# **Pre-processing**

First, we deal with null string features by filling null rows with mode().

As a function written in pre\_processing (Feature\_Mode()) file that calculates the number of most frequent value existing in a specific column then filling null rows by this value.

That’s the set of features we applied mode() on them (club\_team, national\_team, club\_position, national\_team\_position, club\_join\_date, contract\_end\_year).

Then, we dealt with null integer features by filling null rows with mean() as built function passed to (fillna()) function.

mean() applied on the remaining columns of the data set.

Then we applied feature selection (correlation) after apply analysis on the dataset and selecting top 50% features related to (value).

And using correlation method, we dropped low correlated features to (value).

Then scaled these features using (featureScaling) function that written in the pre\_processing file with scale(0,1).

# **analysis**

we split (traits , tags , positions) columns by separator (‘,’) and replaced null rows in them with integer value(0) and non-null rows with value (number of split arrays by the separator).

Then, we encoded string values to integer values (High:3, Medium:2, low:1) in (work\_rate) column and split it using separator (‘/’) and replaced non-null rows with value (sum of two split integers).

Then, we split (contract\_end\_year) column by separator (‘-’) and replaced non-null rows with value (year).

Then, we replaced string values in column (body\_type) by integer values (Stocky:3, Normal:2, Lean:1) and replaced inconsistent rows by default (Normal:2) value.

Then, we replaced string values in column (club\_position) by integer values according to (overall\_rating) column.

Then, we split (last 26 columns) by separator (‘+’) and replaced null rows in them with integer value(40) and non-null rows with value (sum of two split integers).

Then we dropped weak related columns to (value) that is not important (id, name, full\_name, birth\_date, nationality, height\_cm, weight\_kgs, club\_jersey\_number, club\_join\_date, national\_team, national\_rating, national\_team\_position, national\_jersey\_number, club\_team, preferred\_foot).

We saved the dataset after applying all this analysis in a new dataset to reduce running time using to\_csv() function that takes new dataset name ana attribute called index if it false the new dataset will not have column for index.

data.to\_csv('Final\_Fifa\_Data\_after\_preprocessing.csv', index=False).

And then this dataset (Final\_Fifa\_Data\_after\_preprocessing) that is used in the modelling.

In the two models we used feature which is highly correlated features to (value).

Top 50% correlated features are (overall\_rating, potential, wage, release\_clause\_euro, club\_rating, club position, international\_reputation (1-5), reactions, tags).

# **Regression**

We applied two different models are Multiple linear regression and polynomial regression with different degrees.

|  |  |
| --- | --- |
| Multiple linear regression | |
| Mean Square Error (MSE) | training time |
| MSE of test = 1530192305709.41 | 0.972681999206543 |

We applied data split of size 70% for training and 30% for testing

|  |  |  |
| --- | --- | --- |
| polynomial regression | | |
| Degree of complexity | Mean Square Error (MSE) for test | training time |
| degree 2 | 957568750377.1337 | 0.04629063606262207 |
| degree 3 | 492225612687.35 | 0.8348536491394043 |
| degree 4 | 55830280677794.75 | 0.9903497695922852 |
| degree 5 | 1.269915063246461e+19 | 7.192081689834595 |

We applied data split of size 80% for training and 20% for testing

**Conclusion.**

First, we applied some pre-processing such as dealing with the missing values, feature selection that the model learning on it and applied scaling on the dataset and feature encoder.

We applied multiple linear regression and polynomial regression.

The mean square error of multiple linear regression is 1530192305709.41.

The best trial of polynomial regression at degree 3 where mean square error equal 492225612687.35.