Name – Pankaj Chawla
Roll No. - 001811001052
BE IT 4th Year Machine Learning Assignment 1

All steps in Google Collab – Google_Collab

Procedure:-

- I first loaded the datasets from the CSV files available in the links and analyzed a bit of the data using pandas
- Then I imported all the relevant libraries that I was going to use for the assignment
- All three datasets were dealt with in the following manner:-
- First we used Naive Bayes Classification (Gaussian, Multinomial, Bernoulli). Bernoulli gave the worst performance both with and without parameter tuning
- Then we used Decision Tree Classifier (both gini and entropy) for both with and without parameter tuning.
- All performance metrics such as recall score, F1 score, precision score, and accuracy score were measured. I had really good results with the breast cancer dataset and the Iris Dataset with > 90% accuracy but the diabetes dataset had a dismal picture to it with ~60% accuracy on determining the gender attribute.
- Some sample screenshots are attached below for reference. As stated the mail already contains the decision tree images generated. However, since the notebook is pretty large I am unable to attach all the screenshots.

Screenshots:-

Loading data and importing libraries

I have downloaded the relevant datasets and then uploaded them to my drive, here I will be importing them to this colab notebook

```
[277] from google.colab import drive
      drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
[278] import pandas as pd
      iris = pd.read_csv("/content/drive/MyDrive/iris.csv")
      diabetes = pd.read_csv("/content/drive/MyDrive/diabetes_tab_separated.txt",delimiter="\t")
      breast_cancer = pd.read_csv("/content/drive/MyDrive/breast-cancer-wisconsin.csv")
```

```
iris.head(10)
```

	sepal_length	sepal_width	petal_length	petal_width	class
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa
5	5.4	3.9	1.7	0.4	Iris-setosa

```
[280] diabetes.head(10)
```

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
0	59	2	32.1	101.0	157	93.2	38.0	4.00	4.8598	87	151
1	48	1	21.6	87.0	183	103.2	70.0	3.00	3.8918	69	75
2	72	2	30.5	93.0	156	93.6	41.0	4.00	4.6728	85	141
3	24	1	25.3	84.0	198	131.4	40.0	5.00	4.8903	89	206
4	50	1	23.0	101.0	192	125.4	52.0	4.00	4.2905	80	135
5	23	1	22.6	89.0	139	64.8	61.0	2.00	4.1897	68	97
6	36	2	22.0	90.0	160	99.6	50.0	3.00	3.9512	82	138
7	66	2	26.2	114.0	255	185.0	56.0	4.55	4.2485	92	63
8	60	2	32.1	83.0	179	119.4	42.0	4.00	4.4773	94	110
9	29	1	30.0	85.0	180	93.4	43.0	4.00	5.3845	88	310

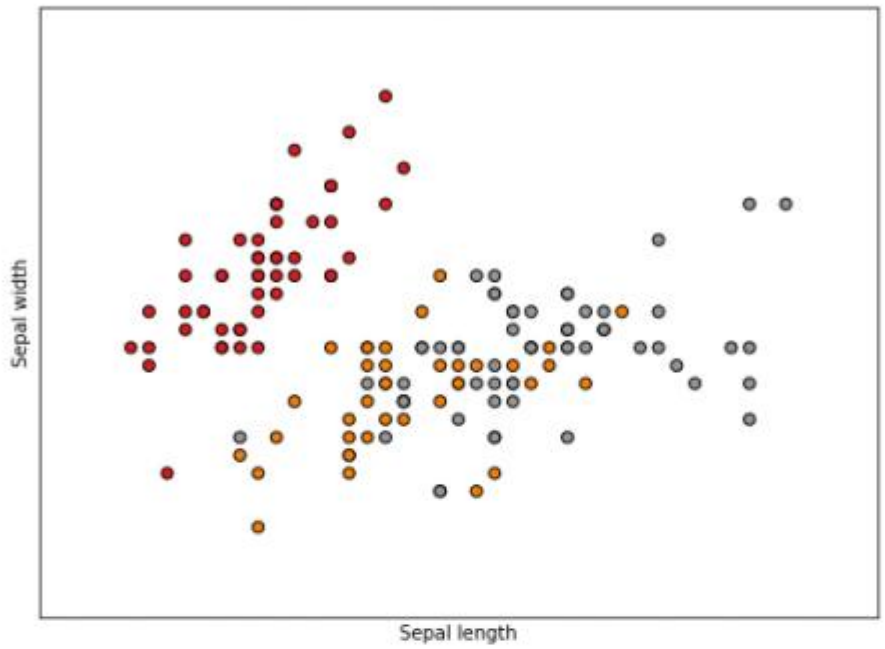
In breast cancer dataset in the class column 2 stands for benign and 4 stands for malignant

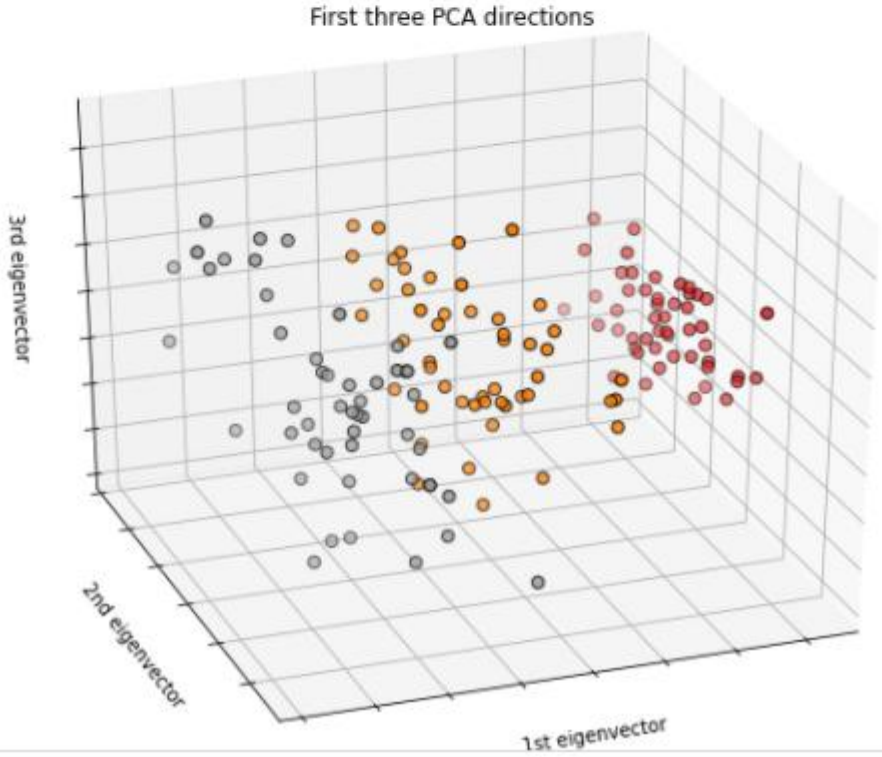
```
[283] breast_cancer.head(10)
```

	sample_number	clump_thickness	cell_size	cell_shape	marginal_adhesion	single_epithelial_cell_size	bare_nuclei	bland_chromatin	normal_nucleoli	mitoses	class	category
0	1000025	5	1	1	1	2	1	3	1	1	2	benign
1	1002945	5	4	4	5	7	10	3	2	1	2	benign
2	1015425	3	1	1	1	2	2	3	1	1	2	benign
3	1016277	6	8	8	1	3	4	3	7	1	2	benign
4	1017023	4	1	1	3	2	1	3	1	1	2	benign
5	1017122	8	10	10	8	7	10	9	7	1	4	malignant
6	1018099	1	1	1	1	2	10	3	1	1	2	benign
7	1018561	2	1	2	1	2	1	3	1	1	2	benign
8	1033078	2	1	1	1	2	1	1	1	5	2	benign
9	1033078	4	2	1	1	2	1	2	1	1	2	benign

Iris dataset analysis:-

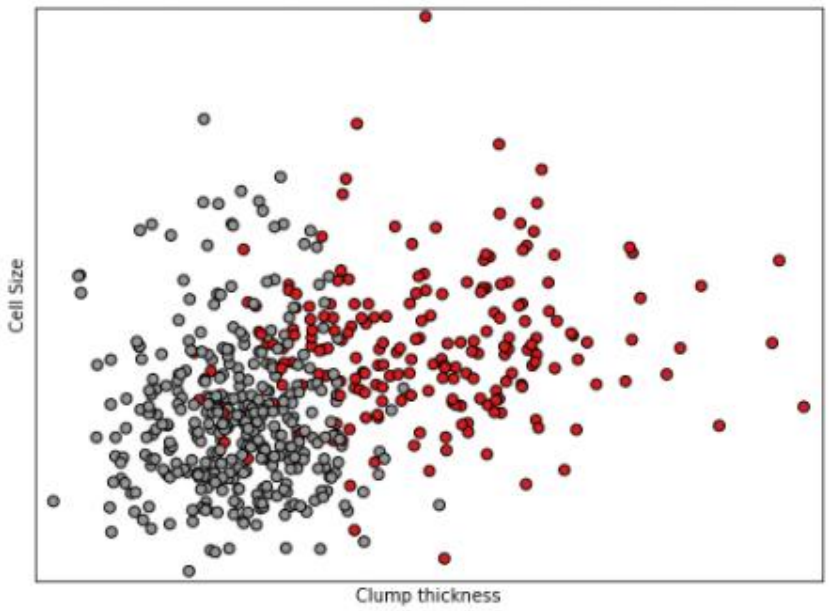
```
ax.w_yaxis.set_ticklabels([])
ax.set_zlabel("3rd eigenvector")
ax.w_zaxis.set_ticklabels([])

plt.show()
```

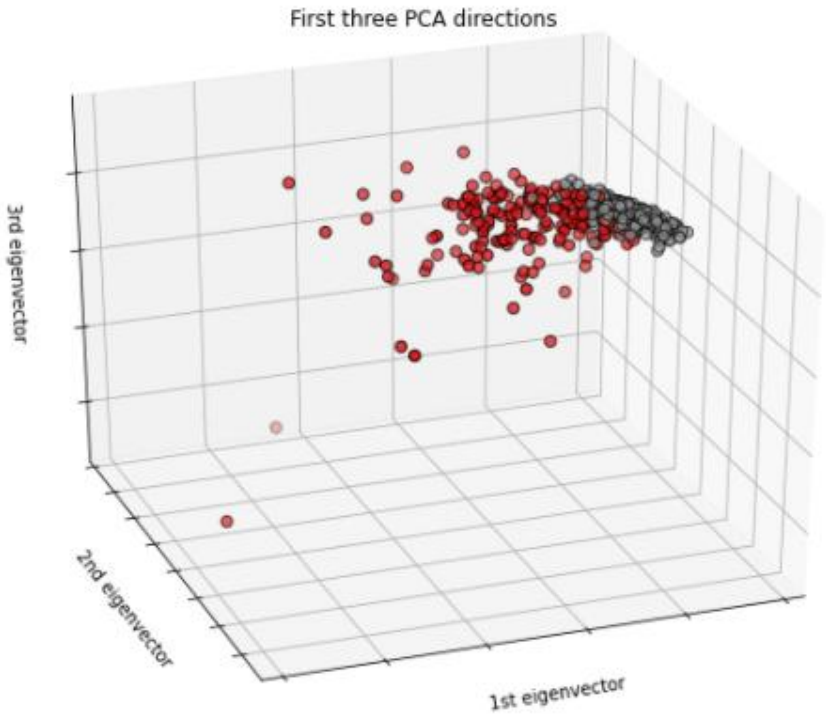




Breast Cancer dataset analysis:-

```
plt.show()
```





Construction of models:-

Without Parameter Tuning Naive Bayes Classification(Gaussian,Multinomial,Bernoulli)

```
[296] iris_gnb = GaussianNB()
      iris_gnb.fit(iris_X_train,iris_y_train)
```

```
GaussianNB(priors=None, var_smoothing=1e-09)
```

```
[297] iris_multi = MultinomialNB()
      iris_multi.fit(iris_X_train,iris_y_train)
```

```
MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
```

```
[298] iris_bern = BernoulliNB()
      iris_bern.fit(iris_X_train,iris_y_train)
```

```
BernoulliNB(alpha=1.0, binarize=0.0, class_prior=None, fit_prior=True)
```

```
[299] iris_gnb_pred = iris_gnb.predict(iris_X_test)
      iris_multinb_pred = iris_multi.predict(iris_X_test)
      iris_bernnb_pred = iris_bern.predict(iris_X_test)
```

With Parameter Tuning gini

```
breast_cancer_clf_tuned = DecisionTreeClassifier(criterion='gini',splitter='random',max_depth=100,min_samples_split=3,min_samples_leaf=2,random_state=0,max_leaf_nodes=100)
breast_cancer_clf_tuned.fit(breast_cancer_X_train,breast_cancer_y_train)
breast_cancer_tune_pred = breast_cancer_clf_tuned.predict(breast_cancer_X_test)
```

Metrics:- Confusion matrix,recall score,f1 score,accuracy,precision score

```
[401] confusion_matrix(breast_cancer_y_test,breast_cancer_tune_pred)
```

```
array([[119,   5],
       [  9,  42]])
```

Error Metrics:-

```
[388] print(f"Micro recall score gini decision non tuned {recall_score(breast_cancer_y_test,breast_cancer_non_tune_pred,average='micro')}")
      print(f"Macro recall score gini decision non tuned {recall_score(breast_cancer_y_test,breast_cancer_non_tune_pred,average='macro')}")
      print(f"Weighted recall score gini decision non tuned {recall_score(breast_cancer_y_test,breast_cancer_non_tune_pred,average='weighted')}")
      print(f"None recall score gini decision non tuned {recall_score(breast_cancer_y_test,breast_cancer_non_tune_pred,average=None)}")

      Micro recall score gini decision non tuned 0.9771428571428571
      Macro recall score gini decision non tuned 0.9665559772296015
      Weighted recall score gini decision non tuned 0.9771428571428571
      None recall score gini decision non tuned [0.99193548 0.94117647]
```

```
[389] print(f"Micro f1 score gini decision non tuned {f1_score(breast_cancer_y_test,breast_cancer_non_tune_pred,average='micro')}")
      print(f"Macro f1 score gini decision non tuned {f1_score(breast_cancer_y_test,breast_cancer_non_tune_pred,average='macro')}")
      print(f"Weighted f1 score gini decision non tuned {f1_score(breast_cancer_y_test,breast_cancer_non_tune_pred,average='weighted')}")
      print(f"None average gini decision non tuned {f1_score(breast_cancer_y_test,breast_cancer_non_tune_pred,average=None)}")

      Micro f1 score gini decision non tuned 0.9771428571428571
      Macro f1 score gini decision non tuned 0.972
      Weighted f1 score gini decision non tuned 0.9770057142857143
      None average gini decision non tuned [0.984 0.96 ]
```

```
[390] print(f"ACCURACY SCORE GINI NON TUNED {accuracy_score(breast_cancer_y_test,breast_cancer_non_tune_pred)}")

      ACCURACY SCORE GINI NON TUNED 0.9771428571428571
```

```
[391] print(f"Micro precision score gini decision non tuned {precision_score(breast_cancer_y_test,breast_cancer_non_tune_pred,average='micro')}")
      print(f"Macro precision score gini decision non tuned {precision_score(breast_cancer_y_test,breast_cancer_non_tune_pred,average='macro')}")
      print(f"Weighted precision score gini decision non tuned {precision_score(breast_cancer_y_test,breast_cancer_non_tune_pred,average='weighted')}")
      print(f"None precision score gini decision non tuned {precision_score(breast_cancer_y_test,breast_cancer_non_tune_pred,average=None)}")

      Micro precision score gini decision non tuned 0.9771428571428571
      Macro precision score gini decision non tuned 0.977891156462585
      Weighted precision score gini decision non tuned 0.977181729834791
      None precision score gini decision non tuned [0.97619048 0.97959184]
```