UE17CS302-
Machine Learning
Project Report

By:

Sujay Gad:PES1201700177

Mayur R B:PES1201700714

Jnanesh D:PES1201701822

Introduction-

Dataset information-

Dataset is a binary classification dataset which consists of 699 rows and 11 columns including the target class.

Attributes present in the dataset are-

1)Id number

2)Clump thickness

3)Uniformity of cell size

4)Uniformity of cell shape

5)Marginal adhesion

6)Single epithelial cell size

7)Bare nuclei

8)Bland chromatin

9)Normal nucleoli

10)Mitosis

11)Target class

Range of values in all the columns except the first column is from 1-10. Bare nuclei column consists of 16 unknown values represented by '?'.Target class consists of 2 distinct values ie - 2 for Benign and 4 malignant.

Data Cleaning and preprocessing-

Unknown values in bare nuclei column were replaced by mean of that column taking into consideration only the known values.

Id column in the original dataset was removed since it has no relation with the rest of the dataset.

```
import math
df2=df.drop(['id'], axis=1)
df4=df2
df5=df4

s=0
k=0
for index,row in df.bare_nuclei.iteritems():
    if(row!='?'):
        s=s+int(row)
        k+=1
p=round(s/k)

df1=df['bare_nuclei'].replace('?',p)
df
df.to_csv('clean_ml.csv')
df4=pd.read_csv('clean_ml.csv')
```

Dimensionality reduction-

Using PCA we observe that 99% variance in the dataset is explained by 8 of the 9 columns present in the dataset.1 column can be removed with any loss of information.

```
array([65.42, 74.05, 80.04, 85.18, 89.4 , 92.81, 96.09, 99.01, 99.99])
```

Using Correlation matrix we observe that correlation between uniformity of cell size and uniformity of cell shape is very high. Both of these attributes have similar correlation with the target variable can remove uniformity of cell size from the original dataset to form another dataset.

```
df['uniformity_of_cell_size'].corr(df['uniformity_of_cell_shape'])
```

```
0.9068819130525925
```

Addition of random noise-

To check the performance of the model against outliers we add 20 instances of random noise into the dataset.

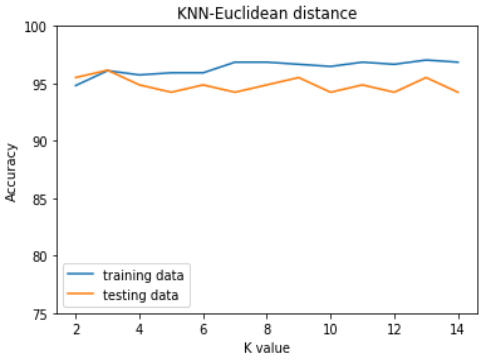
```
df2=df
for i in range(20):
    df2=df2.append(pd.Series([random.randint(1,10),random.randint(1,10)
df2.to_csv('noise_added_another.csv')
```

ML Models-

We have used 2 ML models KNN and decision tree. We have various graphs to compare inter model as well as intra model performance. In KNN we have used 3 different distance measures euclidean,manhattan and Cosine similarity to compare performance. In decision tree we have compared how the performance changes when parameters such as maximum depth change. We have also compared performance of all the models on 3 different datasets ie - Original dataset, dataset after dimensionality reduction and dataset after the addition of random noise.

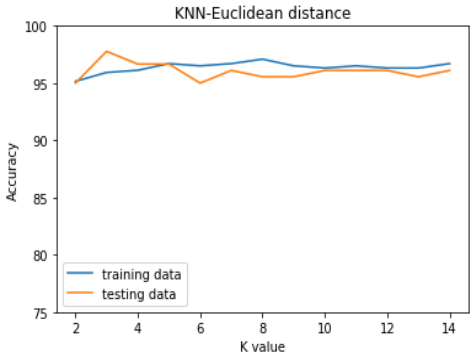
KNN-

1)KNN using euclidean distance-



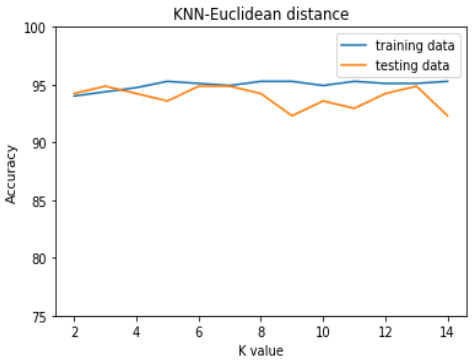

```
avg train accuracy:  96.36752136752138
avg test accuracy:   94.87179487179489
```

Original dataset




```
avg train accuracy:  96.37745974955277
avg test accuracy:   96.02564102564102
```

Dimensionally reduced dataset

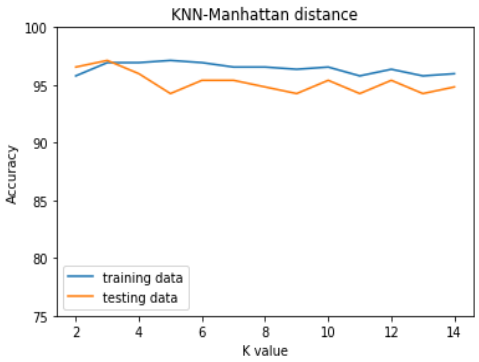



```
avg train accuracy:  94.98327759197325
avg test accuracy:   93.93491124260355
```

Noise added dataset

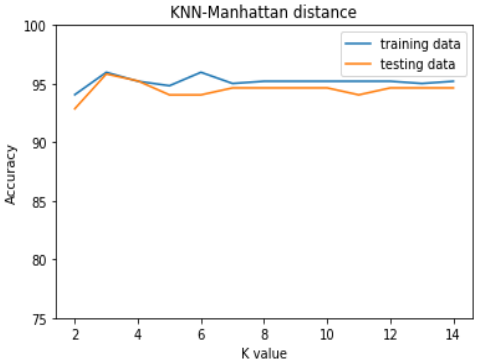
All 3 models perform well however accuracy decreases slightly when noise is added to the dataset.

2)KNN using manhattan distance-



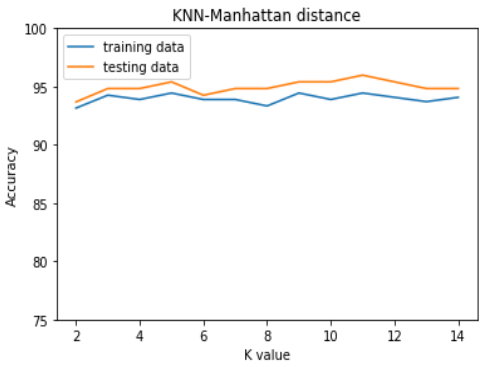

```
avg train accuracy:  96.43383436486884
avg test accuracy:   95.22546419098141
```

Original dataset




```
avg train accuracy:  95.1812555260831
avg test accuracy:   94.5054945054945
```

Dimensionally reduced dataset



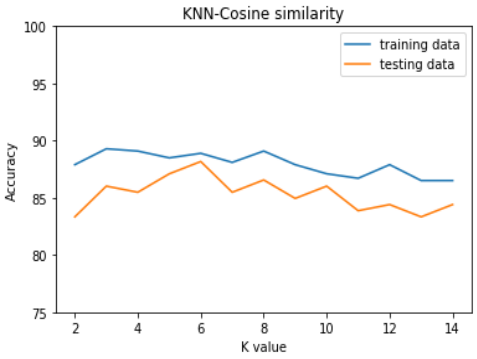

```
avg train accuracy:  93.96011396011396
avg test accuracy:   94.96021220159152
```

Noise added dataset

When random noise is added to the dataset test accuracy overtakes training accuracy.

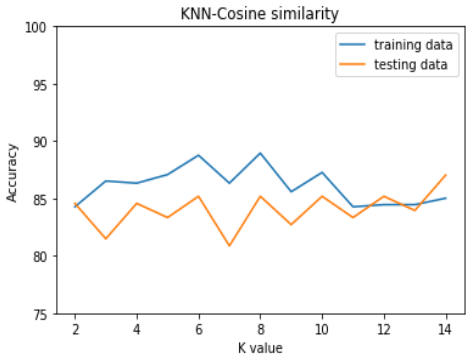
Overall performance is similar to the performance when euclidean distance is used as a distance measure.

3)KNN using cosine similarity-



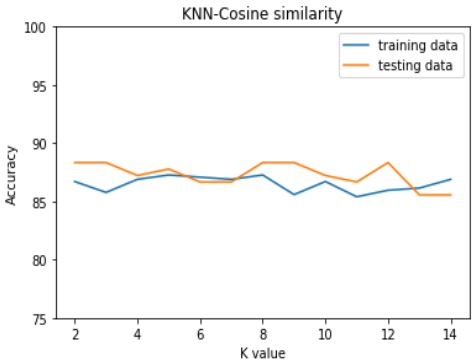

```
avg train accuracy:  87.95787545787545
avg test accuracy:   85.31844499586434
```

Original dataset




```
avg train accuracy:  86.09910688562374
avg test accuracy:   84.04558404558405
```

Dimensionally reduced dataset




```
avg train accuracy:  86.50244886199943
avg test accuracy:   87.30769230769232
```

Noise added dataset

Cosine similarity gives worse results compared to other distance measures used. Similar to the previous case when random noise is added to the dataset test accuracy overtakes training accuracy since the model trains well on the dataset which contains noise.

Conclusions from KNN model-

1)Overall for the given dataset euclidean or manhattan distance is preferred over cosine similarity as a distance measure for finding k nearest neighbours.

2)Performing dimensionality reduction makes negligible difference to the accuracy of the model.

3)Adding random noise to the dataset reduces the training accuracy thereby reducing the risk of overfitting.

Decision tree-

Using decision tree we have compared the performance of original dataset with dimensionally reduced one as well as the dataset containing random noise.

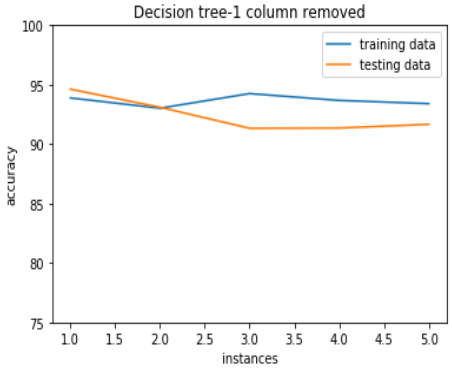
Parameters involved in decision tree model include-

○ Maximum depth - Maximum depth up to which decision tree is allowed to grow.

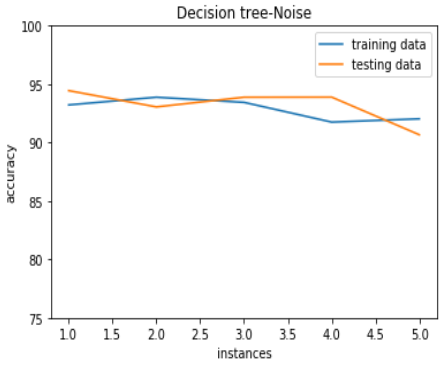
○ Minimum size - If less than 2 children of current node contain less than or equal to minimum size training instances than no further splitting is performed.

Decision tree performance-


```
Avg training accuracy:  94.22221
Avg testing accuracy:   93.18560000000001
```




```
Avg training accuracy:  93.649734
Avg testing accuracy:   92.415706
```



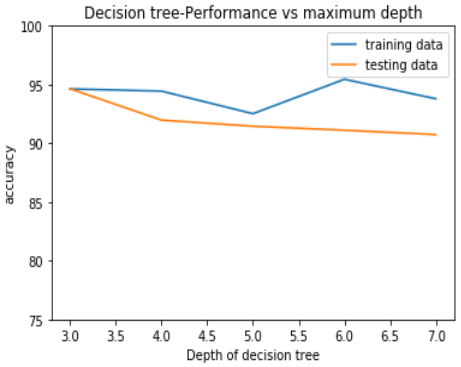

```
Avg training accuracy:  92.86399999999999
Avg testing accuracy:   93.18560000000001
```

Performance is highest for original dataset and dimensionality reduction as well as random noise has minimal effect on performance of decision tree model.

Hence, we can observe that decision trees are robust to outliers.

Similar to KNN model when random noise is added test accuracy overtakes training accuracy.

How decision tree performance varies with parameter values-




```
Avg training accuracy:  94.166
Avg testing accuracy:   91.982
```


```
Avg training accuracy:  92.4475
Avg testing accuracy:   91.3025
```

As the depth of tree increases performance does not reduce drastically as performance is also bounded by the other parameter which is minimum number of training instances which should be present in both the children of the current node in order for split to occur.

Conclusion-

1)Performance of KNN using euclidean and manhattan distance is high and very similar to the performance of decision tree.

2)Performance of KNN using cosine similarity as distance measure gives poor results compared to the other models.

3)Dimensionally reducing the dataset and adding noise to the dataset reduces the overall performance of the model by very small amount.

4)When noise is added training accuracy decreases below test accuracy as the training model learns the noise present in training dataset well.