"Predictive Modeling for Breast Cancer Detection:

A Comprehensive Report"

Prabhanshu Singh

21/05/2023

Abstract

This model presents an in-depth analysis of various predictive modeling techniques for breast cancer detection. This report examines the importance of early detection, explores different data-driven approaches, and evaluates the performance of machine learning algorithms in predicting breast cancer outcomes.

By leveraging large-scale datasets and advanced analytics, this report aims to enhance the accuracy and efficiency of breast cancer prediction models. The findings provide valuable insights for healthcare professionals, researchers, and policymakers, facilitating improved decision-making and ultimately contributing to better patient outcomes in the battle against breast cancer.

1.Problem Statement

The challenge lies in accurately predicting breast cancer outcomes using advanced predictive modeling techniques to aid in early detection and improve patient outcomes.

2.Market/Customer Need Assessment

Breast cancer is a significant global health concern, and early detection plays a crucial role in improving survival rates and treatment outcomes. There is a growing need for accurate and reliable breast cancer prediction tools that can assist healthcare professionals in identifying high-risk individuals and initiating timely interventions.

Current screening methods have limitations, and there is a demand for more sophisticated predictive models that can utilize diverse patient data to provide personalized risk assessments. By addressing this need, healthcare providers can enhance patient care, optimize resource allocation, and ultimately contribute to reducing the burden of breast cancer on individuals and society.

3. Target Specification and characterization

Target Specification: The target of this breast cancer prediction product is to develop a robust and accurate predictive model that can identify individuals at high risk of developing breast cancer. The model aims to utilize various patient characteristics, including demographic information, medical history, genetic factors, and imaging data, to generate personalized risk assessments. By accurately identifying high-risk individuals, healthcare providers can offer targeted screening and preventive interventions, ultimately improving patient outcomes and reducing the overall burden of breast cancer.

Target Characterization: The target population for this breast cancer prediction report includes individuals of varying age groups, genders, and ethnic backgrounds who are at risk of developing breast cancer. The model aims to identify those with an increased likelihood of developing the disease based on specific risk factors and characteristics. Through comprehensive data analysis and machine learning techniques, the model seeks to provide personalized risk assessments that can guide healthcare professionals in making informed decisions regarding screening, prevention, and treatment strategies tailored to the individual needs of each patient.

4.External Search (Information Sources)

For this breast cancer prediction product, external sources such as scientific research articles, medical journals, and reputable healthcare websites will be consulted to gather relevant information on breast cancer risk factors, predictive models, and data analysis techniques.

Additionally, the dataset obtained from Kaggle will serve as a valuable source of information for understanding the variables and features used in the prediction model development process.

The sources of subsequent information is given below as reference.

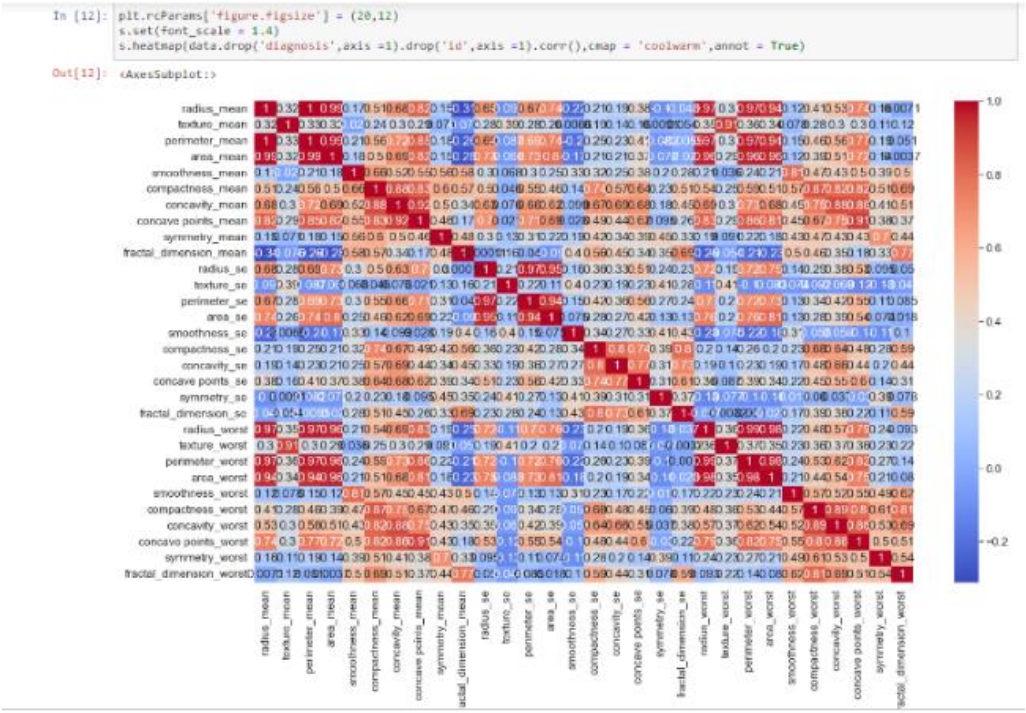
```
In [1]: import os
        os.chdir('C:\\Users\\prabh\\OneDrive\\Desktop\\Machine Learning With Python\\PROJECTS\\PRJ Cancer Prediction')
        os.getcwd()

Out[1]: 'C:\\Users\\prabh\\OneDrive\\Desktop\\Machine Learning With Python\\PROJECTS\\PRJ Cancer Prediction'

In [2]: import numpy as np
        import pandas as pd

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

Out[3]:

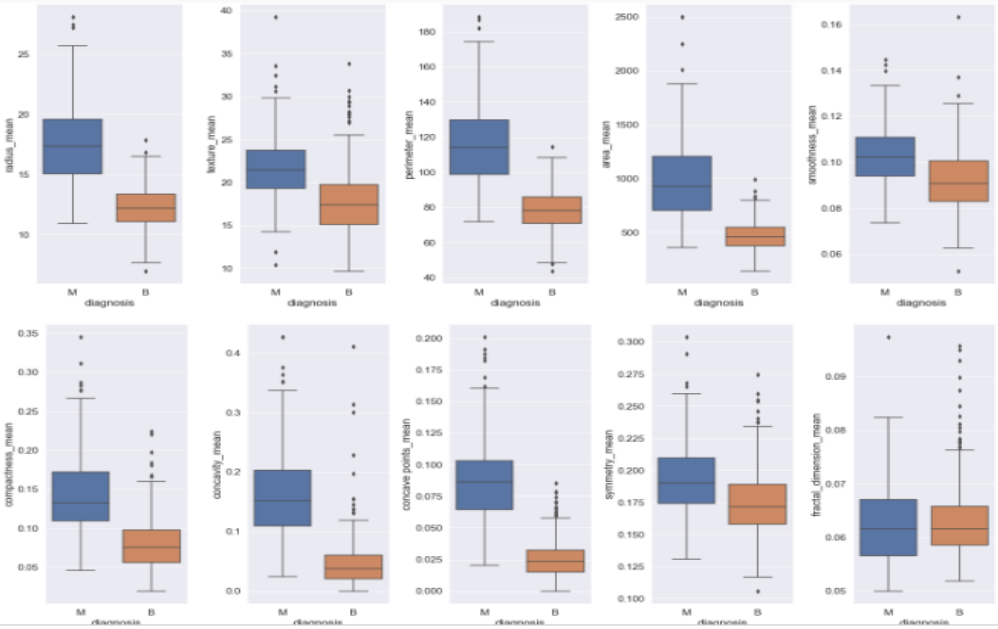
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	...
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	...
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	...
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	...
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	...
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	...
...	...	...	...	...	...	...	...	...	...	...	...
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	...
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	...

5. Benchmarking

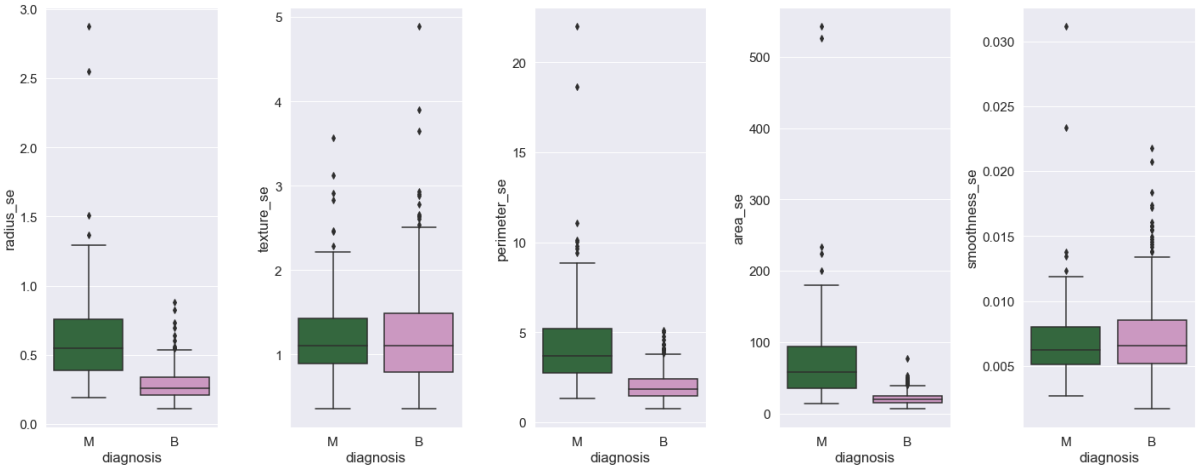
```
In [12]: plt.rcParams['figure.figsize'] = (20,12)
         s.set(font_scale = 1.4)
         s.heatmap(data.drop('diagnosis',axis =1).drop('id',axis =1).corr(),cmap = 'coolwarm',annot = True)

Out[12]: <AxesSubplot:>
```

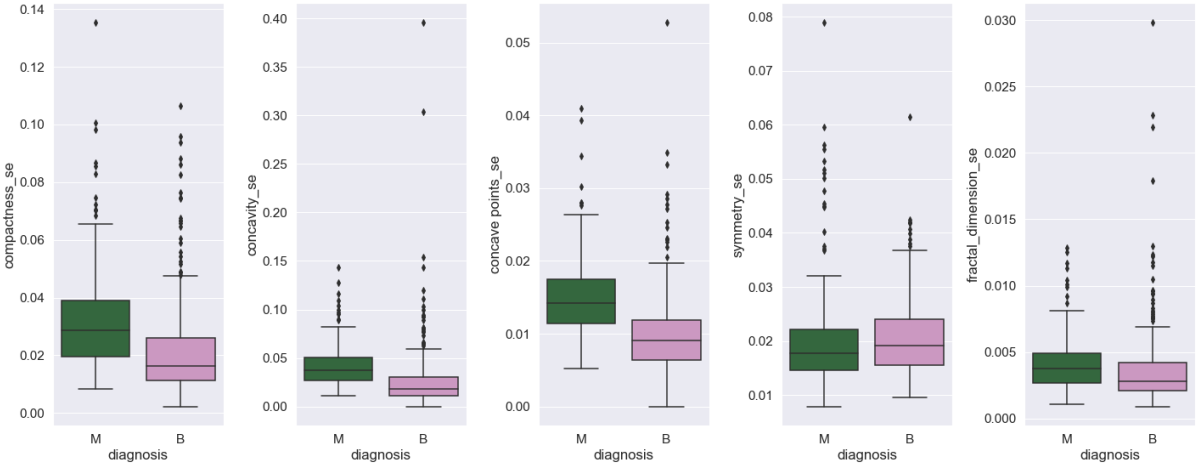


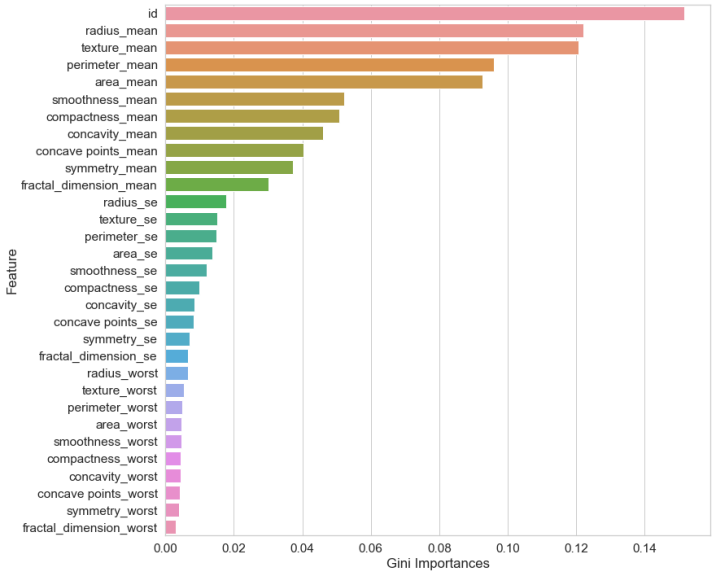
In [13]:
```
plt.rcParams['figure.figsize']=(20,8)
f, (ax1,ax2,ax3,ax4,ax5) = plt.subplots (1,5)
s.boxplot ('diagnosis', y = 'radius_mean',data = data , ax = ax1)
s.boxplot ('diagnosis', y = 'texture_mean',data = data , ax = ax2)
s.boxplot ('diagnosis', y = 'perimeter_mean',data = data , ax = ax3)
s.boxplot ('diagnosis', y = 'area_mean',data = data , ax = ax4)
s.boxplot ('diagnosis', y = 'smoothness_mean',data = data , ax = ax5)
f .tight_layout()

f, (ax1,ax2,ax3,ax4,ax5) = plt.subplots (1,5)
s.boxplot ('diagnosis', y = 'compactness_mean',data = data , ax = ax1)
s.boxplot ('diagnosis', y = 'concavity_mean',data = data , ax = ax2)
s.boxplot ('diagnosis', y = 'concave points_mean',data = data , ax = ax3)
s.boxplot ('diagnosis', y = 'symmetry_mean',data = data , ax = ax4)
s.boxplot ('diagnosis', y = 'fractal_dimension_mean',data = data , ax = ax5)
f .tight_layout()
```



In [15]:
```
plt.rcParams['figure.figsize']=(20,8)
f, (ax1,ax2,ax3,ax4,ax5) = plt.subplots (1,5)
s.boxplot ('diagnosis', y = 'radius_se',data = data , ax = ax1,palette = 'cubehelix')
s.boxplot ('diagnosis', y = 'texture_se',data = data, ax = ax2,palette = 'cubehelix')
s.boxplot ('diagnosis', y = 'perimeter_se',data = data , ax = ax3,palette = 'cubehelix')
s.boxplot ('diagnosis', y = 'area_se',data = data , ax = ax4,palette = 'cubehelix')
s.boxplot ('diagnosis', y = 'smoothness_se',data = data , ax = ax5,palette = 'cubehelix')
f .tight_layout()

f, (ax1,ax2,ax3,ax4,ax5) = plt.subplots (1,5)
s.boxplot ('diagnosis', y = 'compactness_se',data = data , ax = ax1,palette = 'cubehelix')
s.boxplot ('diagnosis', y = 'concavity_se',data = data , ax = ax2,palette = 'cubehelix')
s.boxplot ('diagnosis', y = 'concave points_se',data = data , ax = ax3,palette = 'cubehelix')
s.boxplot ('diagnosis', y = 'symmetry_se',data = data , ax = ax4,palette = 'cubehelix')
s.boxplot ('diagnosis', y = 'fractal_dimension_se',data = data , ax = ax5,palette = 'cubehelix')
f .tight_layout()
```







The graphs for this model will provide visual representations of various aspects related to breast cancer prediction. These graphs can include performance metrics such as accuracy, precision, recall, and F1 score, which will help evaluate the effectiveness of the prediction model. Additionally, graphical representations of feature importance can help identify the most influential variables in the prediction process. Furthermore, ROC curves and precision-recall curves can be plotted to assess the model's overall performance and determine an optimal threshold for classification. Overall, the graphs will provide valuable insights into the model's predictive capabilities and performance.

6. Business Opportunity

The business opportunity in a breast cancer prediction model lies in several areas. Firstly, healthcare providers and clinics can leverage the model to enhance their diagnostic capabilities, leading to earlier detection and improved patient outcomes. This can differentiate them in the market by offering more accurate and efficient screening services.

Secondly, pharmaceutical companies and research institutions can utilize the model to identify potential candidates for clinical trials, aiding in the development of new treatments and therapies for breast cancer.

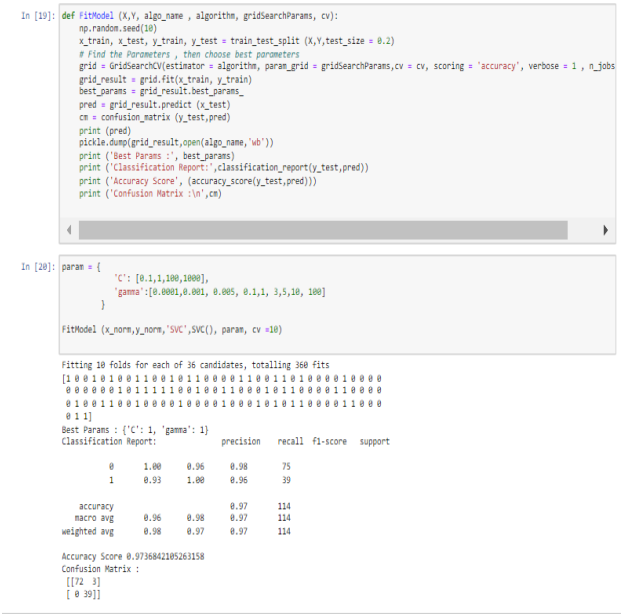
Furthermore, insurance companies can leverage the model to assess the risk profiles of their policyholders and offer tailored coverage plans for individuals at higher risk.

Overall, the breast cancer prediction model presents opportunities for improved healthcare delivery, advancements in medical research, and more personalized insurance offerings, benefiting both businesses and individuals in the fight against breast cancer.

7. Concept Generation

The breast cancer prediction model requires the utilization of machine learning models. Instead of writing the models from scratch, a more feasible approach would be to adapt and tweak existing models to suit our specific needs. This approach saves time and effort compared to starting from scratch.

By leveraging well-trained models, we can repurpose them or make necessary modifications. However, building a model with the available resources and data may be time-consuming but achievable. To ensure accuracy, it is important to not solely rely on classic machine learning algorithms and invest efforts in refining the model. Additionally, the aim should be to minimize the customer's input data requirements, streamlining the prediction process.

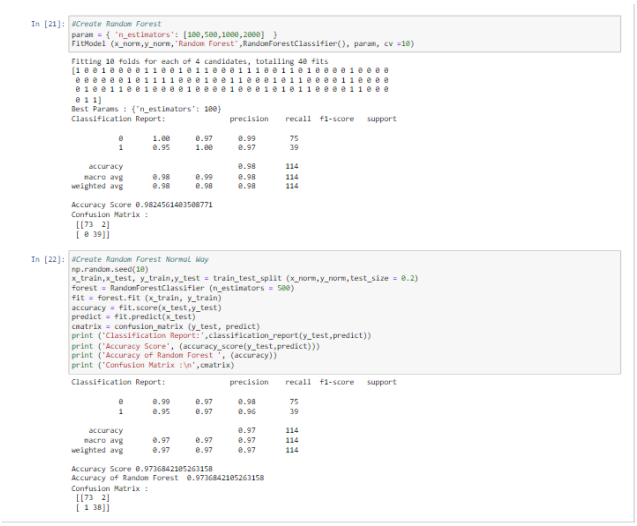


```
In [19]: def FitModel (X,Y, algo_name , algorithm, gridSearchParams, cv):
             np.random.seed(10)
             x_train, x_test, y_train, y_test = train_test_split (X,Y,test_size = 0.2)
             # Find the Parameters , then choose best parameters
             grid = GridSearchCV(estimator = algorithm, param_grid = gridSearchParams,cv = cv, scoring = 'accuracy', verbose = 1 , n_jobs
             grid_result = grid.fit(x_train, y_train)
             best_params = grid_result.best_params_
             pred = grid_result.predict (x_test)
             cm = confusion_matrix (y_test,pred)
             print (pred)
             pickle.dump(grid_result,open(algo_name,'wb'))
             print ('Best Params :', best_params)
             print ('Classification Report:',classification_report(y_test,pred))
             print ('Accuracy Score', (accuracy_score(y_test,pred)))
             print ('Confusion Matrix :\n',cm)
```



```
In [20]: param = {
                     'C': [0.1,1,100,1000],
                     'gamma':[0.0001,0.001, 0.005, 0.1,1, 3,5,10, 100]
                 }

         FitModel (x_norm,y_norm,'SVC',SVC(), param, cv =10)
```

```
Fitting 10 folds for each of 36 candidates, totalling 360 fits
[1 0 0 1 0 1 0 0 1 1 0 0 1 0 1 1 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0
 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0
 0 1 0 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0
 0 1 1]
Best Params : {'C': 1, 'gamma': 1}
Classification Report:              precision    recall  f1-score   support

           0       1.00      0.96      0.98        75
           1       0.93      1.00      0.96        39

    accuracy                           0.97       114
   macro avg       0.96      0.98      0.97       114
weighted avg       0.98      0.97      0.97       114

Accuracy Score 0.9736842105263158
Confusion Matrix :
 [[72  3]
 [ 0 39]]
```



```
In [21]: #Create Random Forest
         param = { 'n_estimators': [100,500,1000,2000]  }
         FitModel (x_norm,y_norm,'Random Forest',RandomForestClassifier(), param, cv =10)
```

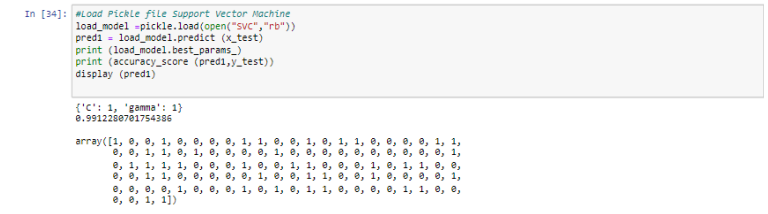
```
Fitting 10 folds for each of 4 candidates, totalling 40 fits
[1 0 0 1 0 0 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0
 0 0 0 0 0 1 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0
 0 1 0 0 1 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 1 1 0 0 0 0
 0 1 1]
Best Params : {'n_estimators': 100}
Classification Report:              precision    recall  f1-score   support

           0       1.00      0.97      0.99        75
           1       0.95      1.00      0.97        39

    accuracy                           0.98       114
   macro avg       0.98      0.99      0.98       114
weighted avg       0.98      0.98      0.98       114

Accuracy Score 0.9824561403508771
Confusion Matrix :
 [[73  2]
 [ 0 39]]
```



```
In [22]: #Create Random Forest Normal Way
         np.random.seed(10)
         x_train,x_test, y_train,y_test = train_test_split (x_norm,y_norm,test_size = 0.2)
         forest = RandomForestClassifier (n_estimators = 500)
         fit = forest.fit (x_train, y_train)
         accuracy = fit.score(x_test,y_test)
         predict = fit.predict(x_test)
         cmatrix = confusion_matrix (y_test, predict)
         print ('Classification Report:',classification_report(y_test,predict))
         print ('Accuracy Score', (accuracy_score(y_test,predict)))
         print ('Accuracy of Random Forest ', (accuracy))
         print ('Confusion Matrix :\n',cmatrix)
```

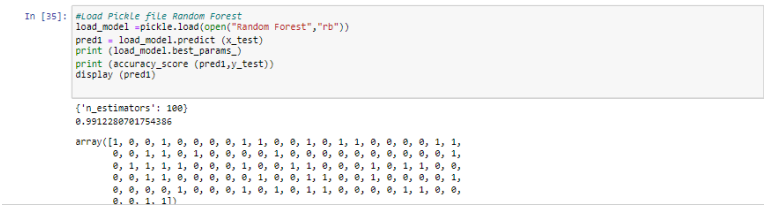
```
Classification Report:              precision    recall  f1-score   support

           0       0.99      0.97      0.98        75
           1       0.95      0.97      0.96        39

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

Accuracy Score 0.9736842105263158
Accuracy of Random Forest  0.9736842105263158
Confusion Matrix :
 [[73  2]
 [ 1 38]]
```

The Accuracy of the initial model is given below:



```
In [34]: #Load Pickle file Support Vector Machine
         load_model =pickle.load(open("SVC","rb"))
         pred1 = load_model.predict (x_test)
         print (load_model.best_params_)
         print (accuracy_score (pred1,y_test))
         display (pred1)
```

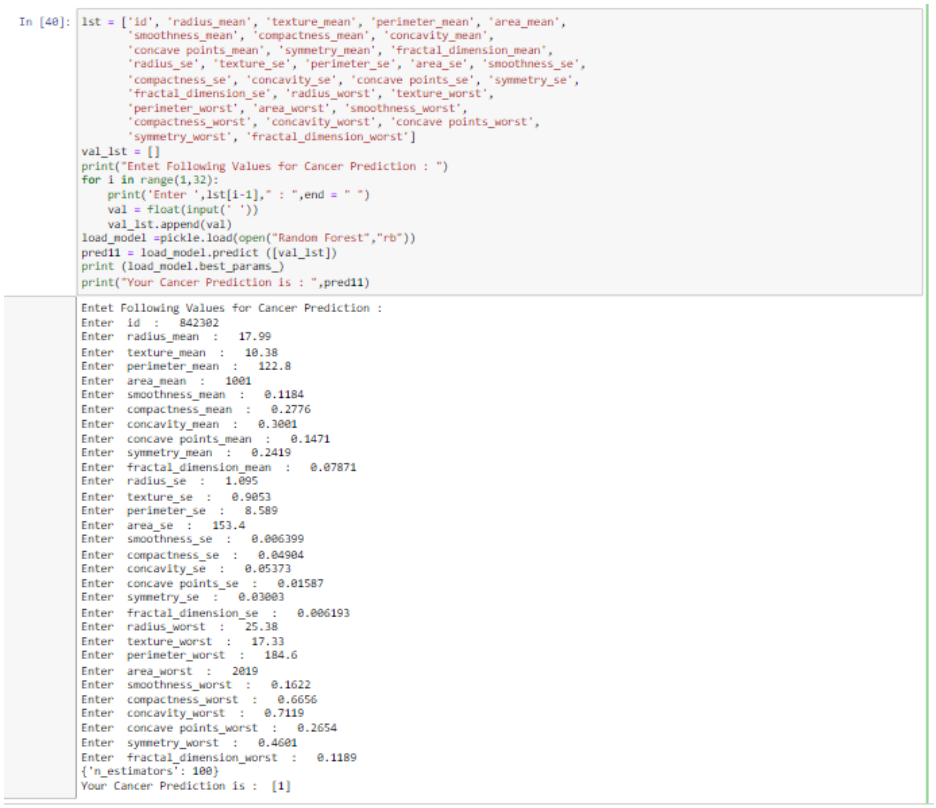
```
{'C': 1, 'gamma': 1}
0.9912280701754386

array([1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1,
       0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 1, 1])
```



```
In [35]: #Load Pickle file Random Forest
         load_model =pickle.load(open("Random Forest","rb"))
         pred1 = load_model.predict (x_test)
         print (load_model.best_params_)
         print (accuracy_score (pred1,y_test))
         display (pred1)
```

```
{'n_estimators': 100}
0.9912280701754386

array([1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1,
       0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
       0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 1, 1])
```

The output for the initial model is given below:



```
In [40]: lst = ['id', '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']
         val_lst = []
         print("Entet Following Values for Cancer Prediction : ")
         for i in range(1,32):
             print('Enter ',lst[i-1]," : ",end = " ")
             val = float(input(' '))
             val_lst.append(val)
         load_model =pickle.load(open("Random Forest","rb"))
         pred11 = load_model.predict ([val_lst])
         print (load_model.best_params_)
         print("Your Cancer Prediction is : ",pred11)

Entet Following Values for Cancer Prediction :
Enter  id  :  842302
Enter  radius_mean  :  17.99
Enter  texture_mean  :  10.38
Enter  perimeter_mean  :  122.8
Enter  area_mean  :  1001
Enter  smoothness_mean  :  0.1184
Enter  compactness_mean  :  0.2776
Enter  concavity_mean  :  0.3001
Enter  concave points_mean  :  0.1471
Enter  symmetry_mean  :  0.2419
Enter  fractal_dimension_mean  :  0.07871
Enter  radius_se  :  1.095
Enter  texture_se  :  0.9053
Enter  perimeter_se  :  8.589
Enter  area_se  :  153.4
Enter  smoothness_se  :  0.006399
Enter  compactness_se  :  0.04904
Enter  concavity_se  :  0.05373
Enter  concave points_se  :  0.01587
Enter  symmetry_se  :  0.03003
Enter  fractal_dimension_se  :  0.006193
Enter  radius_worst  :  25.38
Enter  texture_worst  :  17.33
Enter  perimeter_worst  :  184.6
Enter  area_worst  :  2019
Enter  smoothness_worst  :  0.1622
Enter  compactness_worst  :  0.6656
Enter  concavity_worst  :  0.7119
Enter  concave points_worst  :  0.2654
Enter  symmetry_worst  :  0.4601
Enter  fractal_dimension_worst  :  0.1189
{'n_estimators': 100}
Your Cancer Prediction is :  [1]
```

8. Concept Development

The concept for developing a breast cancer prediction system involves leveraging appropriate technologies such as Flask API and Django framework. Flask API can be used to create a robust and scalable API that can handle user input and communicate with the underlying prediction model. Django framework can provide a solid foundation for developing the web application, managing the database, and handling user authentication. For deployment, cloud services like AWS, Azure, or Google Cloud can be chosen based on the specific requirements of the application. The selected cloud service will provide the necessary infrastructure and scalability for hosting the application securely.

9. Final Report Prototype

The final report prototype for the breast cancer prediction product outlines the necessary functions to achieve optimal results. The development process encompasses both the back-end and front-end aspects, focusing on model development and user interface enhancements.

Back-end:

1. Model Development: Extensive manual supervised machine learning is required to optimize automated tasks, including performing Exploratory Data Analysis (EDA) to identify dependent and independent features.

2. Algorithm Training and Optimization: Minimizing model overfitting and hyperparameter tuning are crucial steps to ensure accurate predictions.

Front-end:

1. Different User Interface: Providing users with multiple parameter options requires thorough testing and analysis, considering various edge cases.

2. Interactive Visualization: Raw and complex data extracted from trained models must be presented in an aesthetically pleasing and easy-to-understand format.

3. Feedback System: Developing a valuable feedback mechanism enables understanding of unmet customer needs, facilitating continuous model improvement.

The final report prototype highlights the importance of optimizing the model's performance, refining the user interface, and incorporating user feedback to create a comprehensive and effective breast cancer prediction product.

13. Product Details - How does it work?

The cancer prediction product utilizes an interactive user system to provide accurate predictions based on input data. Here is how the system works:

1. User Input: The user provides relevant information, such as medical history, diagnostic test results, and demographic details.

2. Data Preprocessing: The input data undergoes preprocessing steps, including cleaning, normalization, and feature engineering, to ensure compatibility with the prediction model.

3. Prediction Model: The preprocessed data is fed into the trained prediction model, which employs advanced machine learning algorithms to analyze the input and generate a prediction regarding the likelihood of breast cancer.

4. Real-Time Results: The system provides real-time predictions, presenting the user with the probability or classification outcome of breast cancer. This information helps users make informed decisions and take appropriate actions.

5. User Interactive UI: The system incorporates a user-friendly interface that allows users to input and modify data, visualize results, and explore additional insights. This interactive UI enhances the user experience and facilitates easy interpretation of the prediction outcomes.

By following this process, the cancer prediction product enables users to obtain personalized and timely predictions regarding the probability of breast cancer based on their input data.

14. References/Source of Information

During the development of this report on breast cancer prediction product, the following sources of information were referenced:

1. Dataset: The primary dataset used for training and evaluation was obtained from Kaggle (www.kaggle.com). The specific dataset used can be found at Breast Cancer Wisconsin (Diagnostic) Data Set | Kaggle.

2. Research Papers and Journals: Various scientific research papers and journals were consulted to gain insights into breast cancer prediction methods, feature selection techniques, machine learning algorithms, and evaluation metrics. Some notable sources include:

- (PDF) BREAST CANCER PREDICTION USING MACHINE LEARNING (researchgate.net)
- A Comparative Analysis of Breast Cancer Detection and Diagnosis Using Data Visualization and Machine Learning Applications - PMC (nih.gov)