Name : Golapkar Atharva Suryakant Roll No : ITA627

Report on Python Mini-project

Details about the dataset:

The dataset I am working on contains all the FIFA players' statistics. It has their names, overall rating, national rating, jersey number, etc. It contains 17954 rows and 92 columns. From all the data available in the dataset, we can predict the following things:

- Overall Rating, National Rating, Club Rating
- Their value in Euros
- Whether there is an increase or decrease in their performance with respect to the age or other factors
- Co-relate between Age and Overall rating
- Co-relate between Age and Nationality
- Co-relate between Age and Potential

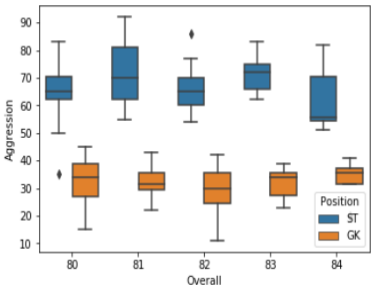
And many more

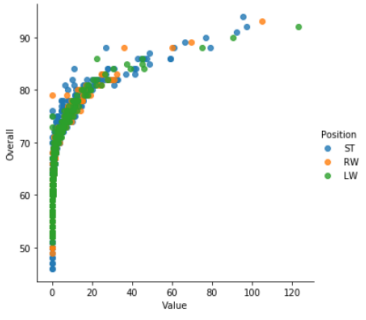
The target field I am working on is the player's overall rating.

Things performed on dataset	Things performed by me
Various fields with null values were present in the dataset	Searched for blank fields and on the basis of the target field dropped those fields with null values

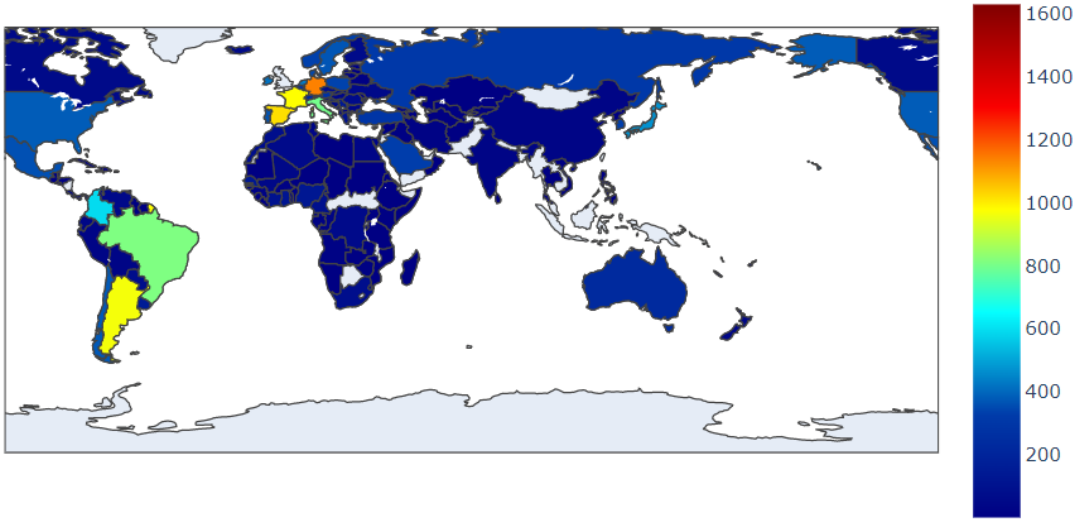
Earlier Work done on Dataset:

Multivariate Scatter plots
and Boxplots are been plotted
using the above dataset





Data visualization on various columns like Nationality of players here is represented as a heat map on a map



Finding the best team

0 years later
Average rating: 86.4

Position	Player	Overall_n_yr_later
GK	De Gea	90.0
LB	Jordi Alba	85.0
CB	Sergio Ramos	90.0
RB	Carvajal	84.0
LM	David Silva	87.0
LW	Marco Asensio	84.0
CM	Sergio Busquets	86.0
CDM	Thiago	88.0
CDM	Javi Martínez	86.0
ST	Diego Costa	86.0
ST	Morata	84.0

1 years later
Average rating: 87.0

Position	Player	Overall_n_yr_later
GK	De Gea	90.4
LB	Jordi Alba	85.8
CB	Sergio Ramos	89.6
RB	Carvajal	84.8
LM	Isco	86.8
LW	Marco Asensio	85.8
CM	Sergio Busquets	86.8
CDM	Thiago	88.4
CDM	Javi Martínez	86.8
ST	Diego Costa	86.8

Work done by me:

In this project, a model is made after applying different regression techniques and training the model. This model takes in the input of the following fields:
PLAYER POTENTIAL RATING:
PLAYER CURVE RATING:
PLAYER REACTION RATING:
PLAYER VISION RATING:
PLAYER COMPOSURE RATING:
And then it predicts what will be the player's overall rating
For prediction I have used 3 different regression techniques and each one of them gives me different accuracy score

- Linear Regression
- Decision Tree Regression
- Random Forest Regression

STEPS DONE BY ME FOR DATA ANALYSIS:

Step 1: Data PreProcessing

Here I scanned for all the null values or outliers using various techniques.
For outlier detection, I used the box-plot technique
Then after executing the df.info() command I was able to get the gist of the dataset from where i was able to see how many columns have null.

Step 2: Data Visualization

The motive behind doing data visualization before data transformation was to work on only that data that is correlated to each other. After finding Karl Pearson's coefficient of correlation test for the entire dataset and plotting it on a heatmap, I was able to decide which columns should be considered to predict the overall rating of a player. I also plotted a histogram for the overall rating field so that I can have a rough idea of in generally where the predicted points will lie.

Step 3: Data Transformation

After gaining knowledge about which columns are required from step 2, I decided to drop all the columns using the df.drop('Column_Name ', axis = 1) command will I got the satisfying dataset.

Step 4: Data Splitting

Here, using the train test split function I split the data set into 2 parts one for training and for testing purposes.

Step 5: Model Training

In this step, using various regressor techniques, I created a model which can be used to predict the overall rating after getting to know about the vision, curve, composure, potential, and reaction rating of the player.
I have used Linear regressor, Random forest Regressor and Decision tree Regressor to create a model.

Step 6: Creating a Simple System

In this step, I have created a simple system where the user is asked to fill in the 5 fields of player rating and then the user will be provided with the overall rating of the player.
Since I have used 3 different techniques to create a regression model , the accuracy of the prediction will also be different so then and there itself, the user is also able to know which regressor technique is having more accuracy

```
print ("Enter the potential , vision , curve , reaction and composure rating of a player to get the overall rating")
potential = input('Potential : ')
curve = input('curve: ')
reactions = input('reactions: ')
vision = input("vision : ")
composure = input('composure : ')
new_player = np.array([[potential,curve,reactions,vision,composure]])
linear = lrm.predict(new_player)
decision = DTree.predict(new_player)
forest = RForest.predict(new_player)
print ('Rating of the new player will be (linear):', linear[0])
print ('Rating of the new player will be (decision):', decision[0])
print ('Rating of the new player will be (random forest):', forest[0])
```

```
Enter the potential , vision , curve , reaction and composure rating of a player to get the overall rating
Potential : 80
curve: 80
reactions: 80
vision : 80
composure : 80
Rating of the new player will be (linear): 79.34676052689869
Rating of the new player will be (decision): 80.0
Rating of the new player will be (random forest): 79.8
```

```
C:\Users\atharva\anaconda3\lib\site-packages\sklearn\utils\validation.py:63: FutureWarning: Arrays of bytes/strings is being co
nverted to decimal numbers if dtype='numeric'. This behavior is deprecated in 0.24 and will be removed in 1.1 (renaming of 0.2
6). Please convert your data to numeric values explicitly instead.
  return f(*args, **kwargs)
```

Result

3 different models predict the overall rating of a player but have different accuracy score

REGRESSOR	ACCURACY
Linear Regression	82.56%
Decision Tree	76.70%
Random Forest	84.95%

Conclusion

In order to predict the overall rating of a player just from the vision rating, potential rating, curve rating, composure rating, and reaction rating, I have used 3 techniques that are Linear Regression, Random forest, as well as Decision Tree out of which Random Forest model have the highest accuracy of 84.95%, then comes Linear Regression model having 82.56% of accuracy and finally Decision Tree model with the accuracy of 76.70%