**Report**

# **Milestone 1:**

**Preprocessing Techniques**

* **Missing Values:** #'club\_team''club\_position','club\_join\_date','contract\_end\_year','tags', 'traits''LS','ST','RS',

'LW','LF','CF','RF','RW','LAM','CAM','RAM','LM','LCM','CM','RCM','RM','LWB','LDM','CDM','RDM','RWB','LB','LCB','CB','RCB','RB': got the mean/median of the col and refill the null values with the mean.

1. **Mean values**: 'club\_rating','club\_jersey\_number'

**Implemented:** using data[i].fillna(data[i].mean())

1. **Median values**: 'LS','ST','RS','LW','LF','CF','RF','RW','LAM','CAM','RAM','LM','LCM','CM','RCM','RM','LWB','LDM','CDM','RDM','RWB','LB','LCB','CB','RCB','RB' , 'club\_join\_date','contract\_end\_year'

**Implemented:** using data[i].replace(0,data[i].median())

* **Transformation**

1. **Feature encoder**: the process of turning categorical data in a dataset into numerical data

Col: nationality, club team, traits

**Implemented :**using function Feature\_Encoder(data, column list)

1. **One hot encoding**: the essential process of converting the categorical data variables to be provided to machine and deep learning algorithms which in turn improve predictions as well as classification accuracy of a model.

Col: 'preferred\_foot','work\_rate','body\_type','club\_position','pos1', 'pos2', 'pos3','pos4','tags'

**Implemented :**using function encode\_and\_bind(data, column list)

**Feature scaling:** method used to normalize the range of independent variables or features of data

Col: overall rating, potential, wage, international reputation , release clause euro , club rating , reactions

**Implemented :**using function featureScaling(X ,value1,value2)

* **Outliers :**

Wage->1592

release\_clause\_euro -> 1847

value->1917

Position col -> split to 4 cols

**Implemented :**using function find\_outliers\_IQR(data[i])

* **Splitting:**

Position col -> split to 4 cols (pos1,pos2,pos3,pos4)

**Implemented :**using data['positions'].str.split(',', expand=True)

* **Sum**

'LS','ST','RS','LW','LF','CF','RF','RW','LAM','CAM','RAM','LM','LCM','CM','RCM','RM','LWB','LDM','CDM','RDM','RWB','LB','LCB','CB','RCB','RB' -> Sum the value in the cell

**Implemented :**using for loop to split each cell two number and sum

* **Convert**

contract\_end\_year, club\_join\_date -> to year

**Implemented :**using for loop to extract year from date

**Analysis**

Here we analyze the coloration of the Value col with the top features(above 0.5)

* Overall rating:0.62
* Potential: 0.57
* Wage : 0.86
* international reputation:0.64
* Release clause euro: 0.97
* Club rating: 0.53
* Reactions: 0.54

**Regression Techniques**

**Multi regression**: statistical technique that can be used to analyze the relationship between a single dependent variable and several independent variables

**Polynomial**: form of Linear regression known as a special case of Multiple linear regression which estimates the relationship as an nth degree polynomial

**Some Differences**

|  |  |  |
| --- | --- | --- |
|  | Model 1 (multi) | Model 2(poly) |
| Accuracy (Test) | 0.9656857017996736 | 0.9829983298409519 |
| Error | 1284228335241.1196 | 391793094071.735 |
| Training Time | 0.008407354354858398 | 0.08975076675415039 |

**Features**

* **Features used:** overall rating, potential, wage, international reputation , release clause euro , club rating , reactions (all rows)
* **Dropped cols**: 'id', 'name','full\_name','birth\_date','national\_team',

'national\_rating','national\_team\_position','national\_jersey\_number','positions'

**Sizes**

Training size: model 1->70 , model 2-> 70

Test size: model 1->30 , model 2-> 30

**Screenshots**

**Chart

Description automatically generated with low confidence**

**Conclusion**

We handled the data by a lot of preprocessing steps (filling the missing values with their median/ mean and detecting the cols of the outliers) and then used two regression techniques to determine the value of different players with respect to his different attributes that affects his value.

# **Milestone 2:**

# **Summarize**:

|  |
| --- |
|  |

# **Feature Selection**

Feature Selection in python is the process where you automatically or manually select the features in the dataset that contribute most to your prediction variable or output in which you are interested.

As not all the features selected are helpful to the model and some may even damage its performance, so we chose the right features carefully for these reasons

1. Improve the accuracy
2. Reduce overfitting
3. It enables the machine learning algorithm to train faster.
4. Reduces the complexity and makes it easier to interpret

Featurewiz: can automatically detect if the problem is regression or classification and it’s used for automatically creating and selecting important features in your dataset that will create the best model with higher performance.

* SULOV: stands for searching for uncorrelated list of variables
* Recursive XGBoost: After selecting the features with less coloration and high mutual score, it finds the best features among the remaining ones.

Explanation: it was proved , because input numerical and output categorical but regression input numerical and output numerical.

# **Hyperparameter Tuning:**

In SVM Model and logistic regression model use three hyperparameter C, max\_iter and penalty.

We've got different accuracy and performance example in SVM:

* When 'C': 0.01, 'max\_iter': 1000, 'penalty': 'l2' 🡪accuracy model = 0.423
* When 'C': 10, 'max\_iter': 10000, 'penalty': 'l2' 🡪 accuracy model = 439
* This is mean accuracy better when ‘C’ is big but this is lead to overfitting

We've got best hyperparameter in logistic regression mode when C=1.0092528860766845, max\_iter=1000, 'penalty': 'l2'

In Decision Trees Model use three hyperparameter criterion, splitter, max\_depth

We've got different accuracy and performance example in Decision Trees:

* When criterion: 'gini', splitter: 'best', max\_depth: 50 🡪 accuracy model = 0.849
* When criterion: 'entropy', splitter: 'random', max\_depth: 70 🡪accuracy model= 0.831
* When criterion: 'gini', splitter: 'random', max\_depth: 60 🡪 accuracy model = 0.831
* When criterion: 'entropy', splitter: 'best', max\_depth: 80 🡪 accuracy model = 0.858
* When criterion: 'gini', splitter: 'best', max\_depth: 40 🡪 accuracy model = 0.853

# **Conclusion**

In this phase we use regression(the previous phase) pre Processing but don’t use feature selection because in this phase different the previous phase(input is numerical and output categorical)

Use many hyperparameter for train model and select best hyperparameter in SVM, logistic and decision tree it is proved by training many model and analysis result

In SVM model select one vs one model because this is more better from one vs all but it is more Complicated