BVRIT HYDERABAD College of Engineering for Women

Department of Information Technology

BREAST CANCER PROGNOSIS USING MACHINE LEARNING

Under the guidance of

Guide Name: Mr. B.Srinivasulu

Designation: Assistant Professor

Team- 09

R. Parimala (19WH1A1268)

P. Aditi Kiran(19WH1A1277)

Ch. Lakshmi Durga(19WH1A1295)

Raveena Yadlapalli (19WH1A1296)

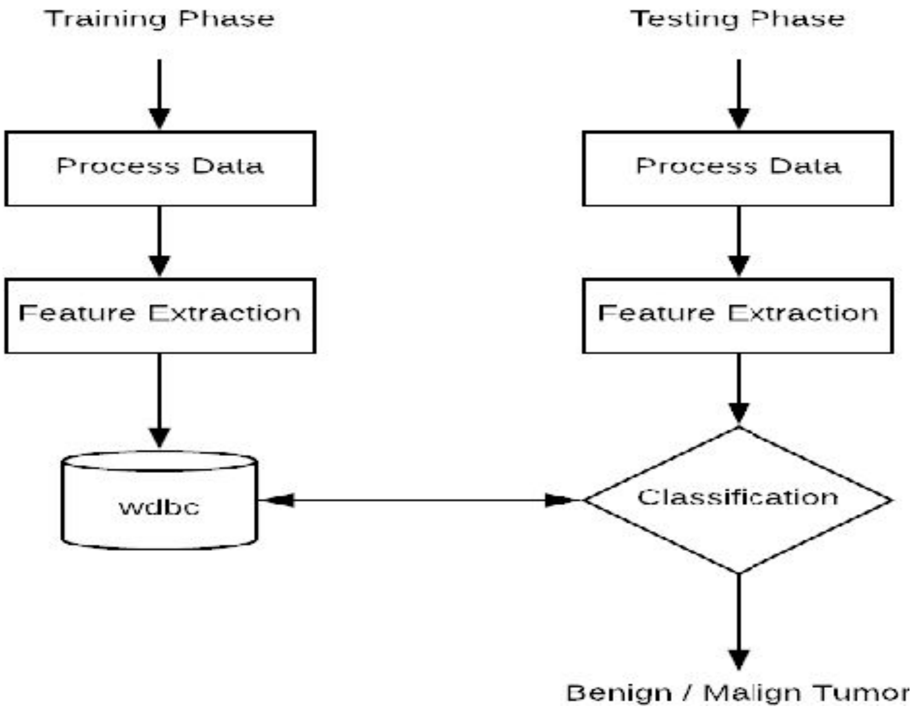
Agenda

- Summary of Stage 1
- Architecture
- Modules
- Implementation of Modules
- Conclusion
- References

Summary of Stage 1

On the Wisconsin Breast Cancer Diagnostic dataset (WBCD) we applied five main algorithms which are: SVM, Random Forests, Logistic Regression, Decision Tree, K -NN, to calculate, compare and evaluate different results obtained based on confusion matrix, accuracy, sensitivity, precision, AUC to identify the best machine learning algorithm that are precise, reliable and find the higher accuracy. All algorithms have been programmed in Python using scikit-learn library in Jupyter notebook environment. After an accurate comparison between our models, we found that Support Vector Machine achieved a higher efficiency of 97.2%, Precision of 97.5%, AUC of 96.6% and outperforms all other algorithms.

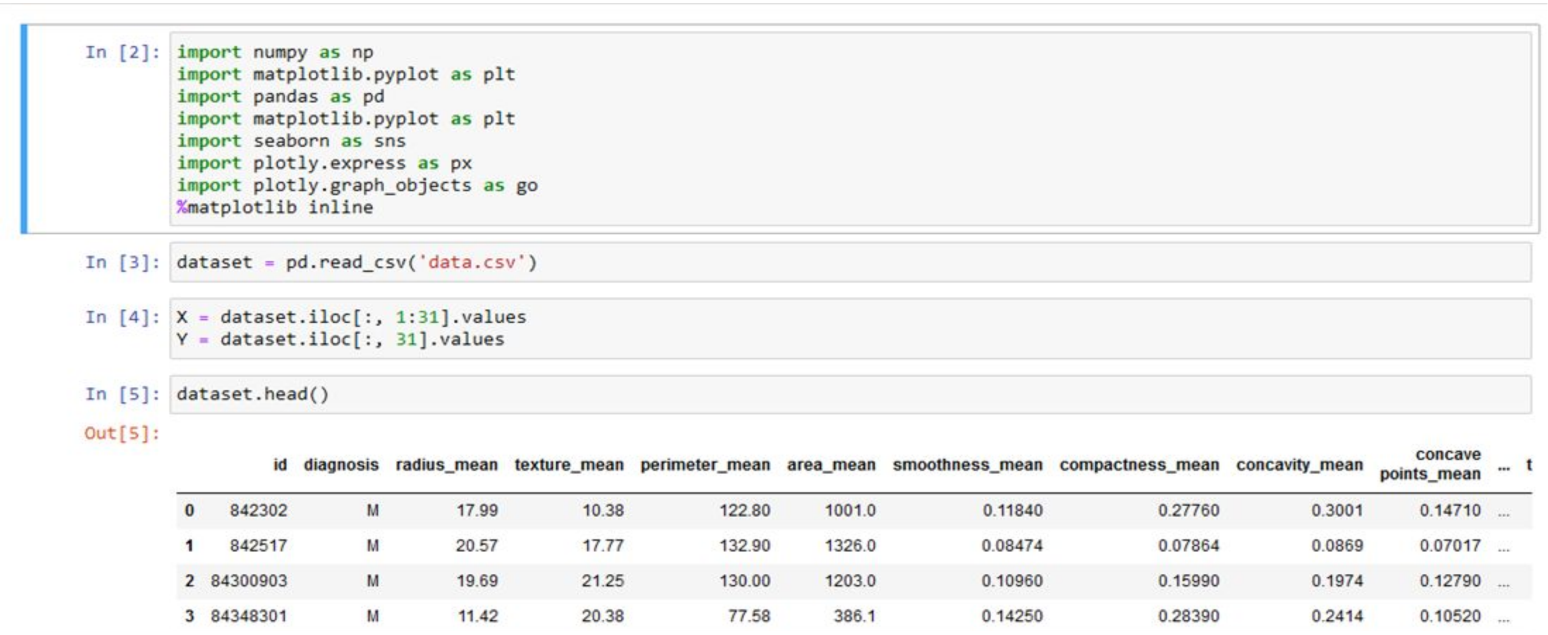
Architecture



Modules

- Importing of data.
- Prediction of Breast cancer
- Prediction of Reoccurence of Breast cancer.

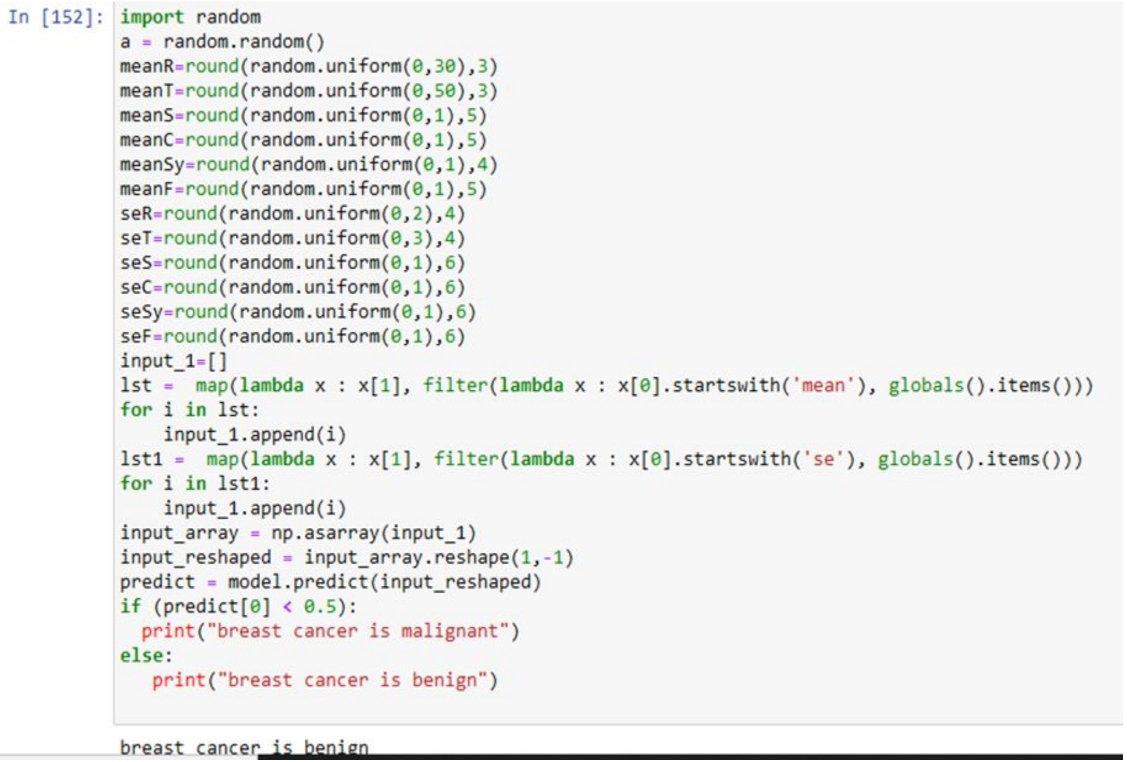
Implementation of Modules



```
In [2]: import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        import plotly.express as px
        import plotly.graph_objects as go
        %matplotlib inline

In [3]: dataset = pd.read_csv('data.csv')

In [4]: X = dataset.iloc[:, 1:31].values
        Y = dataset.iloc[:, 31].values

In [5]: dataset.head()

Out[5]:
```

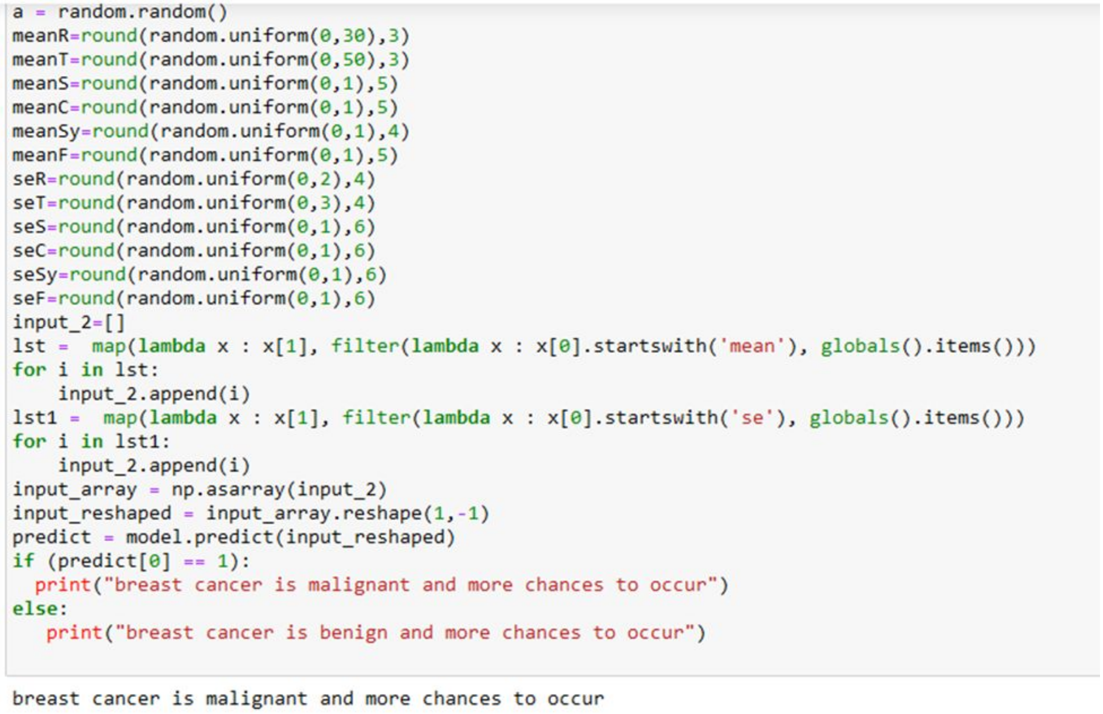
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...	t
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	...	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	...	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	...	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	...	

Implementation of Modules



```
In [152]: import random
          a = random.random()
          meanR=round(random.uniform(0,30),3)
          meanT=round(random.uniform(0,50),3)
          meanS=round(random.uniform(0,1),5)
          meanC=round(random.uniform(0,1),5)
          meanSy=round(random.uniform(0,1),4)
          meanF=round(random.uniform(0,1),5)
          seR=round(random.uniform(0,2),4)
          seT=round(random.uniform(0,3),4)
          seS=round(random.uniform(0,1),6)
          seC=round(random.uniform(0,1),6)
          seSy=round(random.uniform(0,1),6)
          seF=round(random.uniform(0,1),6)
          input_1=[]
          lst =  map(lambda x : x[1], filter(lambda x : x[0].startswith('mean'), globals().items()))
          for i in lst:
              input_1.append(i)
          lst1 =  map(lambda x : x[1], filter(lambda x : x[0].startswith('se'), globals().items()))
          for i in lst1:
              input_1.append(i)
          input_array = np.asarray(input_1)
          input_reshaped = input_array.reshape(1,-1)
          predict = model.predict(input_reshaped)
          if (predict[0] < 0.5):
            print("breast cancer is malignant")
          else:
             print("breast cancer is benign")

          breast cancer is benign
```

Implementation of Modules



```
a = random.random()
meanR=round(random.uniform(0,30),3)
meanT=round(random.uniform(0,50),3)
meanS=round(random.uniform(0,1),5)
meanC=round(random.uniform(0,1),5)
meanSy=round(random.uniform(0,1),4)
meanF=round(random.uniform(0,1),5)
seR=round(random.uniform(0,2),4)
seT=round(random.uniform(0,3),4)
seS=round(random.uniform(0,1),6)
seC=round(random.uniform(0,1),6)
seSy=round(random.uniform(0,1),6)
seF=round(random.uniform(0,1),6)
input_2=[]
lst =  map(lambda x : x[1], filter(lambda x : x[0].startswith('mean'), globals().items()))
for i in lst:
    input_2.append(i)
lst1 =  map(lambda x : x[1], filter(lambda x : x[0].startswith('se'), globals().items()))
for i in lst1:
    input_2.append(i)
input_array = np.asarray(input_2)
input_reshaped = input_array.reshape(1,-1)
predict = model.predict(input_reshaped)
if (predict[0] == 1):
  print("breast cancer is malignant and more chances to occur")
else:
   print("breast cancer is benign and more chances to occur")
```

```
breast cancer is malignant and more chances to occur
```

Prediction of Reoccurence module

The recurrence is predicted on dataset based on diagnosis (malignant) and tumour size.

Conclusion

- In this project we concluded that SVM is the best machine learning algorithm to predict breast cancer.
- We are working on improving the accuracy and integrating all modules and publish the IEEE journal within a period of one month.

References

[1] Usman Naseem,Junaid Rashid, ”An Automatic Detection of Breast Cancer Diagnosis and Prognosis Based on Machine Learning Using Ensemble of Classifiers”12 May 2022

[2] Sharma, A. & Mishra, P. K. Performance analysis of machine learning based optimized feature selection approaches for breast cancer diagnosis. *Int. J. Inf. Tecnol.* 14, 1949–1960. (2022).

[3] Ahmad, S. *et al.* A novel hybrid deep learning model for metastatic cancer detection. *Comput. Intell. Neurosci* (2022).

[4]L Yang, B Fu, Y Li, Y Liu, W Huang, S Feng et al., "Prediction model of the response to neoadjuvant chemotherapy in cancers by a Naive Bayes algorithm", Computer methods and programs in biomedicine, vol. 192, pp. 105458, 2020.

THANK YOU