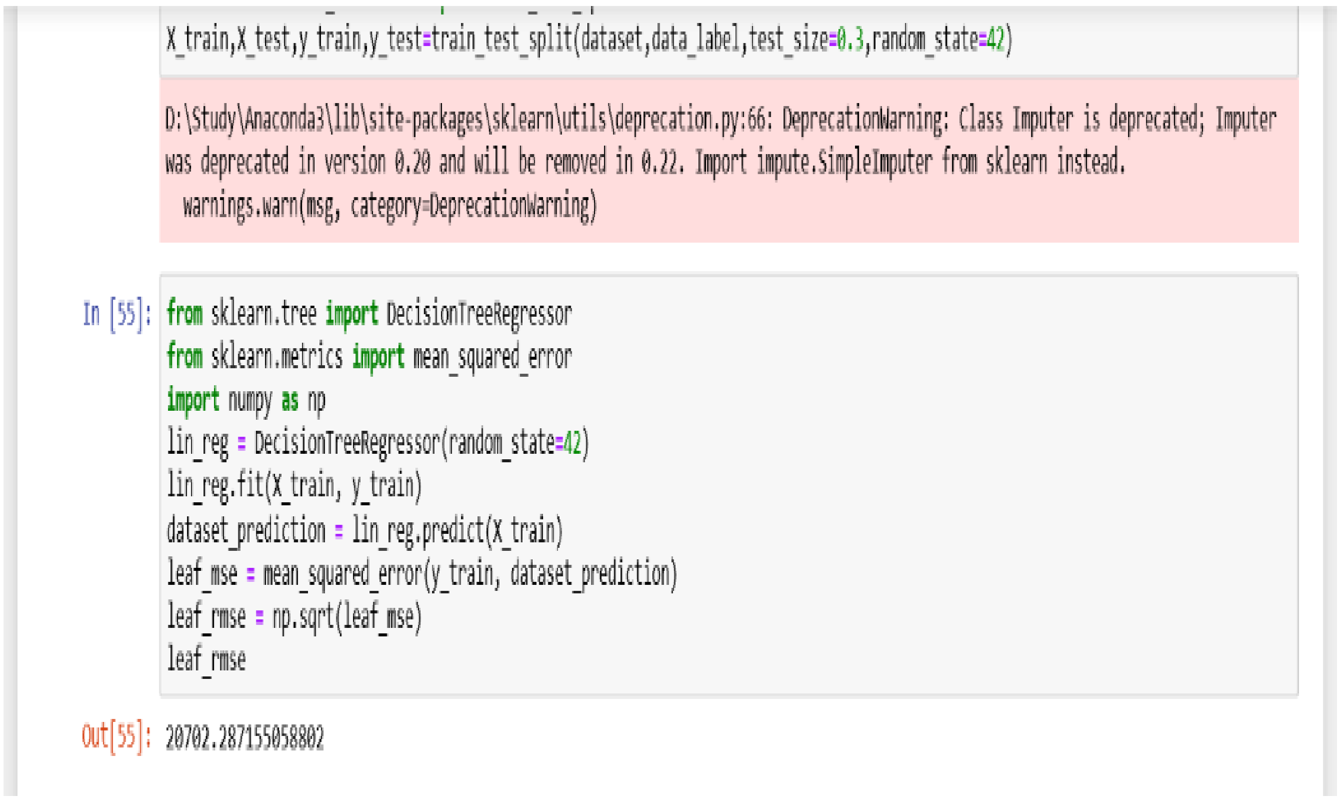


Model	Relative Error	Standard Deviation	Gains	Total Time	Training Time (1,000 ...	Scoring Time (1,
Generalized Linear Model	59.5%	± 0.8%	?	6 s	17 ms	6 ms
Deep Learning	58.9%	± 0.2%	?	51 s	1 s	31 ms
Decision Tree	13.2%	± 0.3%	?	7 s	5 ms	4 ms
Random Forest	20.2%	± 0.3%	?	51 s	20 ms	63 ms
Gradient Boosted Trees	14.2%	± 0.2%	?	1 min 9 s	150 ms	31 ms

As we can see, the Decision Tree is the best model that works on this dataset as it gives us the lowest relative error. Hence we decided to use Python to implement it.

5. Model Building & Evaluation

To construct the model, we used python to implement Decision Tree Regressor as our Target variable is a continuous value.



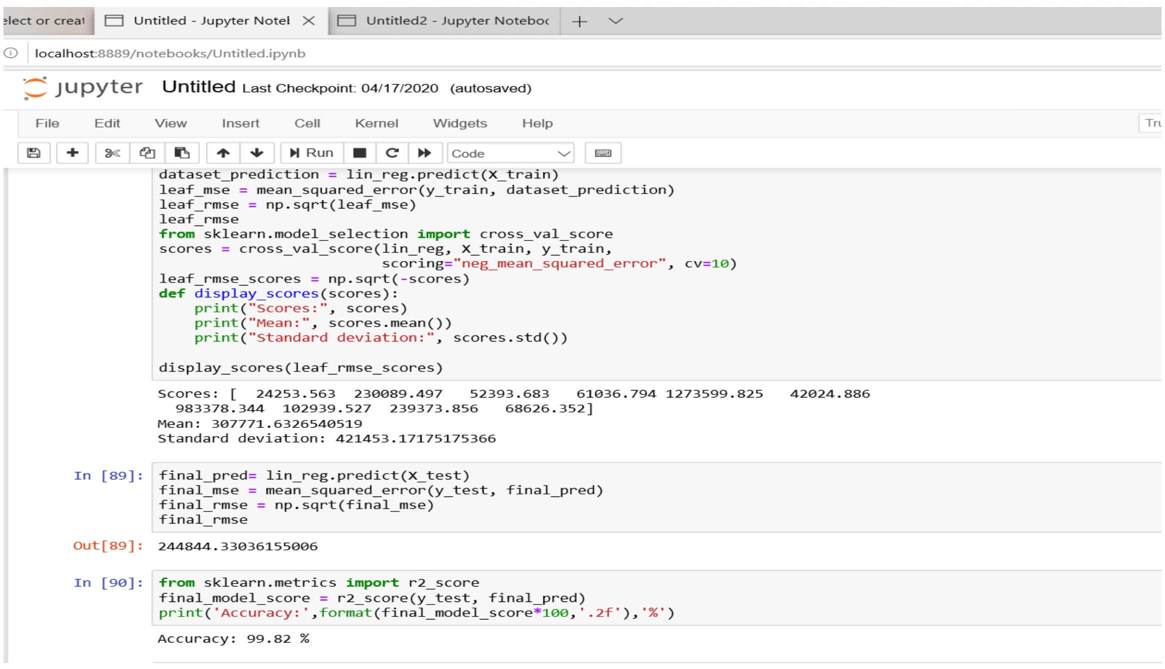
```
X_train,X_test,y_train,y_test=train_test_split(dataset,data_label,test_size=0.3,random_state=42)
```

```
D:\Study\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:66: DeprecationWarning: Class Imputer is deprecated; Imputer
was deprecated in version 0.20 and will be removed in 0.22. Import impute.SimpleImputer from sklearn instead.
  warnings.warn(msg, category=DeprecationWarning)
```

```
In [55]: from sklearn.tree import DecisionTreeRegressor
         from sklearn.metrics import mean_squared_error
         import numpy as np
         lin_reg = DecisionTreeRegressor(random_state=42)
         lin_reg.fit(X_train, y_train)
         dataset_prediction = lin_reg.predict(X_train)
         leaf_mse = mean_squared_error(y_train, dataset_prediction)
         leaf_rmse = np.sqrt(leaf_mse)
         leaf_rmse

Out[55]: 20702.287155058802
```

6.2 Evaluation

For evaluation of this model, we used the RMSE score, MSE error, accuracy score and the r2 score to get the proper evaluation of this model using cross_val_scores. This helped us to cross validate our knowledge within the train set.

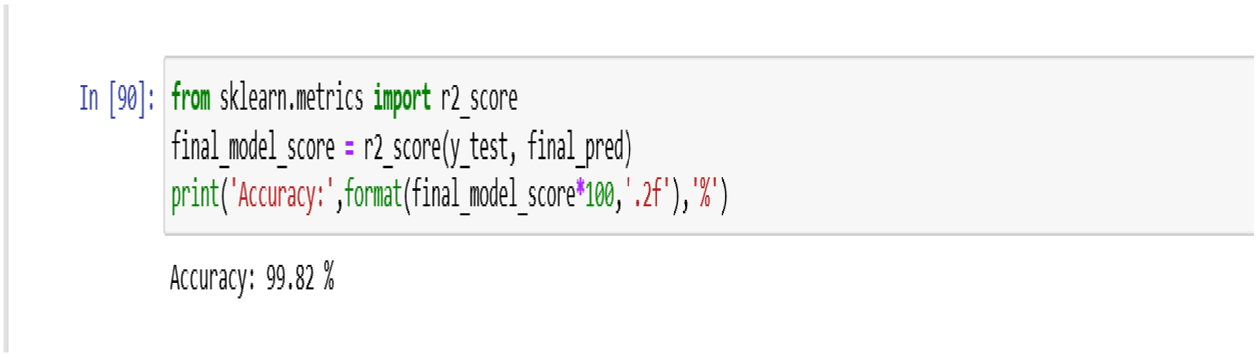


elect or creat
Untitled - Jupyter Notel
Untitled2 - Jupyter Noteboo
localhost:8889/notebooks/Untitled.ipynb
jupyter
Untitled Last Checkpoint: 04/17/2020 (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

```
dataset_prediction = lin_reg.predict(X_train)
leaf_mse = mean_squared_error(y_train, dataset_prediction)
leaf_rmse = np.sqrt(leaf_mse)
leaf_rmse
from sklearn.model_selection import cross_val_score
scores = cross_val_score(lin_reg, X_train, y_train,
                         scoring="neg_mean_squared_error", cv=10)
leaf_rmse_scores = np.sqrt(-scores)
def display_scores(scores):
    print("Scores:", scores)
    print("Mean:", scores.mean())
    print("Standard deviation:", scores.std())

display_scores(leaf_rmse_scores)

Scores: [  24253.563  230089.497   52393.683   61036.794 1273599.825   42024.886
   983378.344  102939.527  239373.856   68626.352]
Mean: 307771.6326540519
Standard deviation: 421453.17175175366

In [89]: final_pred= lin_reg.predict(X_test)
         final_mse = mean_squared_error(y_test, final_pred)
         final_rmse = np.sqrt(final_mse)
         final_rmse

Out[89]: 244844.33036155006

In [90]: from sklearn.metrics import r2_score
         final_model_score = r2_score(y_test, final_pred)
         print('Accuracy:',format(final_model_score*100,'.2f'),'%')

         Accuracy: 99.82 %
```

6. Deployment of Model

The model was then deployed on the test set to get how it eventually performs in a real world scenario and our predictions had an accuracy of 99.82.



```
In [90]: from sklearn.metrics import r2_score
         final_model_score = r2_score(y_test, final_pred)
         print('Accuracy:',format(final_model_score*100,'.2f'),'%')

         Accuracy: 99.82 %
```

7. Conclusion

We were properly able to work on this data by following the steps of CRISP-DM

and in the end we can conclude this assignment by saying that:

- the features that most affect the value of a player are Wage, Overall, Position, Special and Potential.
- Position of a player plays a major role in his/her value.
- Therefore, as all the others are sort of environmental factors, to get the proper gauge of value of a new player, we would have to calculate the “Special” attribute of that player, while the other features are generally available already.