Experiment No. 3 Title: To extract features from given data set and establish training data.

Objectives:

1. To learn how to prepare dataset
2. to understand steps include to upload dataset
3. To learn how to execute python program
4. To get top 10 best features

data set of Wine Quality dataset

```
import pandas as pd

import numpy as np

from sklearn.feature_selection import SelectKBest

from sklearn.feature_selection import chi2


df = pd.read_csv('winequality-red.csv', sep = ';')

df.head()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulp
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-null   float64
```

```
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

```
X = df.iloc[:, 0:11]
y = df.iloc[:, -1:]
```

```
y.value_counts()
```

```
quality
5          681
6          638
7          199
4           53
8           18
3           10
dtype: int64
```

```
best_features = SelectKBest(score_func= chi2, k=5)
```

```
fit = best_features.fit(X, y)
```

```
fit.scores_
```

```
array([1.12606524e+01, 1.55802891e+01, 1.30256651e+01, 4.12329474e+00,
       7.52425579e-01, 1.61936036e+02, 2.75555798e+03, 2.30432045e-04,
       1.54654736e-01, 4.55848775e+00, 4.64298922e+01])
```

```
df_scores = pd.DataFrame(fit.scores_)
df_scores
```

```
X_columns = pd.DataFrame(X.columns)
X_columns
```

```
df_best_fet = pd.concat([df_scores, X_columns], axis=1)
df_best_fet
```

```
df_best_fet.columns = ['scores', 'features']
df_best_fet
```

```
df_best = df_best_fet.sort_values('scores', axis=0, ascending = False)
df_best.head(5)
```

OR

```
df_best_fet.nlargest(5, 'scores')
```

```
best_fet_list = list(df_best_fet.nlargest(5, 'scores')['features'].values)
```

```
df[best_fet_list]
```