Machine Learning
Assignment

Name: K.R.Ram Ganesh -110120090 (ICE dept 2nd year)
Ajayvishagan P - 110120004 (ICE dept 2nd year)

WATER POTABILITY ANALYSIS

Introduction:

Fresh water is the primary source of human health, prosperity, and security. By around 2050 the world's population is expected to reach about nine billion. Assuming that standards of living continue to rise, the requirement of potable water for human consumption will amount to the resources of about three planet Earths. A key United Nations report indicates that water shortages will affect 2.3 billion people or 30% of the world's population in four dozen nations by 2025. Already, the crisis of potable water in most developing countries is creating public health emergencies of staggering proportions. In Bangladesh, for example, it is officially recognized by the government of Bangladesh that 50% of the country's approximately 150 million people, are at risk of arsenic poisoning from groundwater used for drinking. Recently, the government of Bangladesh, in its Action Plan for Poverty Reduction, stated its desire to ensure 100% access to pure drinking water across the region within the shortest possible time frame . This is also consistent with key goals of the Millennium Development Goal “Eradication of extreme poverty and hunger” and “Halving by 2015, the proportion of people without sustainable access to safe drinking water”. Whether this is achievable within the stated time is debatable, but it clearly delineates the state of the world we live in. - Abul Hussam, in Monitoring Water Quality, 2013

This notebook will explore the different features related to water potability, Modeling, and predicting water potability.

Dataset used:- water_potability

Problem statement:

In this project , we have built a linear regression model and calculated bias , variance . Later the feature selection was done and bias and variance is calculated on that.

Theory:

Linear Regression:-Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.

Bias:-The bias is known as the difference between the prediction of the values by the ML model and the correct value. Being high in biasing gives a large error in training as well as testing data. Its recommended that an algorithm should always be low biased to avoid the problem of underfitting.

Variance:-The variability of model prediction for a given data point which tells us spread of our data is called the variance of the model.

The model with high variance has a very complex fit to the training data and thus is not able to fit accurately on the data which it hasn't seen before. As a result, such models perform very well on training data but has high error rates on test data.

When a model is high on variance, it is then said to as **Overfitting of Data**. Overfitting is fitting the training set accurately via complex curve and high order hypothesis but is not the solution as the error with unseen data is high.

Building a Model:

Step 1:

Importing the required libraries. pandas is issued for reding for csv file . df.head()- It is gives the first 5 rows in it.

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
df=pd.read_csv('waterPotability.csv')
df.head()
```

Out[2]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	NaN	204.890456	20791.31898	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.05786	6.635246	NaN	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.54173	9.275884	NaN	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.41744	8.059332	356.886136	363.266516	18.436525	100.341674	4.628771	0
4	9.092223	181.101509	17978.98634	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0

Step 2:

Replacing all the not available values with mean value of that column.

In [3]:
```
col=['ph','Hardness','Solids','Chloramines','Sulfate','Conductivity','Organic_carbon','Trihalomethanes','Turbidity']
for i in col:
    df[i]=df[i].replace(0,np.NAN)
    mean=df[i].mean(skipna=True)
    df[i] = df[i].replace(np.NAN,mean)
df.head()
```

Out[3]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	7.083338	204.890456	20791.31898	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.05786	6.635246	333.775777	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.54173	9.275884	333.775777	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.41744	8.059332	356.886136	363.266516	18.436525	100.341674	4.628771	0
4	9.092223	181.101509	17978.98634	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0

Step 3:

Importing libraries from sklearn like train_test_split,Linear regression and bias_variance_decomp.

train_test_split :- it splits the given dataset into training and test datas with specific test size.

LinearRegression : - It helps in building a linear regression model.

bias_variance_decomp :- Helps in calculating bias and variance.

Before feature selection:
Bias :- 0.240163
Variance :- 0.001437

In [4]:
```
# estimate the bias and variance for a regression model
from pandas import read_csv
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from mlxtend.evaluate import bias_variance_decomp
# separate into inputs and outputs
data = df.values
X, y = data[:, :-1], data[:, -1]
# split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=1)
```

In [5]:
```
X_test
y_train
```

Out[5]:
```
array([0., 1., 1., ..., 1., 0., 0.])
```

In [6]:
```
# define the model
model = LinearRegression()
# estimate bias and variance
mse, bias, var = bias_variance_decomp(model, X_train, y_train, X_test, y_test, loss='mse', num_rounds=100, random_se
```

In [7]:
```
# summarize results
print('Bias: %f' % bias)
print('Variance: %f' % var)
```

```
Bias: 0.240163
Variance: 0.001437
```

Step 4:

Matplotlib - helps to plotting graph

SelectKbest - it selects k best features and make a feature selection dataset.

Select feature function built a feature selected training and test dataset.

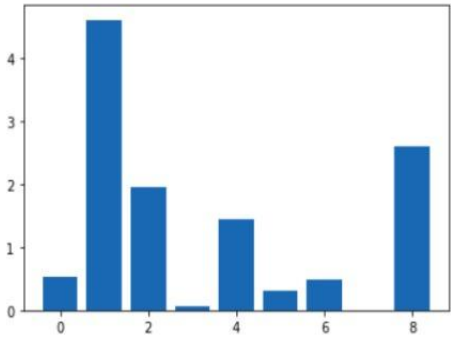
In [8]:
```
from sklearn.datasets import make_regression
from sklearn.model_selection import train_test_split
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_regression
from matplotlib import pyplot
```

In [9]:
```
# feature selection
def select_features(X_train, y_train, X_test):
    # configure to select all features
    fs = SelectKBest(score_func=f_regression, k=8)
    # learn relationship from training data
    fs.fit(X_train, y_train)
    # transform train input data
    X_train_fs = fs.transform(X_train)
    # transform test input data
    X_test_fs = fs.transform(X_test)
    return X_train_fs, X_test_fs, fs
```

Step 5:

In [11]:
```
# feature selection
X_train_fs, X_test_fs, fs = select_features(X_train, y_train, X_test)
# what are scores for the features
for i in range(len(fs.scores_)):
    print('Feature %d: %f' % (i, fs.scores_[i]))
# plot the scores
pyplot.bar([i for i in range(len(fs.scores_))], fs.scores_)
pyplot.show()
```

```
Feature 0: 0.526213
Feature 1: 4.602876
Feature 2: 1.947322
Feature 3: 0.051360
Feature 4: 1.430491
Feature 5: 0.296669
Feature 6: 0.481509
Feature 7: 0.000025
Feature 8: 2.593094
```



In [12]:
```
# define the model
model = LinearRegression()
# estimate bias and variance
mse, bias, var = bias_variance_decomp(model, X_train_fs, y_train, X_test_fs, y_test, loss='mse', num_rounds=100, ran
```

In [13]:
```
# summarize results
print('Bias: %f' % bias)
print('Variance: %f' % var)
```

```
Bias: 0.240153
Variance: 0.001307
```

Calculated Bias and Variance of the feature selected model.

Conclusion:

The bias and variance of the feature selected model is less than the actual dataset.