|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Metrics | Wrapper Methods | | | | Embedded Methods | |
| Forward feature Selection | Backward Feature Elimination | Recursive feature Elimination | Boruta Feature Selection | Logistics Regression L1 | Random Forests |
| No of Features | Low | Low | Low | low | HIGH | HIGH |
| Speed | Low | Low | High | High | High | High |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Function | Parameters | Algorithms Used | Syntax |
| **Wrapper Method** | Forward  Feature  Selection | Sequential  Feature  Selector() | 1.estimator 2.k\_features  3.forward  4.verbose  5.scoring  6.cv | Random Forest Classifier | SequentialFeatureSelector(RandomForestClassifier(n\_jobs **=** 1),  k\_features **=** 15,  forward **= True**,  verbose **=** 2,  scoring **=** 'roc\_auc',  cv = 4) |
| Backward  Feature  Elimination | Sequential  Feature  Selector() | 1.estimator 2.k\_features  3.forward  4.verbose  5.scoring  6.cv | Random Forest Classifier | SequentialFeatureSelector(RandomForestClassifier (n\_jobs **=** 1),  k\_features **=** 15,  forward **= False**,  verbose **=** 2,  floating **= False**,  scoring **=**'roc\_auc',  cv **=** 4) |
| Recurssive  Feature  Elimination | RFECV() | 1.estimator  2.verbose  3.scoring  4.cv | Random Forest Classifier | RFECV(estimator **=** RandomForestClassifier (n\_jobs**=**1),  cv **=** 10,  verbose **=** 2,  scoring **=** 'roc\_auc'  ) |
| Boruta | BorutaPy() | 1.estimator  2.n\_estimators  3.verbose | Random Forest Classifier | BorutaPy(RandomForestClassifier(n\_jobs **=** 1),  n\_estimators **=** 'auto',  verbose **=** 2) |
| Embedded Methods | Logistics  Regression  L1 | Select  From  Model() | 1.estimator | Logistic Regression | SelectFromModel(LogisticRegression(penalty**=**'l1')) |
| Random Forest | Select  From  Model() | 1.estimator | Random  Forest | SelectFromModel(RandomForestClassifier (n\_estimators = 100)) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Method | Function() | Input | Output |
| **Wrapper Method** | Forward  Feature  Selection | Sequential  Feature  Selector() | Total no:of features(**“57”** in our example) | K\_features specified (“**15**” in our example) |
| Backward  Feature  Elimination | Sequential  Feature  Selector() | Total no:of features(“**57**” in our example) | K\_features specified  (“**15**” in our example) |
| Recurssive  Feature  Elimination | RFECV() | Total no:of features(“**57**” in our example) | performs RFE in a cross-validation loop to find the optimal number of features.(“**12”** features in our example) |
| Boruta | BorutaPy() | Total no:of features(“**57**” in our example) | The algorithm reshuffles the data to create shadow features. It eliminates the features that have significantly worst importance than shadow ones(it returns “**4**” features in our example) |
| Embedded Methods | Logistics  Regression  L1 | Select  From  Model() | Total no:of features(“**57**” in our example) | Lasso or L1 has the property that is able to shrink some of the coefficients to zero.  Therefore, that feature can be removed from the model.  (it returns “**54**” in our example) |
| Random Forest | Select  From  Model() | Total no:of features(“**57**” in our example) | At each node,  the tree divides the dataset into 2 buckets, each of them hosting observations that are more similar among themselves and different from the ones in the other bucket.(it returns ‘**4**’ features) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| F  I  LTER    METHODS | Funtion | Usage | Attributes | Syntax | Input | Output |
| VarianceThreshold() | To remove  Constant and quasi constant features | threshold | VarianceThreshold  (threshold **=** 0) | dataset | Features\_set with out constants or quasi\_canstants |
| duplicated() | To\_identify duplicated fetures  But we need to transpose our data set since it finds duplicatd rows. | Subset  Keep | duplicated(subset=None, keep='first’) | dataset | Return boolean Series denoting duplicate rows |
| drop\_duplicates() | Removes duplicated columns | keep | drop\_duplicates(subset=None, keep='first', inplace=False) | dataset | DataFrame with removed dupliate rows |
| Corr() | used to find the pairwise correlation of all columns in the dataframe. | Method  Min\_periods | corr(method='pearson', min\_periods=1) | dataset | Data Frame  With more correlated features(colunms) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Methods | When to use | Accuracy | | | |
| Training | | Test | |
| Before(#57) | After | Before(#57) | After |
| Forward Feature Selection | It is used when the dataset is large and the no of features to be selected are less(<20) | 0.70990 | 0.70766  (on 15 features) | 0.70964 | 0.70911  (on 15 features) |
| Backward  Feature  Elimination | It is used when the dataset is large and the no of features to be selected are less(<20) | 0.70990 | 0.70796  (on 15 features) | 0.70964 | 0.71242  (on 15 features) |
| Recurssive  Feature  Elimination | It is used to select the features automatically and if the cross validation sets are low then it selects more features and viceversa | 0.70990 | 0.70279  (on 12 features) | 0.70964 | 0.70878  (on 12 features) |
| Boruta | It is used to select the optimal no of features automatically and better than Recursive Feature Elimination | 0.70990 | 0.70353  (on 4 features) | 0.70964 | 0.71125  (on 4 features) |
| Logistics  Regression  L1 | It is used when the dataset is large and it automatically selects the features. | 0.70990 | 0.70380  (on 53 features) | 0.70964 | 0.70481  (on 53 features) |
| Random Forest | It is same as Logistic Regresssion but selects optimal no of features | 0.70990 | 0.70353  (on 4 features) | 0.70964 | 0.71125  (on 4 features) |