JOSEPH KINGORI NJIRI

TASK 5 HACKATHON

BREAST CANCER PROGNOSIS USING MACHINE LEARNING SOLUTIONS

Each data has a unique ID as an attribute, the prognosis will give out whether a patient has cancer or not using M(malignant) or B(benign). The data will classify this outcome using the attributes the radius of the infection/lobes, the texture, compactness, area and the perimeter of tumor. The link of the dataset is https://www.kaggle.com/datasets/yasserh/breast-cancer-dataset. I will fine tune the hyper parameters and evaluation metrics of various classifications.

Problem Statement

Breast cancer is the most common cancer amongst women in the world. It accounts for 25% of all cancer cases, and affected over 2.1 Million people in 2015 alone. It starts when cells in the breast begin to grow out of control. These cells usually form tumors that can be seen via X-ray or felt as lumps in the breast area. The key challenges against its detection is how to classify tumors into malignant (cancerous) or benign (non-cancerous). I will use machine learning for analysis and classifying these tumors. Tests such as MRI, mammogram, ultrasound and biopsy are commonly used to diagnose breast cancer performed.

Objectives

1. Identifying the problem and Data Sources
2. Exploratory Data Analysis
3. Pre-Processing the Data
4. Build model to predict whether breast cell tissue is malignant or Benign

Exploratory Data Analysis

colab.research.google.com/drive/1O3dMqWzxXy8VCQkrHnIJG1bSaoPt0h7U

Joseph Njiri Task 2

File Edit View Insert Runtime Tools Help Last edited on November 18

+ Code + Text

```
# importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv('/content/breast-cancer.csv')
df.head(10)
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	radius_worst	te
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	...	25.38	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	...	24.99	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	...	23.57	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	...	14.91	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	...	22.54	
5	843786	M	12.45	15.70	82.57	477.1	0.12780	0.17000	0.15780	0.08089	...	15.47	
6	844359	M	18.25	19.98	119.60	1040.0	0.09463	0.10900	0.11270	0.07400	...	22.88	

Type here to search

6:50 PM 11/22/2022

colab.research.google.com/drive/1O3dMqWzxXy8VCQkrHnIJG1bSaoPt0h7U

Joseph Njiri Task 2

File Edit View Insert Runtime Tools Help Last edited on November 18

+ Code + Text

```
df.dtypes
```

```
id                        int64
diagnosis                object
radius_mean            float64
texture_mean           float64
perimeter_mean         float64
area_mean              float64
smoothness_mean        float64
compactness_mean       float64
concavity_mean         float64
concave points_mean    float64
symmetry_mean          float64
fractal_dimension_mean float64
radius_se              float64
texture_se             float64
perimeter_se           float64
area_se                float64
smoothness_se          float64
compactness_se         float64
concavity_se           float64
concave points_se      float64
symmetry_se            float64
fractal_dimension_se   float64
radius_worst           float64
```

Type here to search

6:50 PM 11/22/2022

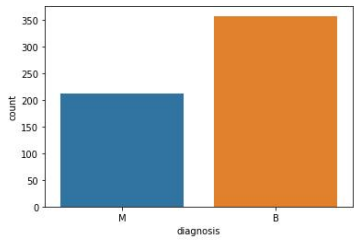
Joseph Njiri Task 2

File Edit View Insert Runtime Tools Help Last edited on November 18

```
[ ]        'compactness_worst', 'concavity_worst', 'concave points_worst',
           'symmetry_worst', 'fractal_dimension_worst'],
          dtype='object')
```

```
[ ] df.drop('id',inplace  =True,axis = 1)
```

```
sns.countplot(data=df,x='diagnosis')
plt.savefig('targetdiagnosis')
```




```
[ ] data=df.describe().T
```

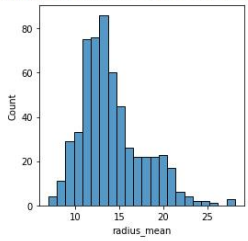
Joseph Njiri Task 2

File Edit View Insert Runtime Tools Help Last edited on November 18

```
[ ] perimeter_worst          84.110000   97.660000   125.400000   251.20000
    area_worst              515.300000  686.500000  1084.000000  4254.00000
    smoothness_worst          0.116600    0.131300     0.146000     0.22260
```

```
plt.figure(figsize=(4,4))
sns.histplot(data = df,x = 'radius_mean')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f59d75c5590>
```



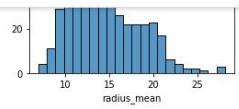

```
[ ] plt.figure(figsize=(4,4))
    sns.boxplot(data = df,x = 'texture_mean')
```

colab.research.google.com/drive/1O3dMqWzxXy8VCQkrHnIJG1bSaoPt0h7U

Joseph Njiri Task 2

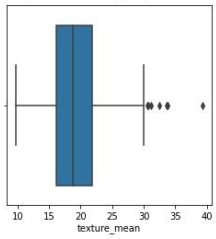
File Edit View Insert Runtime Tools Help Last edited on November 18

+ Code + Text




```
plt.figure(figsize=(4,4))
sns.boxplot(data = df,x = 'texture_mean')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f59d7538c90>
```



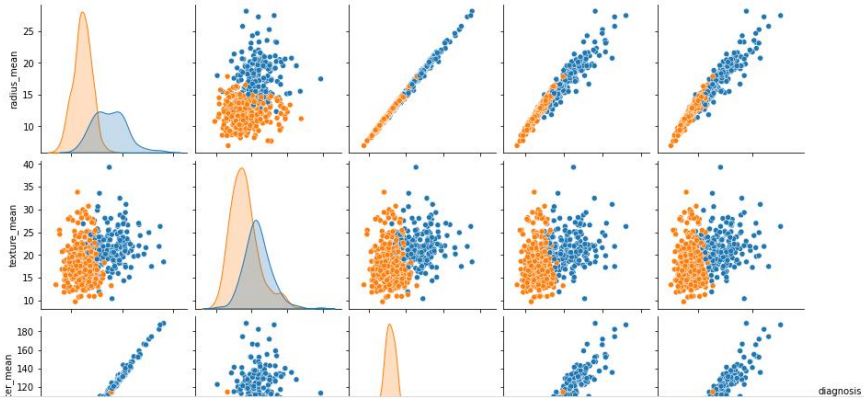
colab.research.google.com/drive/1O3dMqWzxXy8VCQkrHnIJG1bSaoPt0h7U

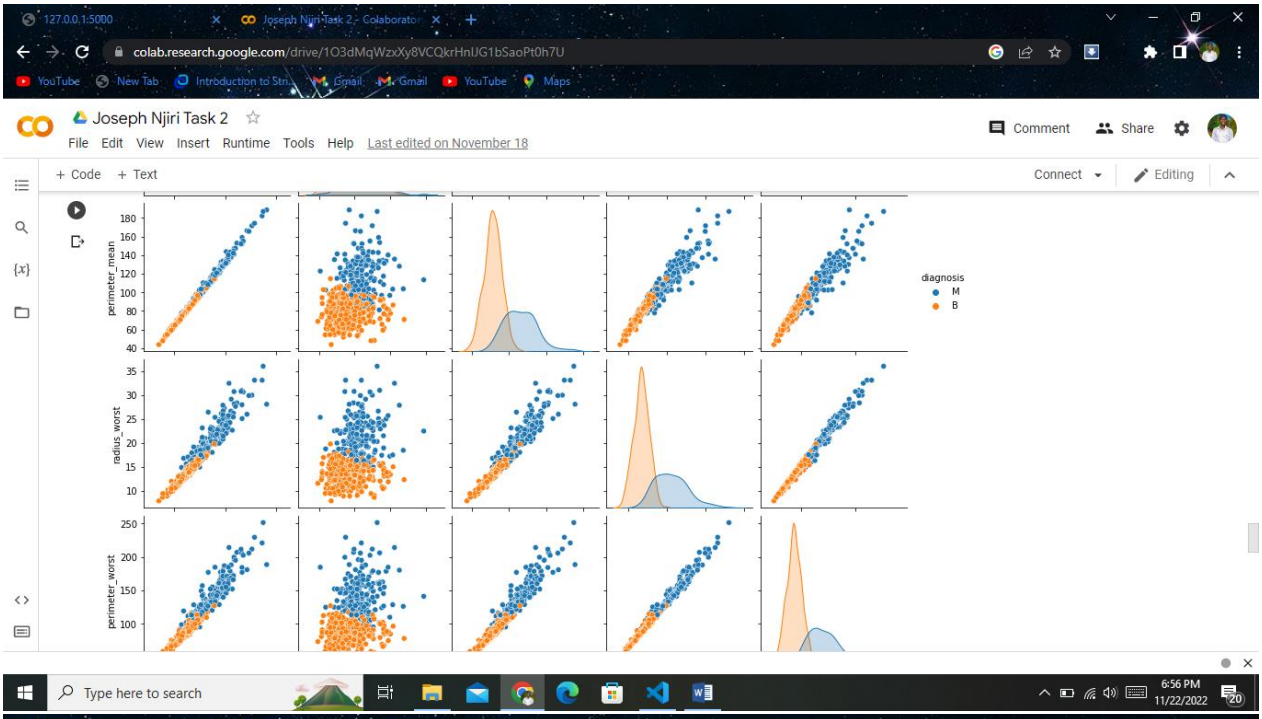
Joseph Njiri Task 2

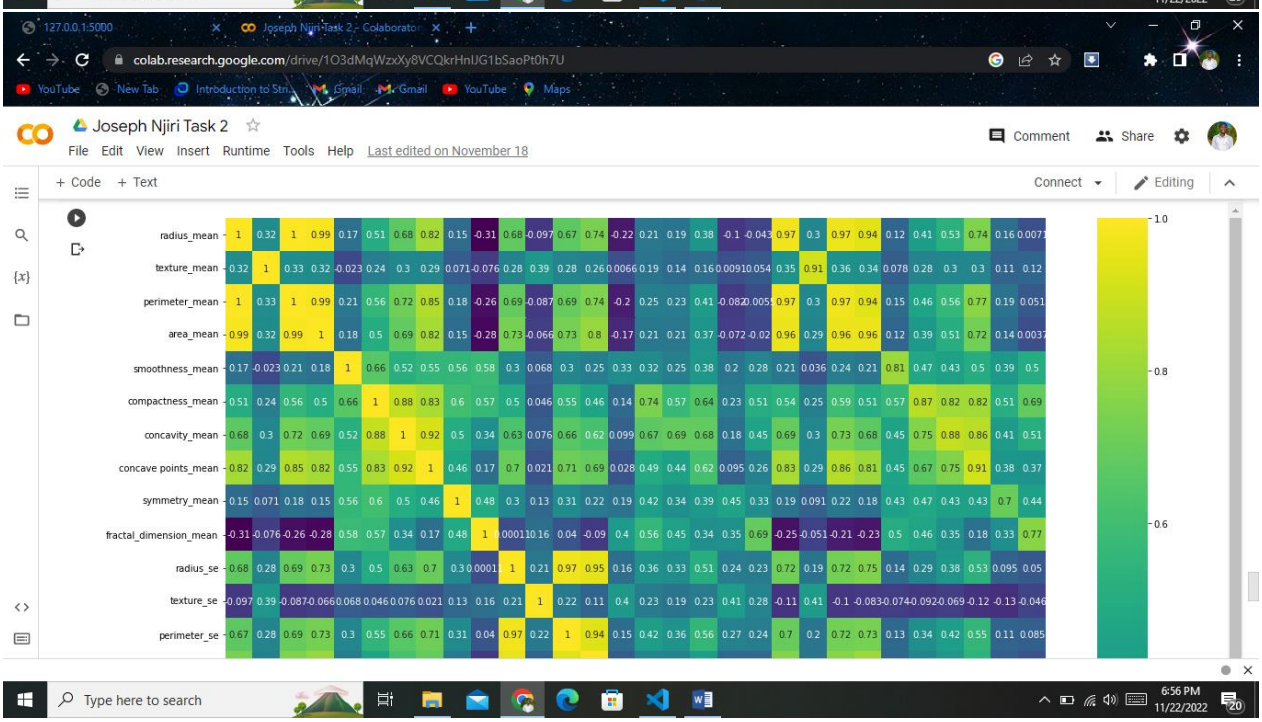
File Edit View Insert Runtime Tools Help Last edited on November 18

+ Code + Text

```
sns.pairplot(df, hue="diagnosis", vars=["radius_mean", "texture_mean", "perimeter_mean", "radius_worst", "perimeter_worst"])
plt.show()
```

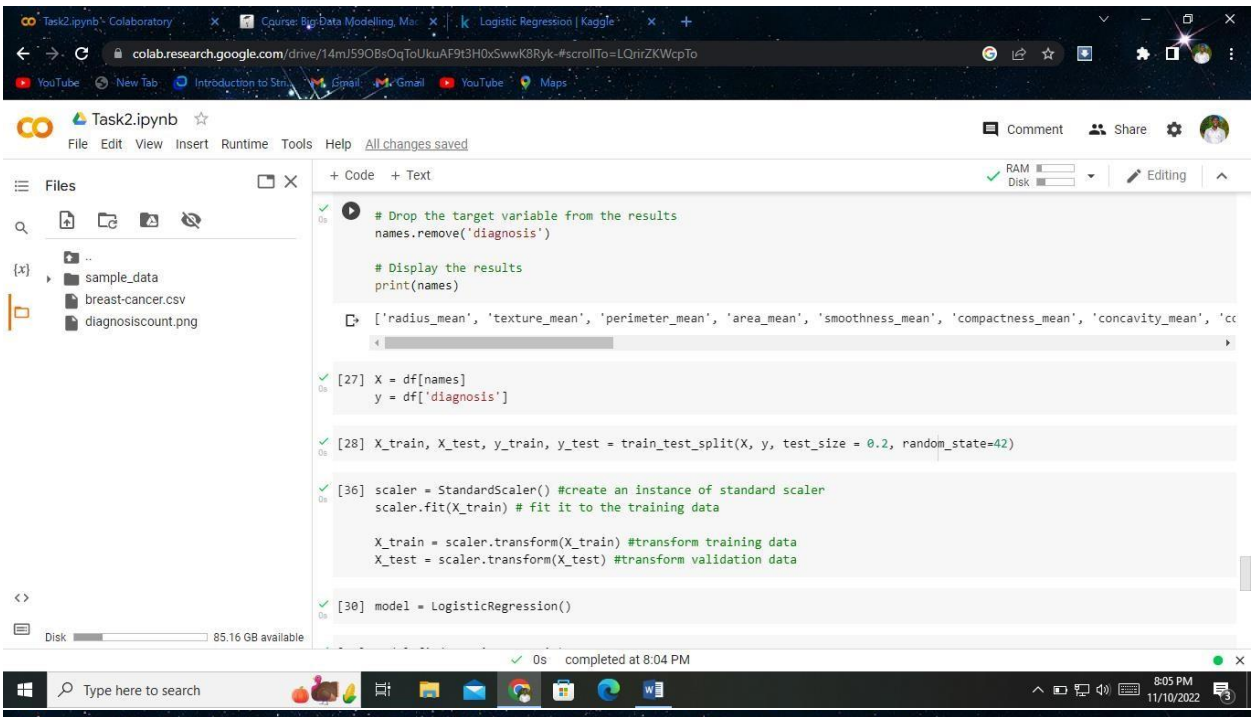


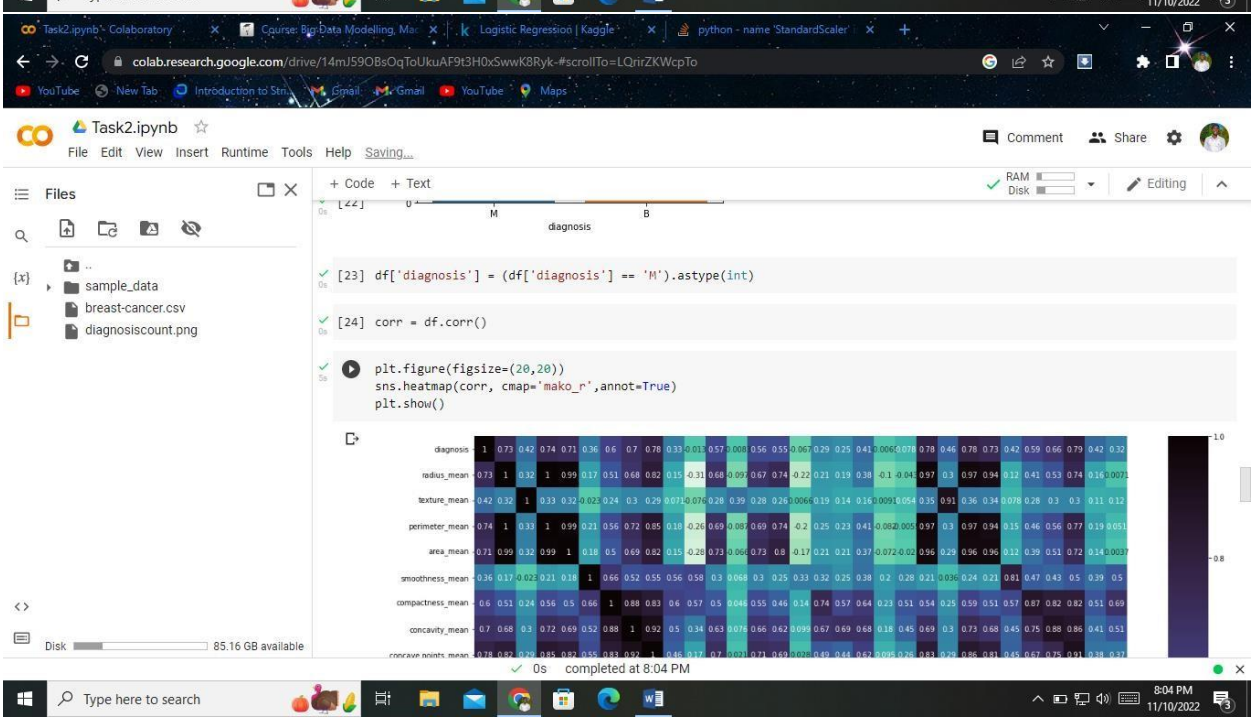

127.0.0.1:5000
Joseph Njiri Task 2 - Colaborator
colab.research.google.com/drive/1O3dMqWzxXy8VCQkrHnIJG1bSaoPt0h7U
YouTube
New Tab
Introduction to Stri
Gmail
Gmail
YouTube
Maps
Joseph Njiri Task 2
File Edit View Insert Runtime Tools Help Last edited on November 18
Comment
Share
+ Code + Text
Connect
Editing
perimeter_mean
radius_worst
perimeter_worst
diagnosis
M
B
Type here to search
6:56 PM
11/22/2022

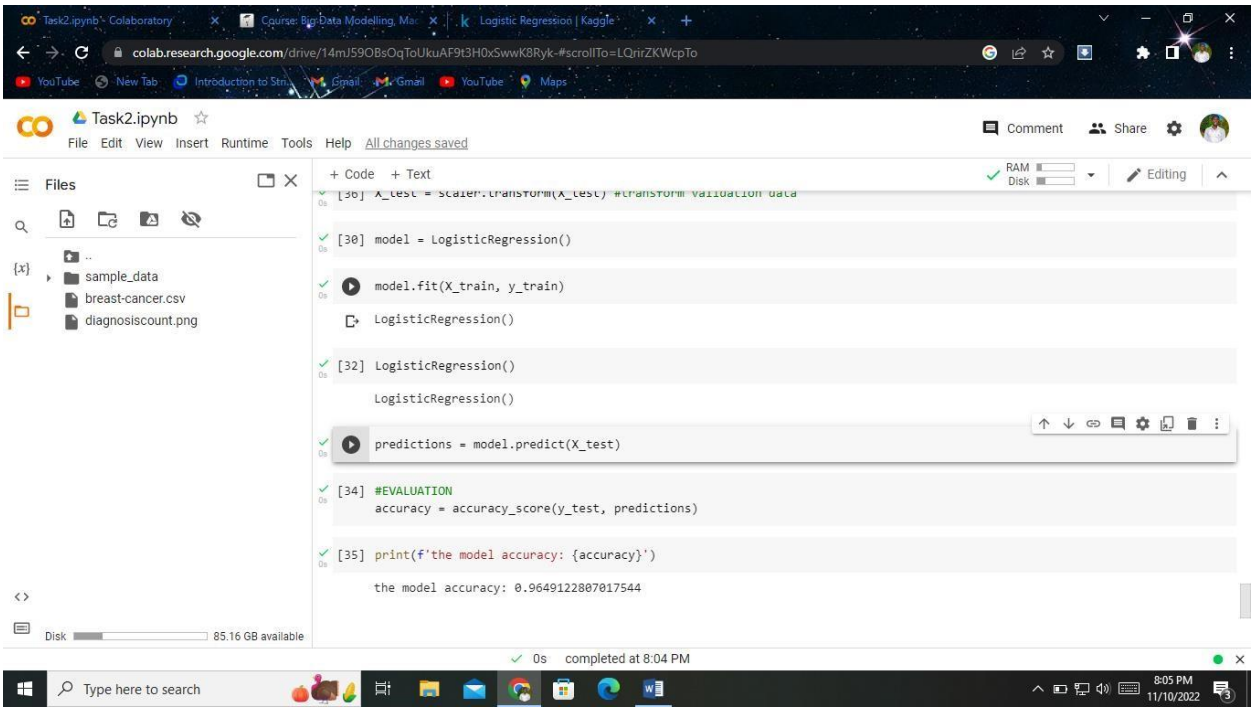

127.0.0.1:5000
Joseph Njiri Task 2 - Colaborator
colab.research.google.com/drive/1O3dMqWzxXy8VCQkrHnIJG1bSaoPt0h7U
YouTube
New Tab
Introduction to Stri
Gmail
Gmail
YouTube
Maps
Joseph Njiri Task 2
File Edit View Insert Runtime Tools Help Last edited on November 18
Comment
Share
+ Code + Text
Connect
Editing
radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
radius_se
texture_se
perimeter_se
1.0
0.8
0.6
Type here to search
6:56 PM
11/22/2022


127.0.0.1:5000
Joseph Njiri Task 2 - Colaborator
colab.research.google.com/drive/1O3dMqWzxXy8VCQkrHnIJG1bSaoPt0h7U
Joseph Njiri Task 2
File Edit View Insert Runtime Tools Help Last edited on November 18
Comment
Share
+ Code + Text
Connect
Editing
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst
radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst
0.0
−0.2
Type here to search
6:56 PM
11/22/2022

Logistic Regression







SUPPORT VECTOR MACHINES(SVM)

(SVM) support vector machine can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well in this case the benign and the malignant type of cancer it gave the highest accuracy of 0.98/98% thus it's the best machine learning solution for the breast cancer dataset. Support Vectors are simply the coordinates of individual observation. The SVM classifier is a frontier that best segregates the two classes (hyperplane/ line).

Task2.ipynb

```
[42] pred = model.predict(X_test)
```

```
[43] from sklearn.metrics import classification_report,confusion_matrix
```

```
print(classification_report(y_test,pred))
print('\n')

print(confusion_matrix(y_test,pred))
```

```
              precision    recall  f1-score   support

           0       0.97      0.99      0.98        71
           1       0.98      0.95      0.96        43

    accuracy                           0.97       114
   macro avg       0.97      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114


[[70  1]
 [ 2 41]]
```

```
[45] from sklearn.model_selection import GridSearchCV
```

Task2.ipynb

```
[ 2 41]]
```

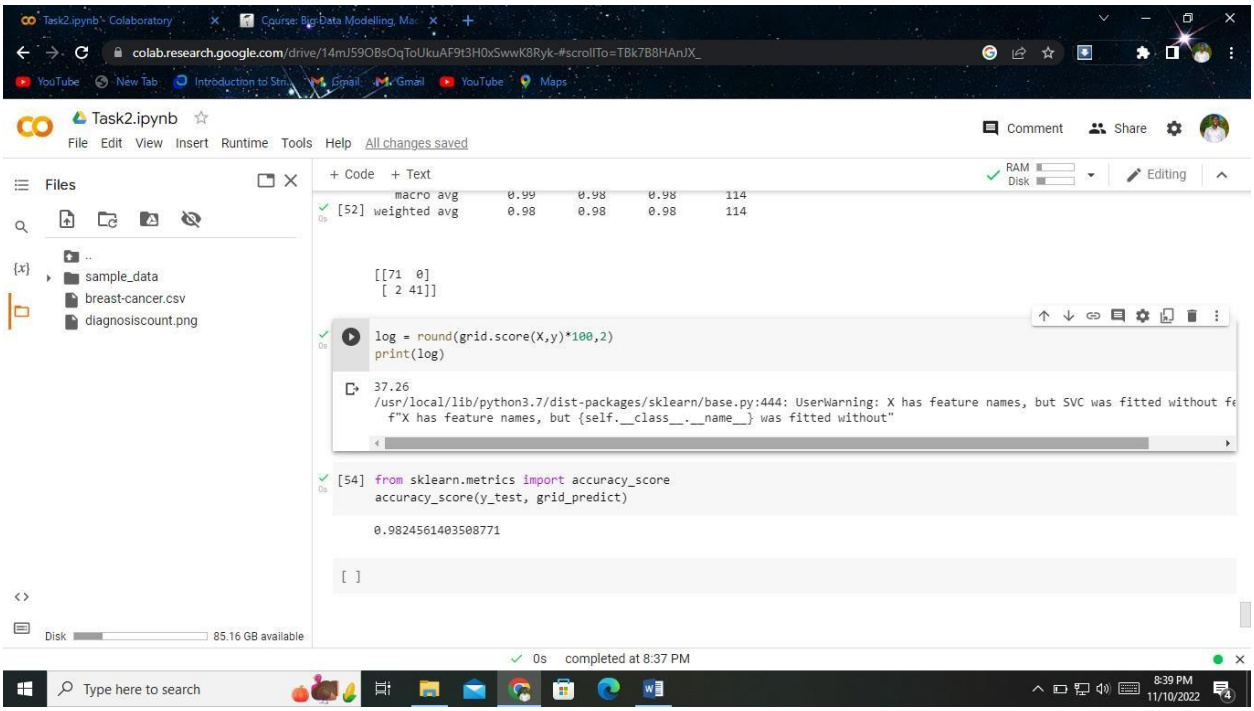
```
[45] from sklearn.model_selection import GridSearchCV
```

```
[46] param_grid = {'C':[0.1,1.1,10,100,1000],'gamma':[1,0.1,0.01,0.001,0.0001]}
```

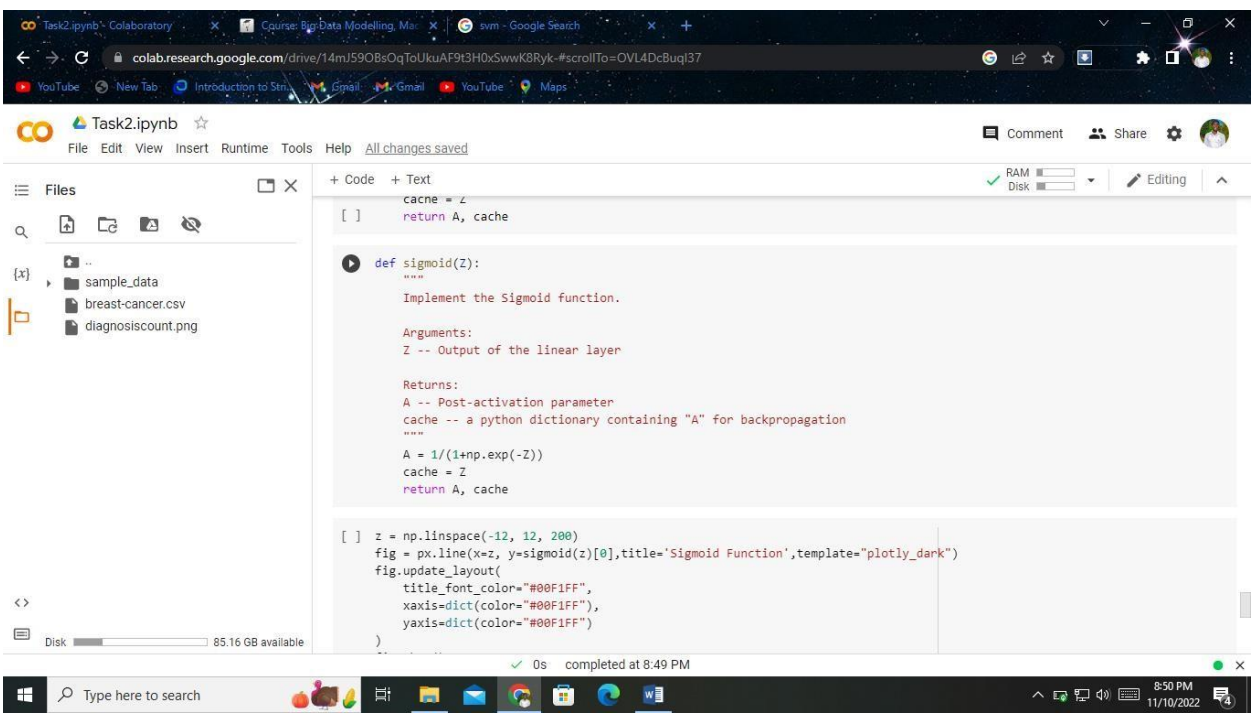
```
[47] grid = GridSearchCV(SVC(),param_grid,verbose=3)
```

```
[48] grid.fit(X_train,y_train)
```

```
Fitting 5 folds for each of 25 candidates, totalling 125 fits
[CV 1/5] END ....................C=0.1, gamma=1;, score=0.637 total time=   0.0s
[CV 2/5] END ....................C=0.1, gamma=1;, score=0.626 total time=   0.0s
[CV 3/5] END ....................C=0.1, gamma=1;, score=0.626 total time=   0.0s
[CV 4/5] END ....................C=0.1, gamma=1;, score=0.626 total time=   0.0s
[CV 5/5] END ....................C=0.1, gamma=1;, score=0.626 total time=   0.0s
[CV 1/5] END .................C=0.1, gamma=0.1;, score=0.912 total time=   0.0s
[CV 2/5] END .................C=0.1, gamma=0.1;, score=0.967 total time=   0.0s
[CV 3/5] END .................C=0.1, gamma=0.1;, score=0.934 total time=   0.0s
[CV 4/5] END .................C=0.1, gamma=0.1;, score=0.934 total time=   0.0s
[CV 5/5] END .................C=0.1, gamma=0.1;, score=0.912 total time=   0.0s
[CV 1/5] END ................C=0.1, gamma=0.01;, score=0.934 total time=   0.0s
[CV 2/5] END ................C=0.1, gamma=0.01;, score=0.923 total time=   0.0s
[CV 3/5] END ................C=0.1, gamma=0.01;, score=0.956 total time=   0.0s
```

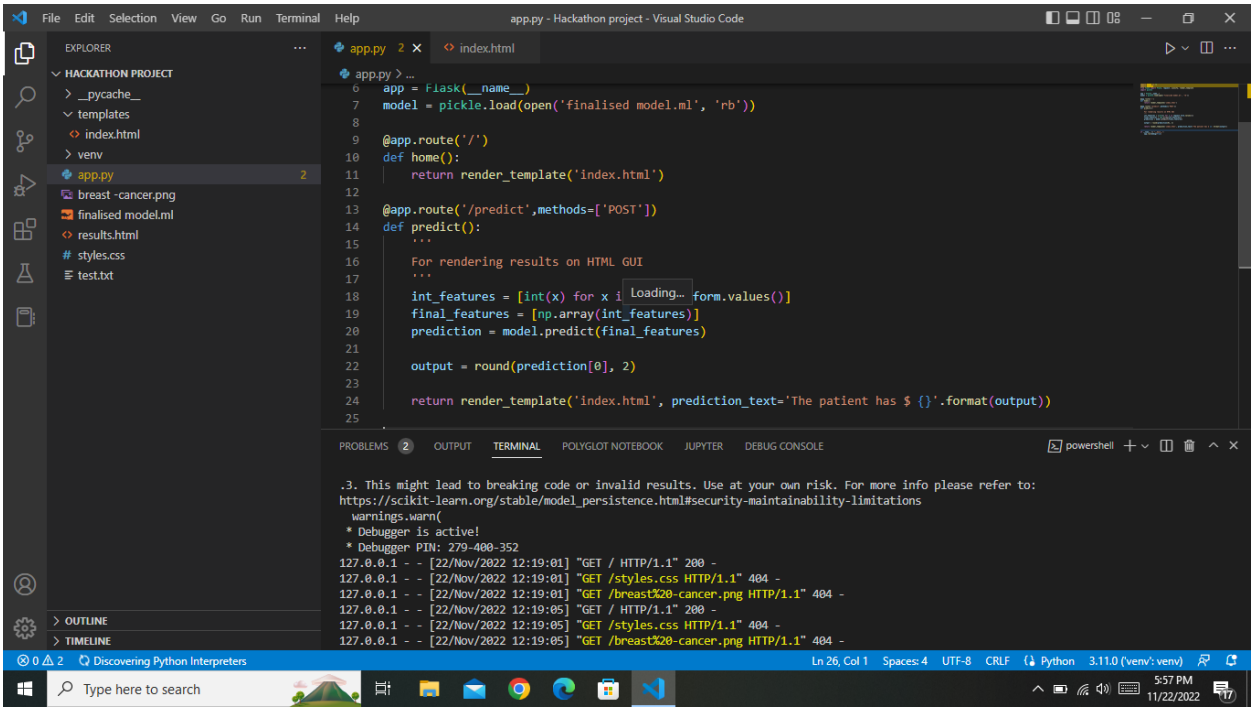


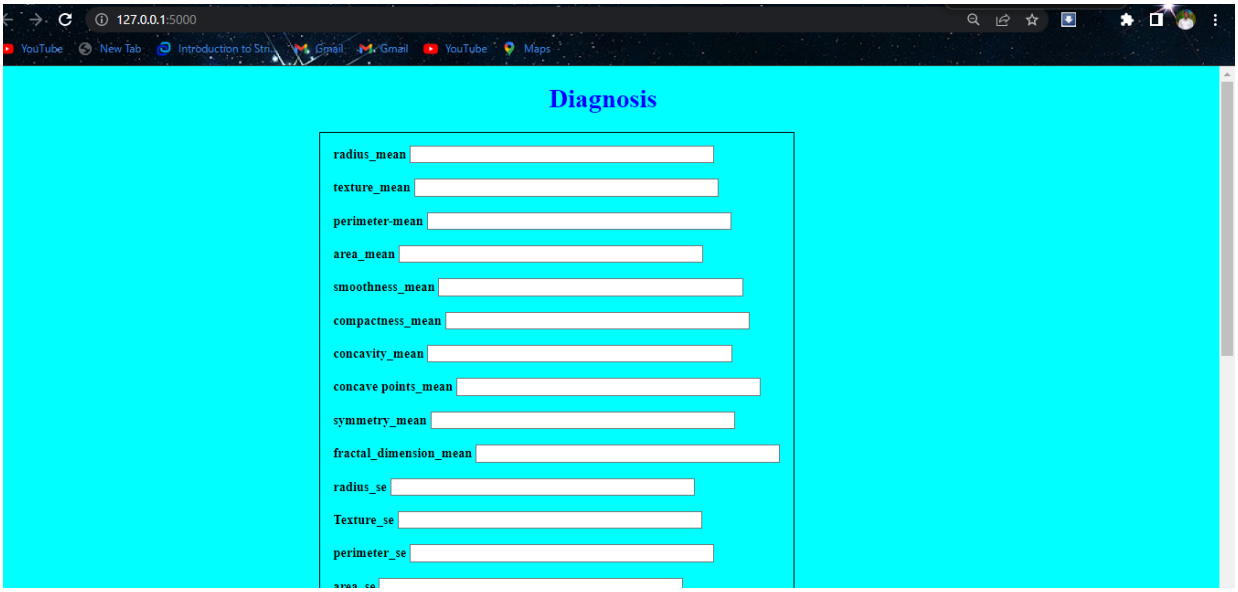
NEURAL NETWORKS

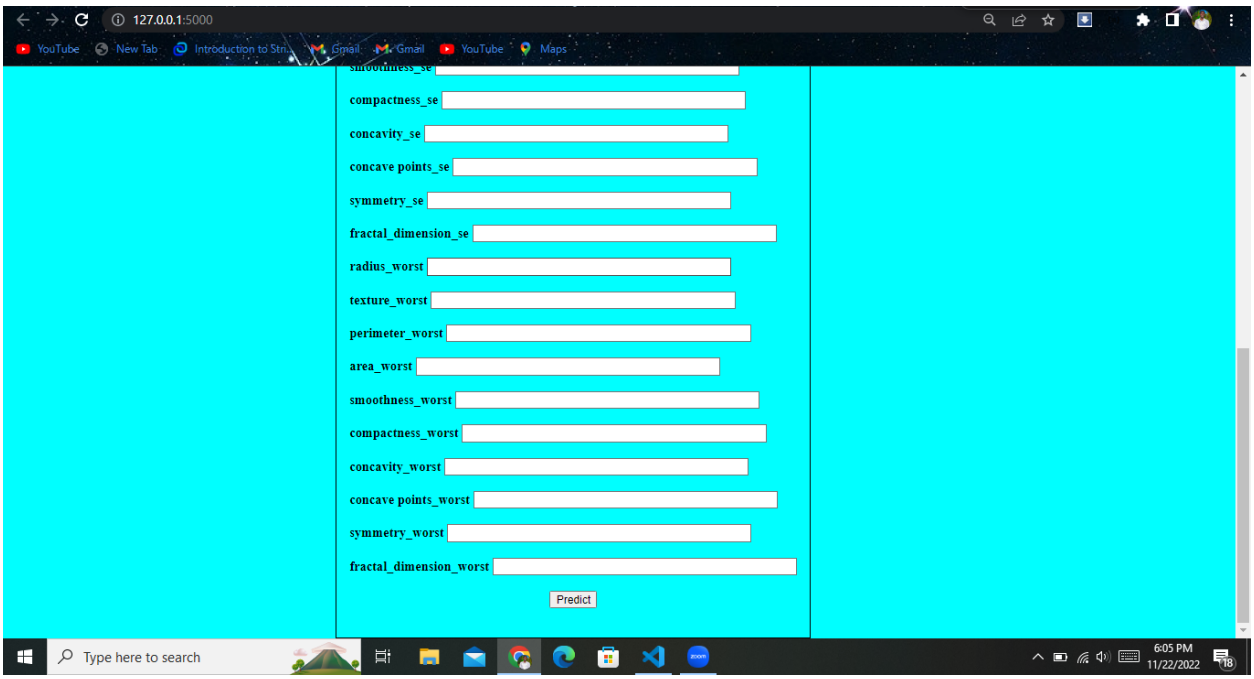


The above machine learning technique gives an accuracy of 0.96. It evaluates the model using the sigmoid function and creating a class neural networks. The process of model evaluation was too long but it gave a lower accuracy compared to support vector machine algorithm svm.

Model Deployment







Given breast cancer results from breast fine needle aspiration (FNA) test (is a quick and simple procedure to perform, which removes some fluid or cells from a breast lesion or cyst (a lump, sore or swelling) with a fine needle similar to a blood sample needle). Since this build a model that can classify a breast cancer tumor using two training classification:

1= Malignant (Cancerous) - Present

0= Benign (Not Cancerous) -Absent

After inputing different types of data a diagnosis will be given in form of text showing whether the cancer is malignant(M) or benign(B).