**Experiment 4: Feature Selection**

**Objective** :To learn and implement the feature selection techniques

**Time Required** : 3 hrs

**Programming Language** : Python

**Software Required** : Anaconda

**Introduction**

Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model**.** The data features that you use to train your machine learning models have a huge influence on the performance you can achieve. Irrelevant or partially relevant features can negatively impact model performance.

Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in. Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features.

The most commonly and easy to use Feature selection techniques which provide good results are as follows:

1. **Univariate Selection:**

Statistical tests can be used to select those features that have the strongest relationship with the output variable.

The scikit-learn library provides the [SelectKBest](http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html" \l "sklearn.feature_selection.SelectKBest" \t "_blank) class that can be used with a suite of different statistical tests to select a specific number of features.

**TASK 1:**

Download the dataset from this link: <https://www.kaggle.com/iabhishekofficial/mobile-price-classification#train.csv>

**Description of the variables in dataset:**

*battery\_power:* Total energy a battery can store in one time measured in mAh

*blue:* Has Bluetooth or not

*clock\_speed:* the speed at which microprocessor executes instructions

*dual\_sim:* Has dual sim support or not

*fc:* Front Camera megapixels

*four\_g:* Has 4G or not

*int\_memory:* Internal Memory in Gigabytes

*m\_dep:* Mobile Depth in cm

*mobile\_wt:* Weight of mobile phone

*n\_cores:* Number of cores of the processor

*pc:* Primary Camera megapixels

px\_height

Pixel Resolution Height

*px\_width:* Pixel Resolution Width

*ram:* Random Access Memory in MegaBytes

*sc\_h:* Screen Height of mobile in cm

*sc\_w:* Screen Width of mobile in cm

*talk\_time:* the longest time that a single battery charge will last when you are

*three\_g:* Has 3G or not

*touch\_screen:* Has touch screen or not

*wifi:* Has wifi or not

*price\_range:* This is the target variable with a value of 0(low cost), 1(medium cost), 2(high cost) and 3(very high cost).

Use the chi-squared (chi²) statistical test for non-negative features to select 10 of the best features from the above dataset which is used for Mobile Price Range Prediction.

1. **Feature Importance**

You can get the feature importance of each feature of your dataset by using the feature importance property of the model. Feature importance gives you a score for each feature of your data, the higher the score more important or relevant is the feature towards your output variable.

*Feature importance* is an inbuilt class that comes with Tree Based Classifiers.

**TASK 2:**

Load the dataset again and use Extra Tree Classifier for extracting the top 10 features for the dataset and plot your results.

* To import the Extra Tree Classifier, use the following command:

from sklearn.ensemble import ExtraTreesClassifier

* Use the following inbuilt class for feature importances:

feature\_importances\_

1. **Correlation Matrix with Heatmap**

Correlation states how the features are related to each other or the target variable. Correlation can be positive (increase in one value of feature increases the value of the target variable) or negative (increase in one value of feature decreases the value of the target variable)

Heatmap makes it easy to identify which features are most related to the target variable.

**TASK 3:**

Load the dataset and plot heatmap of correlated features using the seaborn library.

* To import the seaborn library, use the following command:

import seaborn as sns

* To get correlations of each features in dataset, use the following command:

corrmat = data.corr()

* To plot heat map, use the following command:

g=sns.heatmap(corrmat , annot=True, cmap="RdYlGn")

After plotting the results, see how the price range, in the last row, is correlated with other features.