uday sai

4 Followers About

Feature Engineering : Feature Selection

uday sai Apr 25, 2020 · 6 min read

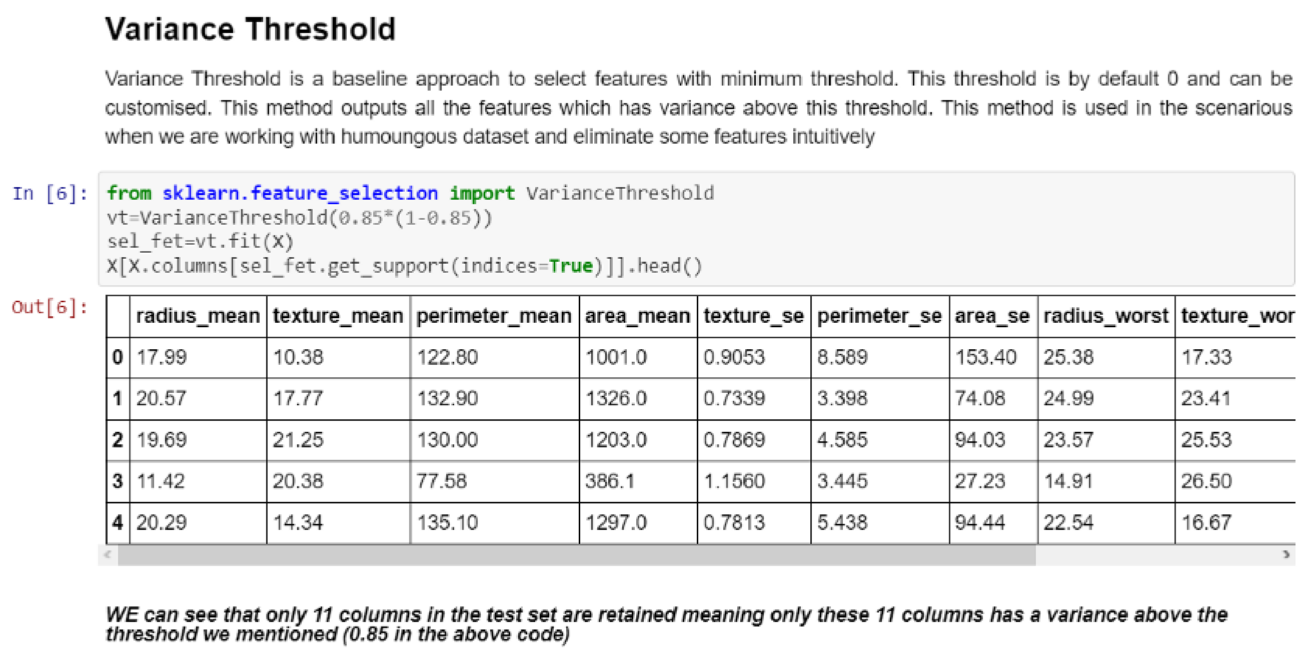
Feature Selection in data science is a crucial aspect for compelling data story. The feature selection framework assists us to identify the most influencing features on the target. However it's the domain expertise which plays a crucial role in determining the feature importance.

Let us explore the feature selection techniques in detail:

> *Remove Low Variance Features*
>
> *Filter methods or Univariate Feature Selection*
>
> *Wrapper Methods*
>
> *Embedded Methods*

Remove Low Variance Features

doesn't meet some threshold. By default it removes all zero-variance features viz., features that have same values in all samples.



Variance Threshold

Variance Threshold is a baseline approach to select features with minimum threshold. This threshold is by default 0 and can be customised. This method outputs all the features which has variance above this threshold. This method is used in the scenarious when we are working with humoungous dataset and eliminate some features intuitively

```
In [6]: from sklearn.feature_selection import VarianceThreshold
        vt=VarianceThreshold(0.85*(1-0.85))
        sel_fet=vt.fit(X)
        X[X.columns[sel_fet.get_support(indices=True)]].head()
```

Out[6]:

	radius_mean	texture_mean	perimeter_mean	area_mean	texture_se	perimeter_se	area_se	radius_worst	texture_wor
0	17.99	10.38	122.80	1001.0	0.9053	8.589	153.40	25.38	17.33
1	20.57	17.77	132.90	1326.0	0.7339	3.398	74.08	24.99	23.41
2	19.69	21.25	130.00	1203.0	0.7869	4.585	94.03	23.57	25.53
3	11.42	20.38	77.58	386.1	1.1560	3.445	27.23	14.91	26.50
4	20.29	14.34	135.10	1297.0	0.7813	5.438	94.44	22.54	16.67

WE can see that only 11 columns in the test set are retained meaning only these 11 columns has a variance above the threshold we mentioned (0.85 in the above code)

Image: Variance Threshold

Filter Methods (or) Univariate Feature Selection

Statistical Tests are used to select features in this univariate feature selection methods. Some of the statistical tests are chi2 (pronounced as ki-square in kite), ANOVA (Analysis of Variance), Correlation (Pearson, Spearman, Kendall), LDA (Linear Discriminant Analysis), Mutual Information Gain are some of the statistical methods used for univariate feature selection methods.

Python's Sklearn library exposes feature selection routines as objects that implement the transform method. Besides we can define custom routines by using scipy.stats package which holds methods to implement all the above mentioned statistical tests. Some of the sklearn feature selection methods are: SelectKBest, SelectPercentile, SelectFpr, SelectFdr and GenericUnivariateSelect. We can use routine of our choice, which takes a scoring function (statistical test) as input which returns univariate scores and p-values (or only scores for SelectKBest and SelectPercentile) as output. However there are some conventions on which statistical test to perform based on feature and target variables:

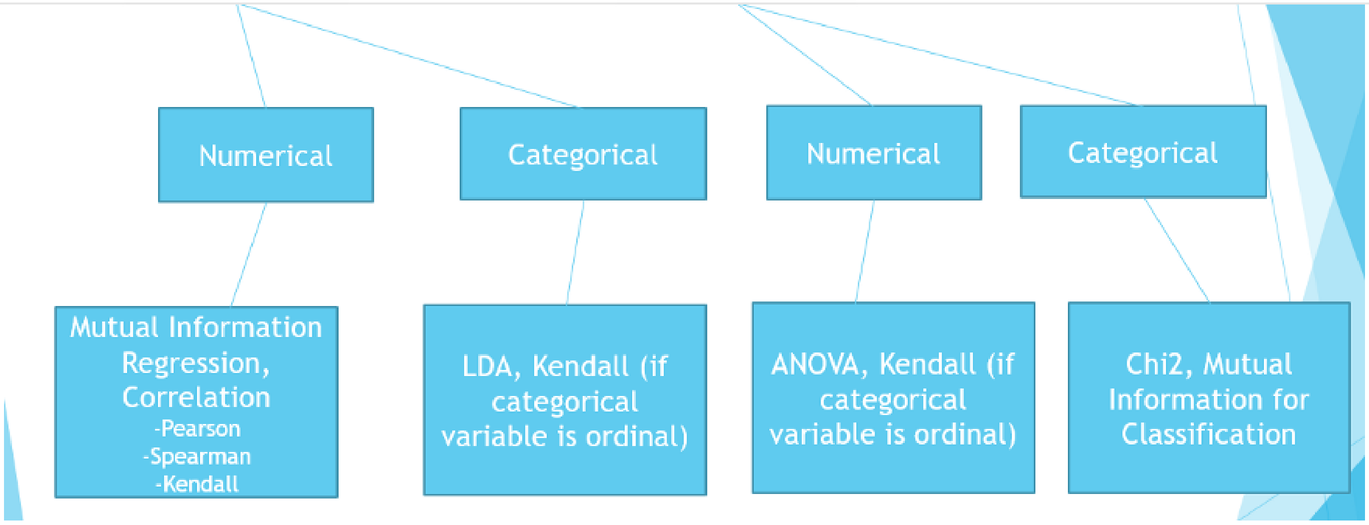

Numerical
Categorical
Numerical
Categorical
Mutual Information Regression, Correlation -Pearson -Spearman -Kendall
LDA, Kendall (if categorical variable is ordinal)
ANOVA, Kendall (if categorical variable is ordinal)
Chi2, Mutual Information for Classification

Image : Statistical Tests for Univariate Feature Selection Methods

Correlation : It is a statistical measure of association between two numerical variables. It takes values between -1 and +1, with -1 being strong negative association and +1 being strong positive association. There are three variants of correlation which takes different assumptions.

Pearson Correlation is a statistical measure based on linear measure and has some rigid assumptions that variables have gaussian distribution.

Spearman rank Correlation is based on monotonic relationship between nominal and ordinal variables. This is a non-parametric (which means the variables need not be normally distributed) correlation

Kendall Correlation is based on concordant and discordant relationship between nominal and ordinal variables. Like Spearman, this is also a non-parametric correlation and works best with small samples.

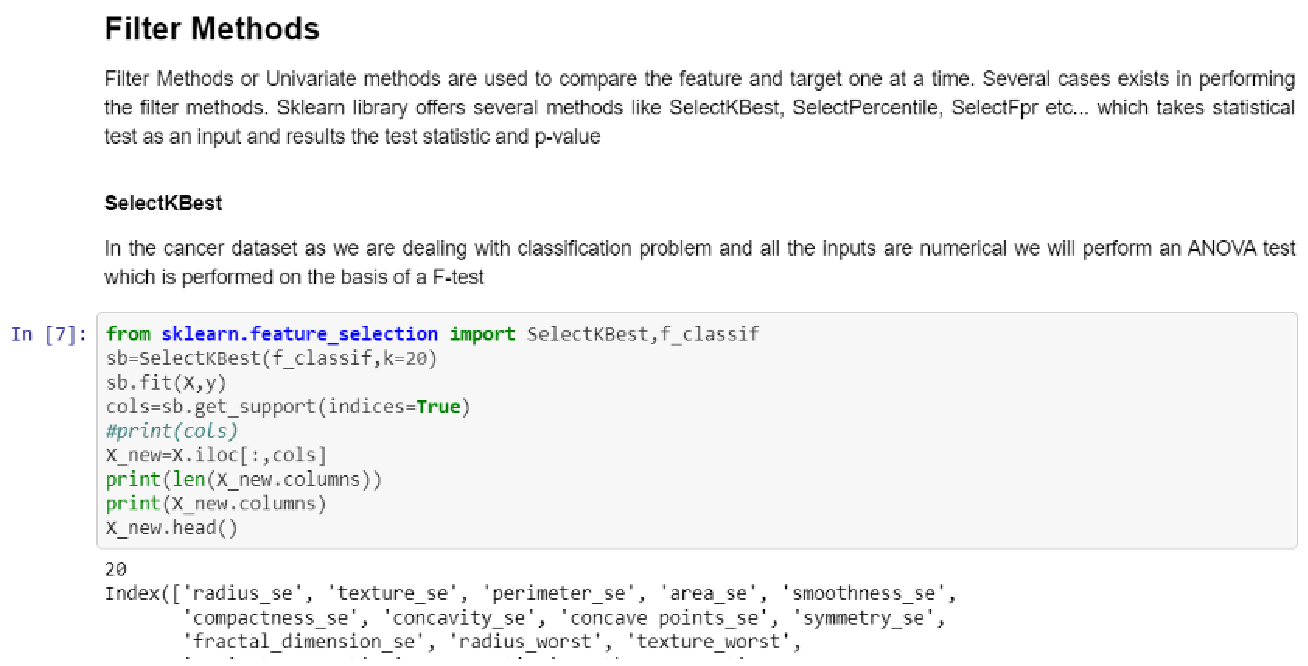
All the three correlations are available in scipy.stats and can be used in custom defined routines based on insights in EDA.

Linear Discriminant Analysis : LDA is a supervised linear transformation method which can be used for feature selection. On the basis of class, we construct d-dimensional mean vectors followed by in-between and within class scatter matrices. This

ANOVA : Analysis of Variance is a statistical measure to compare the means of two or different groups. However, we will be discussing the One-Way ANOVA here. The means of different labels of categorical variable are compared based on F-test. F-test is a statistical test for comparison of variance with Null hypothesis being the groups does not have variance and alternate hypothesis being the groups have variance. The rationale is the groups which have variance influence the target variable and are better qualified features to predict dependent variable.

Chi2 : Chi2 test of independence is a statistical measure based on the contingency table (equivalent to crosstab). The observed and expected values of the table are calculated and the features which as high chi2 values being most influential features.

Mutual Information : Mutual Information is calculated between two variables and measures the reduction in uncertainty for one variable given a known value of the other variable. A quantity called mutual information measures the amount of information one can obtain from one random variable given another. Mutual information is always larger than or equal to zero, where the larger the value, the greater the relationship between the two variables. If the calculated result is zero, then the variables are independent. Mutual information is often used as a general form of a correlation coefficient, e.g. a measure of the dependence between random variables.



Filter Methods

Filter Methods or Univariate methods are used to compare the feature and target one at a time. Several cases exists in performing the filter methods. Sklearn library offers several methods like SelectKBest, SelectPercentile, SelectFpr etc... which takes statistical test as an input and results the test statistic and p-value

SelectKBest

In the cancer dataset as we are dealing with classification problem and all the inputs are numerical we will perform an ANOVA test which is performed on the basis of a F-test

```
In [7]: from sklearn.feature_selection import SelectKBest,f_classif
        sb=SelectKBest(f_classif,k=20)
        sb.fit(X,y)
        cols=sb.get_support(indices=True)
        #print(cols)
        X_new=X.iloc[:,cols]
        print(len(X_new.columns))
        print(X_new.columns)
        X_new.head()
```

```
20
Index(['radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
```

Image:Filter Methods (SelectKBest)

Wrapper Methods

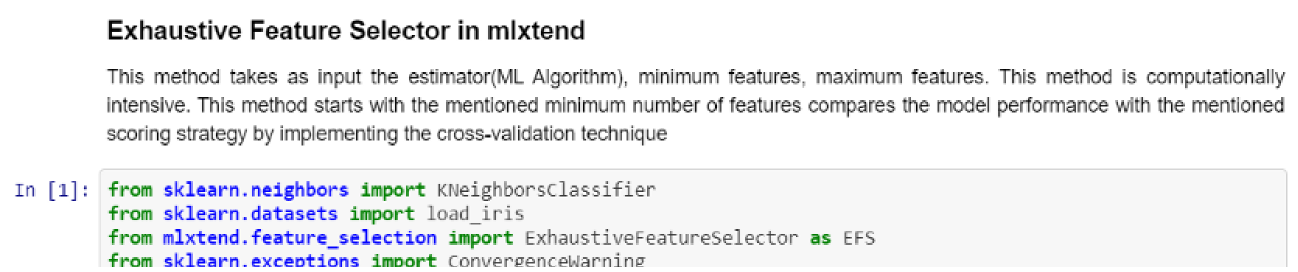
These methods search for best subset of features which are the most- influential features, construct a machine learning model and evaluate performance. The methods include heuristics like forward or backward pass, stochastic gradient descent or hill-climbing problems. Consequently, these techniques make the wrapper methods computationally intensive. Let us look at some wrapper methods:

Forward Feature Selection : In this technique we start with a single feature, construct a ml model, evaluates the performance. The technique iterates over all the features, add one feature at a time and evaluates the model performance. If the model results in better performance it retrieves the feature or else eliminate the feature.

Backward Feature Elimination : This is a vice-versa of the Forward feature selection method. Unlike forward selection this method uses all the features to construct an ml model, evaluates the performance and eliminate the feature one at a time, re-evaluate the ml model and based on the results we retrieve a feature or eliminate it.

Recursive Feature Selection : The Recursive Feature Elimination (or RFE) works by recursively removing attributes and building a model on those attributes that remain. It uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute. RFE is available in sklearn package and can be imported as from sklearn.feature_selection import RFE statement.

Exhaustive Feature Selection : This method is available in mlxtend package. This method takes namely two parameters 1 and 'N' which are minimum and maximum features that we want our model to come up with. This method can be imported using from mlxtend.feature_selection import ExhaustiveFeatureSelector

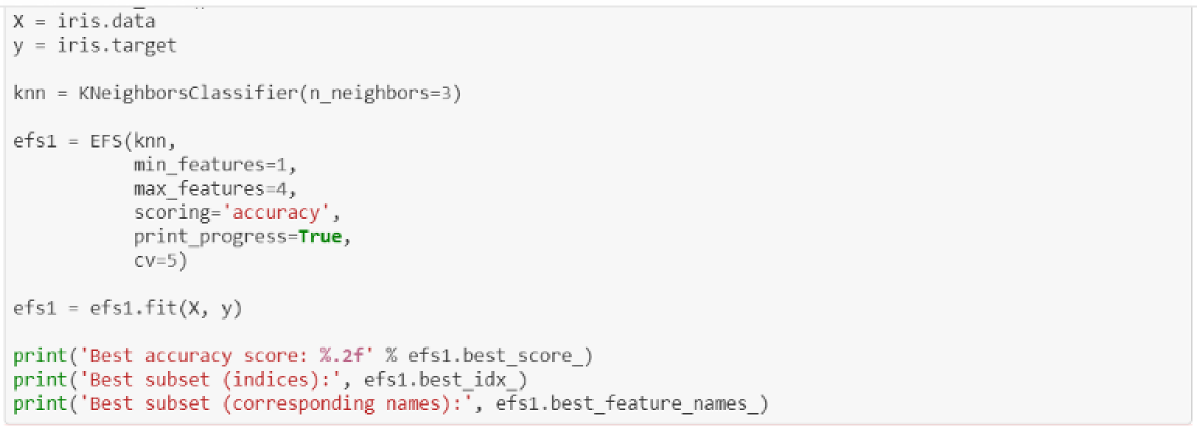


Exhaustive Feature Selector in mlxtend

This method takes as input the estimator(ML Algorithm), minimum features, maximum features. This method is computationally intensive. This method starts with the mentioned minimum number of features compares the model performance with the mentioned scoring strategy by implementing the cross-validation technique

```
In [1]: from sklearn.neighbors import KNeighborsClassifier
        from sklearn.datasets import load_iris
        from mlxtend.feature_selection import ExhaustiveFeatureSelector as EFS
        from sklearn.exceptions import ConvergenceWarning
```

Open in app

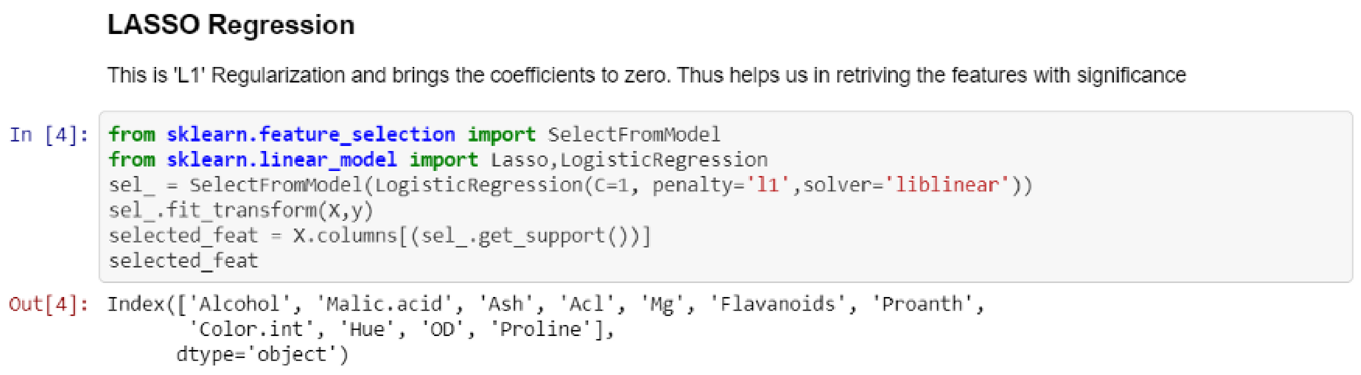


```
X = iris.data
y = iris.target

knn = KNeighborsClassifier(n_neighbors=3)

efs1 = EFS(knn,
          min_features=1,
          max_features=4,
          scoring='accuracy',
          print_progress=True,
          cv=5)

efs1 = efs1.fit(X, y)

print('Best accuracy score: %.2f' % efs1.best_score_)
print('Best subset (indices):', efs1.best_idx_)
print('Best subset (corresponding names):', efs1.best_feature_names_)
```

Image: Wrapper Methods (ExhaustiveFeatureSelector)

Embedded Methods

This is a widely used method to capture the subset of features which are most influential on the target variable. These methods use SelectFromModel method in the sklearn package. The algorithms used can be Regularization (Ridge, LASSO and Elastic-Net methods) and feature_importances_ of any tree based algorithms like XGBoost, CatBoost and Random Forest algorithms.



LASSO Regression

This is 'L1' Regularization and brings the coefficients to zero. Thus helps us in retriving the features with significance

```
In [4]: from sklearn.feature_selection import SelectFromModel
        from sklearn.linear_model import Lasso,LogisticRegression
        sel_ = SelectFromModel(LogisticRegression(C=1, penalty='l1',solver='liblinear'))
        sel_.fit_transform(X,y)
        selected_feat = X.columns[(sel_.get_support())]
        selected_feat

Out[4]: Index(['Alcohol', 'Malic.acid', 'Ash', 'Acl', 'Mg', 'Flavanoids', 'Proanth',
               'Color.int', 'Hue', 'OD', 'Proline'],
              dtype='object')
```

Image: Embedded Methods (LASSO)

Please refer my GitHub link below for full code:

udaysai50/Full-Stack-Data-Science

Permalink Dismiss GitHub is home to over 40 million developers working together to host and review code, manage…

github.com

An Introduction to Feature Selection - Machine Learning Mastery

Which features should you use to create a predictive model? This is a difficult question that may require deep…

machinelearningmastery.com

Hands-on with Feature Selection Techniques: Wrapper Methods

Part 3: Forward feature selection, backward feature elimination, exhaustive feature selection, and bidirectional search

heartbeat.fritz.ai

Machine Learning   Data Science   Feature Selection   Feature Engineering   Python