Feature-Engineering

Univariate Feature Selection

In [1]:

```
# Feature Extraction with Univariate Statistical Tests (Chi-squared for classification)
from pandas import read_csv
from numpy import set_printoptions
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
# load data
filename = 'C:/Users/Ashraf/Documents/Datafiles/pima-indians-diabetes.data.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
# feature extraction
test = SelectKBest(score_func=chi2, k=4)
fit = test.fit(X, Y)
# summarize scores
set_printoptions(precision=3)
print(fit.scores_)
features = fit.transform(X)


#For regression: f_regression, mutual_info_regression
#For classification: chi2, f_classif, mutual_info_classif
```

```
[ 111.52  1411.887   17.605   53.108 2175.565  127.669    5.393  181.304]
```

Recursive Feature Elimination

In [2]:

```
# Feature Extraction with RFE
from pandas import read_csv
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
# load data
filename = 'C:/Users/Ashraf/Documents/Datafiles/pima-indians-diabetes.data.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
# feature extraction
model = LogisticRegression(max_iter=400)
rfe = RFE(model, 3)
fit = rfe.fit(X, Y)
```

```
C:\Users\Ashraf\anaconda3\lib\site-packages\sklearn\utils\validation.py:70:
FutureWarning: Pass n_features_to_select=3 as keyword args. From version 1.0
(renaming of 0.25) passing these as positional arguments will result in an e
rror
  warnings.warn(f"Pass {args_msg} as keyword args. From version "
```

In [3]:

```
#Num Features:
fit.n_features_
```

Out[3]:

```
3
```

In [4]:

```
#Selected Features:
fit.support_
```

Out[4]:

```
array([ True, False, False, False, False,  True,  True, False])
```

In [5]:

```
# Feature Ranking:
fit.ranking_
```

Out[5]:

```
array([1, 2, 4, 6, 5, 1, 1, 3])
```

Feature Importance using Decision Tree

In [6]:

```
# Feature Importance with Extra Trees Classifier
from pandas import read_csv
from sklearn.tree import  DecisionTreeClassifier
# load data
filename = 'C:/Users/Ashraf/Documents/Datafiles/pima-indians-diabetes.data.csv'
names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class']
dataframe = read_csv(filename, names=names)
array = dataframe.values
X = array[:,0:8]
Y = array[:,8]
# feature extraction
model = DecisionTreeClassifier()
model.fit(X, Y)
print(model.feature_importances_)
```

```
[0.063 0.311 0.083 0.023 0.05  0.224 0.132 0.114]
```

In [ ]: