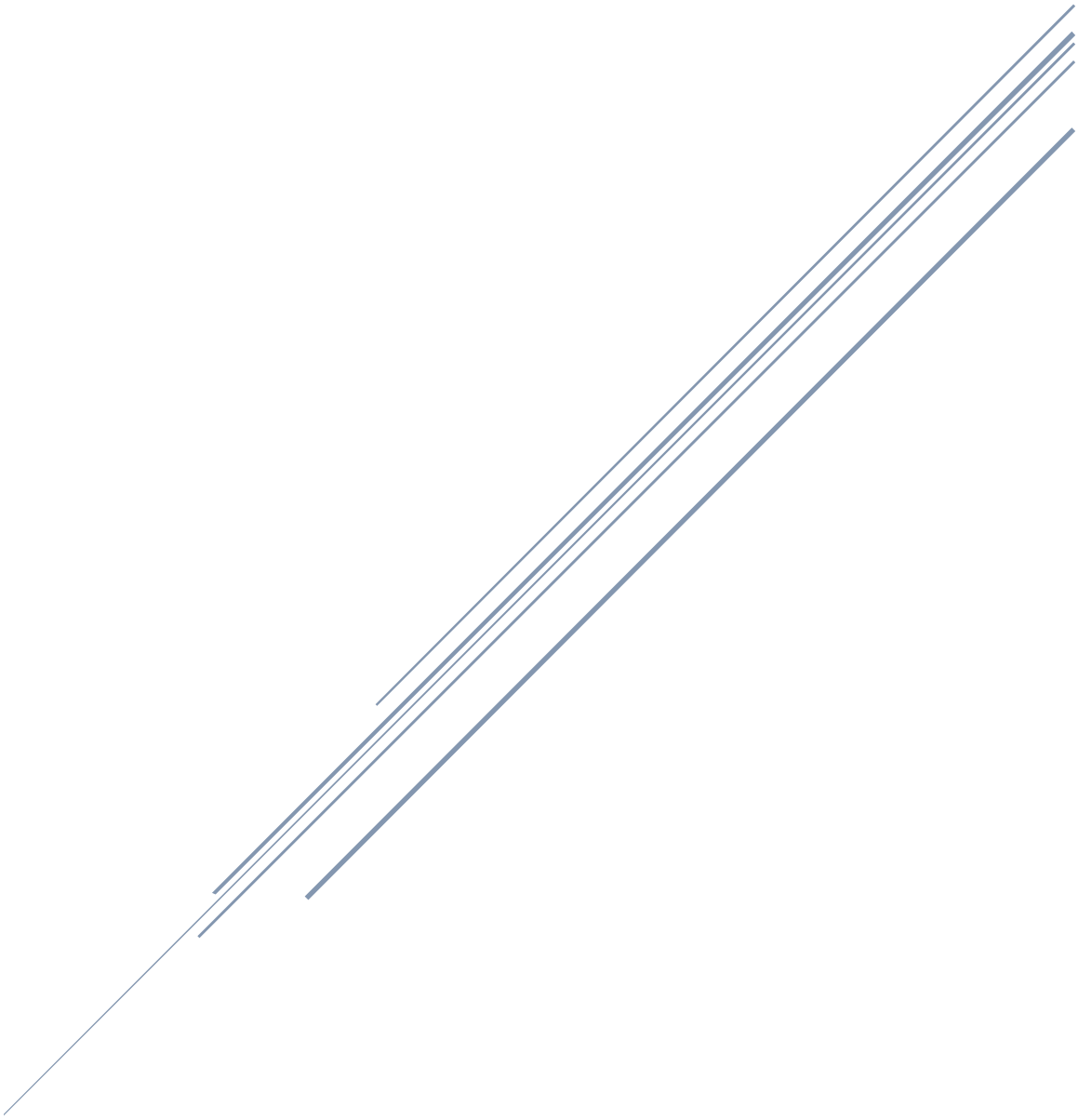
BREAST CANCER PREDICTION

Milestone 4 - Finalizing Your Results



BREAST CANCER PREDICTION

Table of Contents

Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

BREAST CANCER PREDICTION

Milestone 2 - Data Selection and Project Proposal

Introduction

At first, I had planned to examine the diabetes dataset for my project. However, due to insufficient data, I have opted to focus on breast cancer prediction for the final project. Breast cancer is a type of cancer that affects the cells in the breasts. It is a significant concern for women and can also occur in men. Fortunately, thanks to scientific advancements, the mortality rates associated with breast cancer are decreasing. Detecting breast cancer at an early stage can improve the chances of survival.

Dataset Source

I found the breast cancer dataset on Kaggle (Pmotta. (2021, June 6). *Breast cancer prediction*. Kaggle) to be a helpful starting point, as it provides valuable information required for the analysis.

Models

I plan to build the following models with this dataset.

1. Decision Tree Implementation

   A decision tree is a useful tool for breaking down complex data into more manageable parts. From what I understand, decision trees are often used for prediction analysis, data classification, and regression.

2. Random Forest Classification

   Random forest classification helps with any overfitting issues with decision trees. Random Forest Classification also maintains its accuracy in case of missing data.

3. KNN Classification

Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

BREAST CANCER PREDICTION

I choose to evaluate the KNN Classification model as it is a non-parametric algorithm, which means it does not make assumptions about the underlying data distribution. This flexibility allows it to capture complex relationships between features, making it potentially suitable for breast cancer prediction.

Based on the outcomes from the models, I would also like to explore other models.

How do you plan to evaluate your results

To evaluate a model's results effectively, it is important to go beyond the accuracy score alone. As such I plan to evaluate models using train-test-split, by dividing the dataset into a training and a test set. I would also like to construct a confusion matrix that gives a comprehensive view of the model's performance for different classes and helps identify any specific issues, such as imbalanced class predictions.

What do you hope to learn

Using this analysis, I am hoping to understand the factors and symptoms that cause Breast Cancer. Based on the outcome the objective is to help early detection of breast cancer.

Assess any risks and ethical implications with your proposal

The Kaggle dataset is sourced from samples that arrive periodically as Dr. Wolberg reports his clinical cases. The dataset has not been updated in years making it unclear if other factors could potentially cause breast cancer.

Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

Identify a contingency plan if your original project plan does not work out

At this point, I am hoping we have enough information to begin the modeling process. However, based on my previous experience, I encountered an issue where I began to doubt the accuracy of my models and suspected that the dataset lacked essential features to make accurate predictions. In preparation for this scenario, I started working on a similar dataset within the same field. Similarly, for my current project, I will have a contingency plan based on the same grounds. I will primarily look for similar and more recent datasets in the same domain. As an alternative option, if the dataset does not work, I will switch to the airline delay prediction dataset.

Include anything else you believe is important

Feature selection is key for model building. I plan to spend some time understanding the dataset to ensure the right features are selected for model building. I would also like to explore other models such as XGBClassifier and/or GaussianNB.

References

Pmotta. (2021, June 6). *Breast cancer prediction*. Kaggle - https://www.kaggle.com/code/pmotta/breast-cancer-prediction/input

BREAST CANCER PREDICTION

Milestone 3 - Preliminary Analysis

Data Expectations

The selected breast cancer dataset contains key features that reflect the underlying health factors of breast cancer patients, which should be adequate for prediction purposes. However, before modeling, the dataset needs to be prepared and data balance checks must be conducted.

If I am unable to find the answers with the selected dataset, as an alternative, I have identified another dataset for breast cancer prediction (Architkuiya. (2022, November 28). Breast_cancer_detection).

Visualizations

For this analysis, I would like to use the following visualizations:

i. A Bar/Pie chart with the number of benign vs malignant tumors to identify data balance. This is useful to identify any imbalance in the dataset before modeling to avoid bias.
ii. Different Bar charts for features such as (Cell size, Clump Thickness, and Cell shape) vs Class (Benign/Malignant) to see the effect of these features on cancerous cells.
iii. A correlation matrix (heatmap) to observe the correlations between features.

I would also like to build line and/or scatter plots based on features with a strong correlation with the class variable (Benign/Malignant), based on the outcome from the correlation heatmap. However, since we already would have built a bar chart for different features vs Class, I doubt if the scatter/line charts would add any additional value.

Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

BREAST CANCER PREDICTION

Data Preparation

As a first step in data preparation, I would perform some checks such as

1. Duplicate checks – Remove duplicate rows from the dataset
2. NA/Null checks – Replace missing values with the median value of the column
3. Drop empty rows and columns, if any.

Additionally, if from visualization (1), the dataset turns out to be imbalanced, I would have to implement techniques such as SMOTE (Synthetic Minority Oversampling Technique) to balance the data before model building.

Additional Model Selection

My main choices for modeling breast cancer prediction include Decision Tree, Random Forest, and KNN. If time allows, I also aim to investigate the implementation of an SVM (support vector machines) model. This is because SVMs possess the capability to capture complex relationships among features, which is beneficial considering the non-linear nature of the data. SVMs are popular with health datasets, as they are recognized for their proficiency in performing both linear and non-linear classifications. The ultimate objective is to assess the accuracy of each model and determine which one yields the most precise predictions.

Expectations

I believe my primary goal of predicting breast cancer is still achievable with the available dataset features. As a contingency plan, I have an alternate dataset for breast cancer and also an airline delay prediction dataset. I am confident that I can build decent models with the available breast cancer dataset.

Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

References

Architkuiya. (2022, November 28). *Breast_cancer_detection*. Kaggle. https://www.kaggle.com/code/architkuiya/breast-cancer-detection/data

Milestone 4 - Finalizing Your Results

Data Preparation

The selected dataset is relatively small and demands only a modest amount of cleaning and preparation steps. Nonetheless, certain validations are essential to ready the data for modeling purposes. The characteristics of the dataset's attributes (Patel, J., Patel , U., Patel, R., & Shah, P. (2019, April 28). Breast Cancer Analysis) are outlined in the following table.

Clump Thickness:	(1-10). Benign cells tend to be grouped in monolayers, while cancerous cells are often grouped in multilayers.
Uniformity of Cell Size:	(1-10). Cancer cells tend to vary in size and shape. That is why these parameters are valuable in determining whether the cells are cancerous or not.
Uniformity of Cell Shape:	(1-10). Uniformity of cell size/shape: Cancer cells tend to vary in size and shape. That is why these parameters are valuable in determining whether the cells are cancerous or not.
Marginal Adhesion:	(1-10). Normal cells tend to stick together. Cancer cells tend to lose this ability. So, the loss of adhesion is a sign of malignancy.
Single Epithelial Cell Size:	(1-10). It is related to the uniformity mentioned above. Epithelial cells that are significantly enlarged may be a malignant cell.
Bare Nuclei:	(1-10). This is a term used for nuclei not surrounded by cytoplasm (the rest of the cell). Those are typically seen in benign tumors.

Bland Chromatin:	(1-10). Describes a uniform "texture" of the nucleus seen in benign cells. In cancer cells, the chromatin tends to be more coarse.
Normal Nucleoli:	(1-10). Nucleoli are small structures seen in the nucleus. In normal cells, the nucleolus is usually very small if visible at all. In cancer cells, the nucleoli become more prominent, and sometimes there are more of them.
Mitoses:	(1-10). Cancer is essentially a disease of uncontrolled mitosis.
Class:	(2 or 4) (2) Benign (non-cancerous) or (4) malignant (cancerous) lump in a breast.

The executed steps for data preparation are as follows:

i. Renaming columns to enhance accessibility and clarity.

ii. Checking for null rows and columns. This step aimed to remove columns or rows with missing data. Notably, the dataset did not contain any empty rows or columns.

iii. Detecting duplicates to eliminate redundant information. No duplicate entries were identified.

iv. Identify data classification by “Class” columns. The dataset is divided into benign and malignant classes.

Rename Columns

```
breast_cancer_df.columns = ['Sample_code_number','Clump_Thickness','Uniformity_Cell_Size','Uniformity_Cell_Shape',
                            'Marginal_Adhesion','Single_Epithelial_Cell_Size','Bare_Nuclei','Bland_Chromatin',
                            'Normal_Nucleoli','Mitoses','Class']

breast_cancer_df.head(5)
```

3]:

Uniformity_Cell_Size	Uniformity_Cell_Shape	Marginal_Adhesion	Single_Epithelial_Cell_Size	Bare_Nuclei	Bland_Chromatin	Normal_Nucleoli	Mitoses	Class
1	1	1	2	1	3	1	1	2
4	4	5	7	10	3	2	1	2
1	1	1	2	2	3	1	1	2
8	8	1	3	4	3	7	1	2
1	1	3	2	1	3	1	1	2

I renamed columns by replacing spaces with '_' (underscore) for ease of column reference.

Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

BREAST CANCER PREDICTION

Check for null rows and/or columns

```
1 breast_cancer_df.isnull().sum()
```

```
[4]: Sample_code_number            0
     Clump_Thickness               0
     Uniformity_Cell_Size          0
     Uniformity_Cell_Shape         0
     Marginal_Adhesion             0
     Single_Epithelial_Cell_Size   0
     Bare_Nuclei                   0
     Bland_Chromatin               0
     Normal_Nucleoli               0
     Mitoses                       0
     Class                         0
     dtype: int64
```

Check for duplicates

```
1 print('Dataframe before dropping duplicates :', breast_cancer_df.shape)
2 flight_data_df = breast_cancer_df.drop_duplicates() # 1,389 rows dropped
3 print('Dataframe after dropping duplicates :',breast_cancer_df.shape)
```

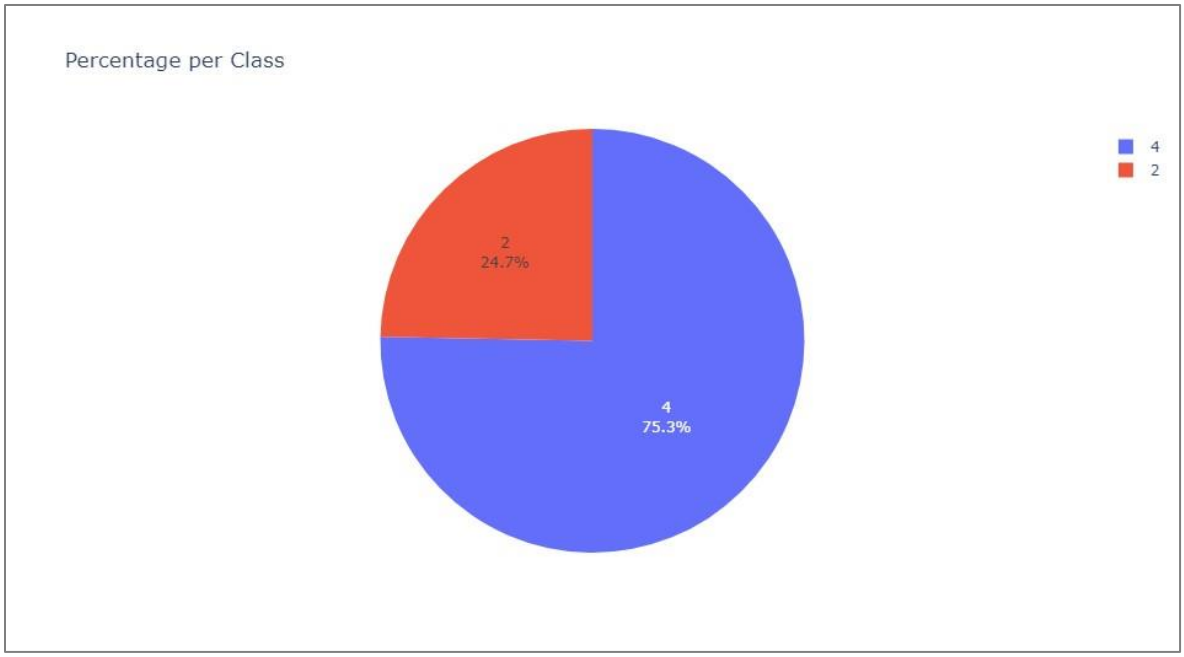
```
Dataframe before dropping duplicates : (683, 11)
Dataframe after dropping duplicates : (683, 11)
```

A pie chart was constructed for the "Class" attribute to detect potential data imbalances. The analysis revealed an imbalance in the data, with 75.3% categorized as malignant and 24.7% as benign.

PIE CHART

```
1 fig = px.pie(breast_cancer_df, values='Bare_Nuclei', names='Class', title='Percentage per Class')
2 fig.update_traces(textposition='inside', textinfo='percent+label')
3 fig.show("notebook")
```

Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)



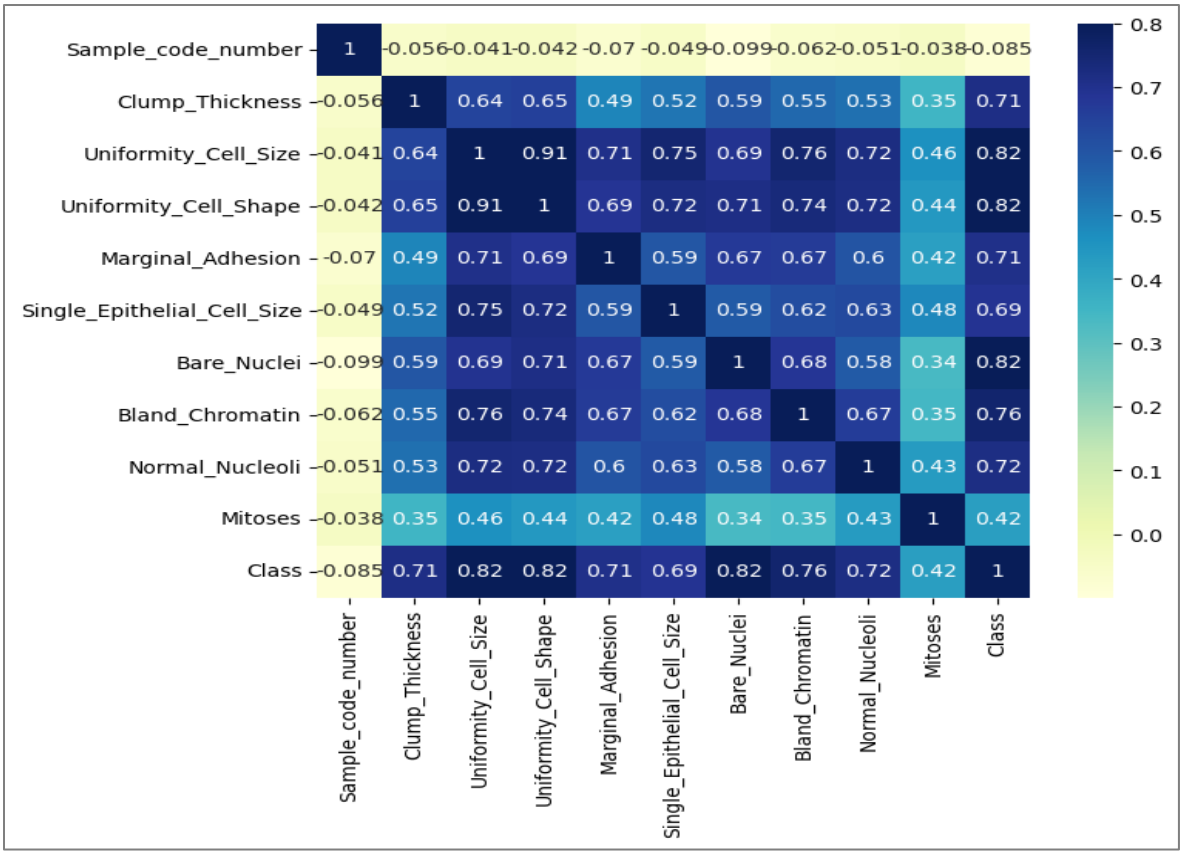
Further visualizations were generated to uncover correlations among the attributes to help get a better understanding of the available features.

Visualizations

HEATMAP

```
1 corrmat = breast_cancer_df.corr()
2 f, ax = plt.subplots(figsize=(8, 6))
3 sns.heatmap(corrmat, vmax=.8, square=True,annot=True,cmap='YlGnBu');
4 plt.show()
```

BREAST CANCER PREDICTION



SCATTER PLOTS

```
sns.set(font_scale=1)
sns.set_theme('notebook', style='dark')
plt.style.use("grayscale")

sns.pairplot(breast_cancer_df, hue='Class',
             vars=['Clump_Thickness','Uniformity_Cell_Size','Uniformity_Cell_Shape',
                   'Marginal_Adhesion','Single_Epithelial_Cell_Size','Bare_Nuclei','Bland_Chromatin',
                   'Normal_Nucleoli','Mitoses'])
```

Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

BREAST CANCER PREDICTION

To construct an effective model, it was essential to address the dataset's imbalance. To achieve this, the Synthetic Minority Oversampling Technique (SMOTE) was employed. SMOTE generates synthetic instances for the minority class, thus balancing the dataset. With the dataset

Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

now balanced, the subsequent step was to partition the data into training and testing datasets.

SMOTE to balance the imbalanced data

```
smote = SMOTE()
x, y = smote.fit_resample(X, Y)
```

```
#Split the smote (balanced) data into train and test subsets:
x_train_sm, x_test_sm, y_train_sm, y_test_sm = train_test_split(x,y,test_size = 0.2, random_state=0)
```

The following models were built, and the observations are noted. Following the initial plan, I assessed the model's performance using metrics like accuracy, precision, recall, and the confusion matrix. These insights guided decisions to enhance its predictive abilities.

Models

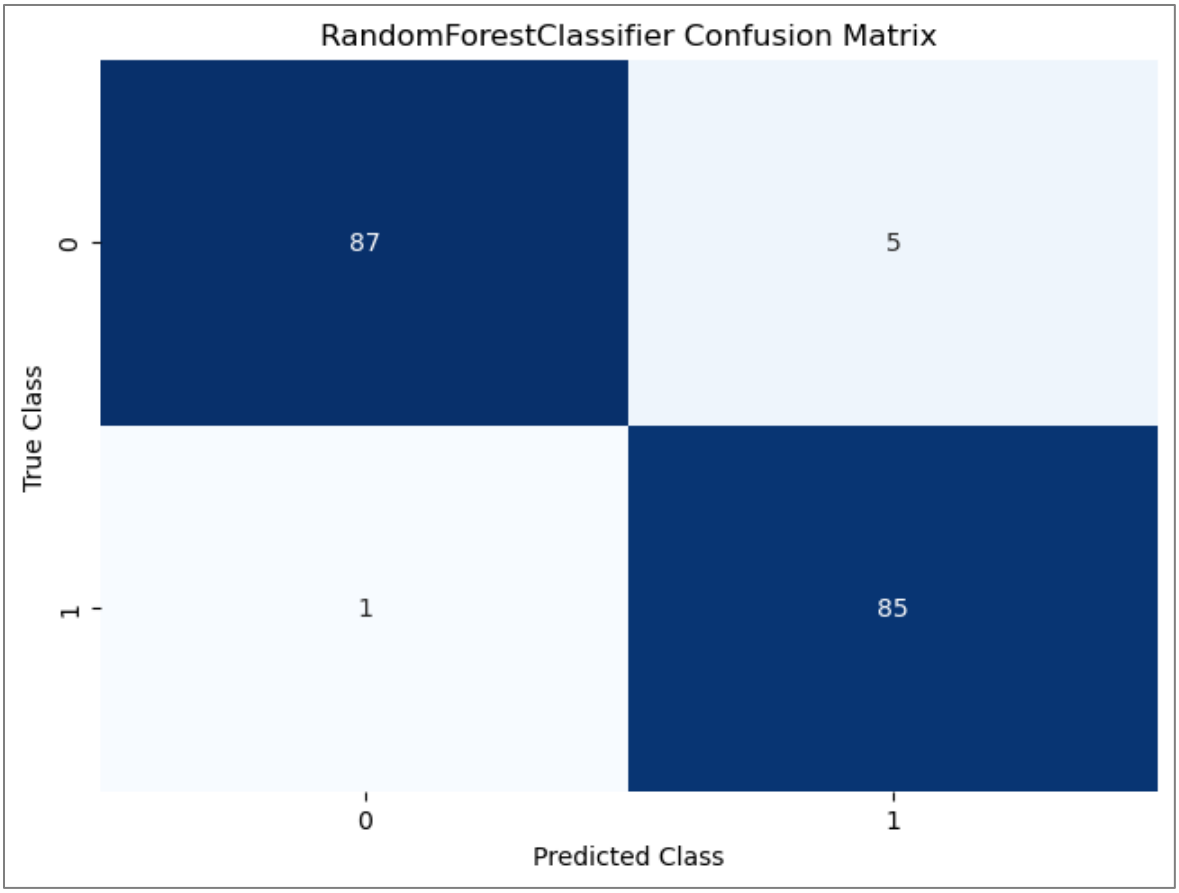
Random Forest Classifier

```
rfc = RandomForestClassifier()
rfc.fit(x_train_sm, y_train_sm)

y_pred_sm = rfc.predict(x_test_sm)
```

```
#Build the confusion matrix
matrix_sm = confusion_matrix(y_test_sm, y_pred_sm)

print(matrix_sm)

# Create pandas dataframe
df_sm = pd.DataFrame(matrix_sm)

# Create a heatmap
sns.heatmap(df_sm, annot=True, cbar=None, cmap="Blues",fmt='.0f')
plt.title("RandomForestClassifier Confusion Matrix"), plt.tight_layout()
plt.ylabel("True Class"), plt.xlabel("Predicted Class")
plt.show()
```

```
[[87  5]
 [ 1 85]]
```

BREAST CANCER PREDICTION




```
print(classification_report(y_test_sm, y_pred_sm))
print(confusion_matrix(y_test_sm, y_pred_sm))
print(f'ROC-AUC score : {roc_auc_score(y_test_sm, y_pred_sm)}')
print(f'Accuracy score : {accuracy_score(y_test_sm, y_pred_sm)}')
```

```
              precision    recall  f1-score   support

           2       0.99      0.95      0.97        92
           4       0.94      0.99      0.97        86

    accuracy                           0.97       178
   macro avg       0.97      0.97      0.97       178
weighted avg       0.97      0.97      0.97       178

[[87  5]
 [ 1 85]]
ROC-AUC score : 0.9670121334681496
Accuracy score : 0.9662921348314607
```

For predicting cases of benign tumors (class 2), the model has a high precision (0.99), meaning that when it predicts a tumor as benign, it is correct 99% of the time. The recall (sensitivity) is 0.95, indicating that the model captures 95% of the actual benign cases. The F1-score (a balance between precision and recall) is 0.97.

Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

For predicting cases of malignant tumors (class 4), the model has slightly lower precision (0.94) compared to class 2, but higher recall (0.99). This suggests that while the model might be slightly less precise in predicting malignant cases, it is very effective at capturing most of the actual malignant cases. The F1-score for class 4 is also 0.97. The confusion matrix indicates that there were 5 false positive predictions and 1 false negative prediction. The accuracy of 0.97 indicates that the model correctly predicts around 97% of the instances in the dataset.

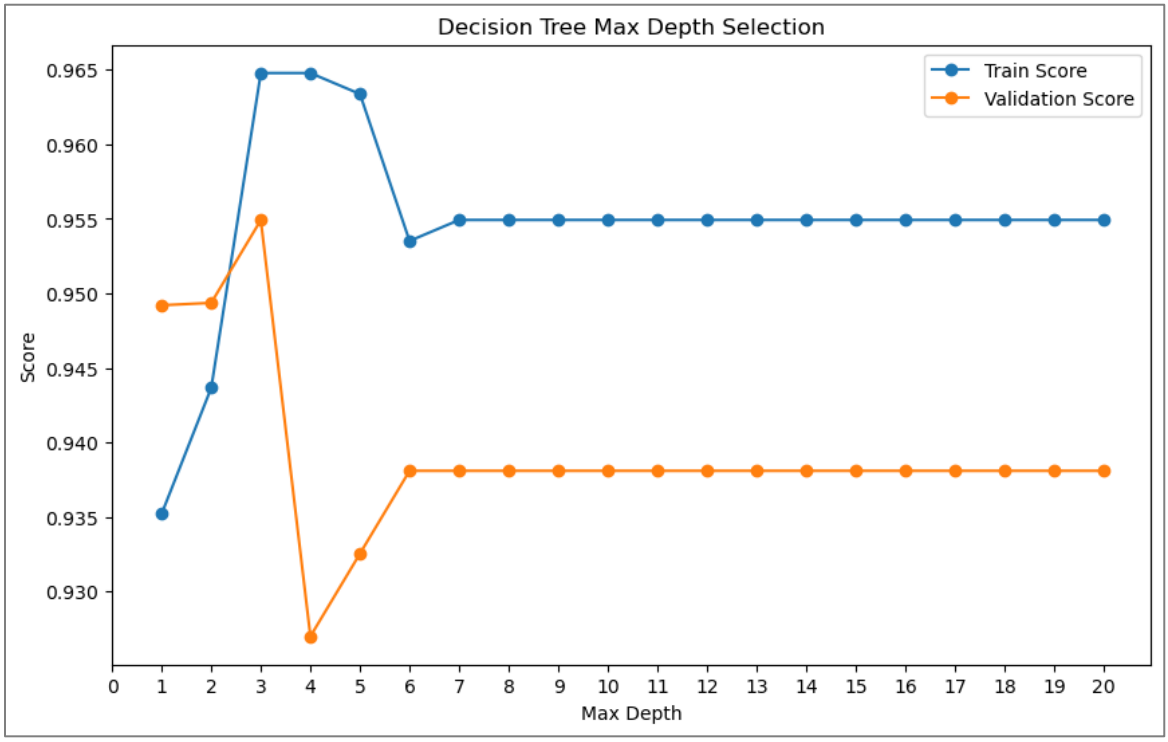
Overall, the model appears to perform well with high precision and recall for both classes, as well as a strong F1-score. This suggests that the Random Forest Classifier is effective for predicting breast cancer, distinguishing between benign and malignant tumors based on the provided metrics.

Decision Tree Classifier

Initially, the model was configured with a max_depth of 5, which did not yield the optimal accuracy. However, by conducting cross-validation across various max_depth values, it became evident that the best accuracy was achieved when using a max_depth of 3. This iterative process of exploring different max_depth values through cross-validation led to the identification of an improved accuracy level.

DecisionTreeClassifier

```
# Range of max depth values to try
max_depth_values = np.arange(1, 21)

# Lists to store cross-validation scores
train_scores = []
val_scores = []

# Perform cross-validation for different max depth values
for max_depth in max_depth_values:
    model = DecisionTreeClassifier(max_depth=max_depth, random_state=0)
    train_score = np.mean(cross_val_score(model, x_train_sm, y_train_sm, cv=5))
    val_score = np.mean(cross_val_score(model, x_test_sm, y_test_sm, cv=5))
    train_scores.append(train_score)
    val_scores.append(val_score)

# Plot the validation curve
plt.figure(figsize=(10, 6))
plt.plot(max_depth_values, train_scores, label='Train Score', marker='o')
plt.plot(max_depth_values, val_scores, label='Validation Score', marker='o')
plt.xticks(np.arange(0, 21, 1.0))
plt.xlabel('Max Depth')
plt.ylabel('Score')
plt.title('Decision Tree Max Depth Selection')
plt.legend()
plt.show()
```



The above plot shows that the best accuracy for the model is when the parameter max_depth is 3.

Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

BREAST CANCER PREDICTION

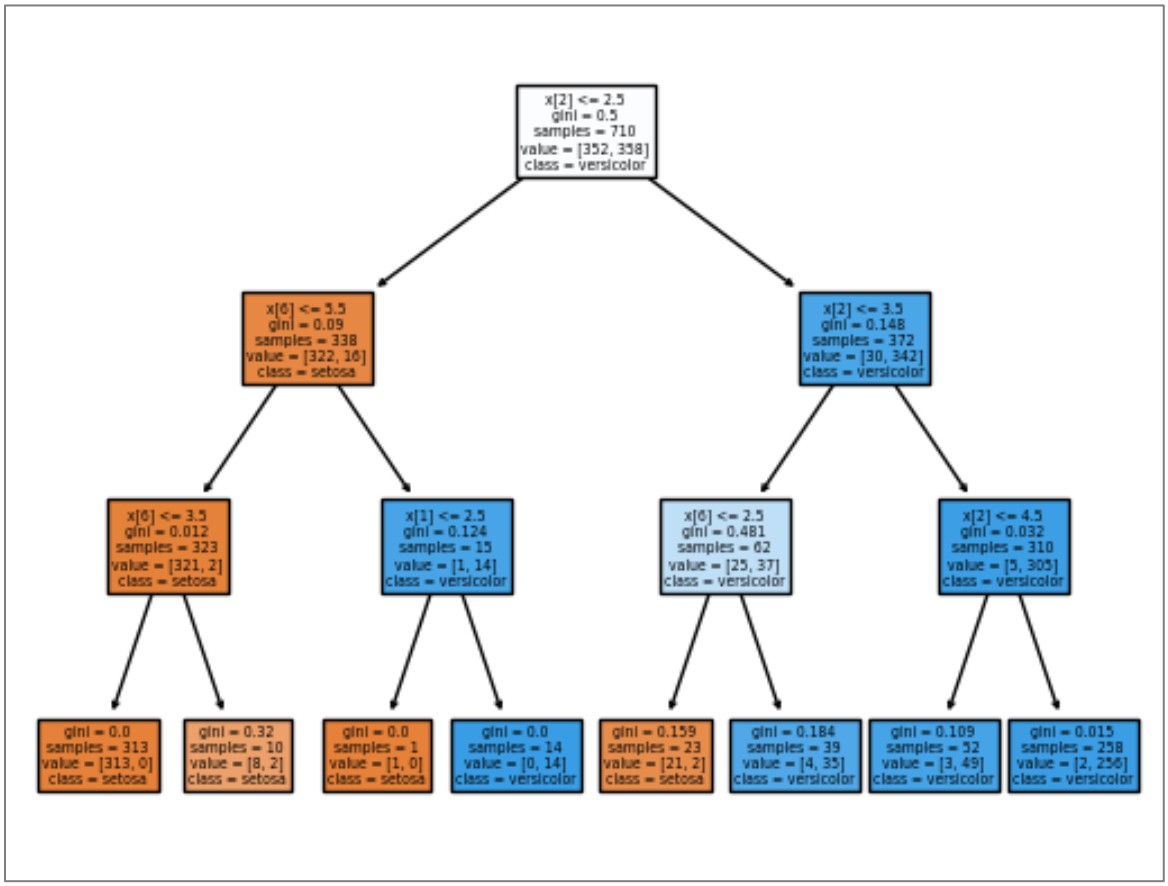
```
# Use the DecisionTreeClassifier to fit data
clf = DecisionTreeClassifier(max_depth=3 ,random_state=0)
clf.fit(x_train_sm, y_train_sm)
```

]:

```
DecisionTreeClassifier(max_depth=3, random_state=0)
```

```
#Predict y data with classifier:
y_pred_dtc = clf.predict(x_test_sm)
```

```
# Plot the decision tree

cn=['setosa', 'versicolor', 'virginica']
fig, axes = plt.subplots(nrows = 1,ncols = 1)
tree.plot_tree(clf,
               class_names=cn,
               filled = True);
fig.savefig('dtc.png')
```

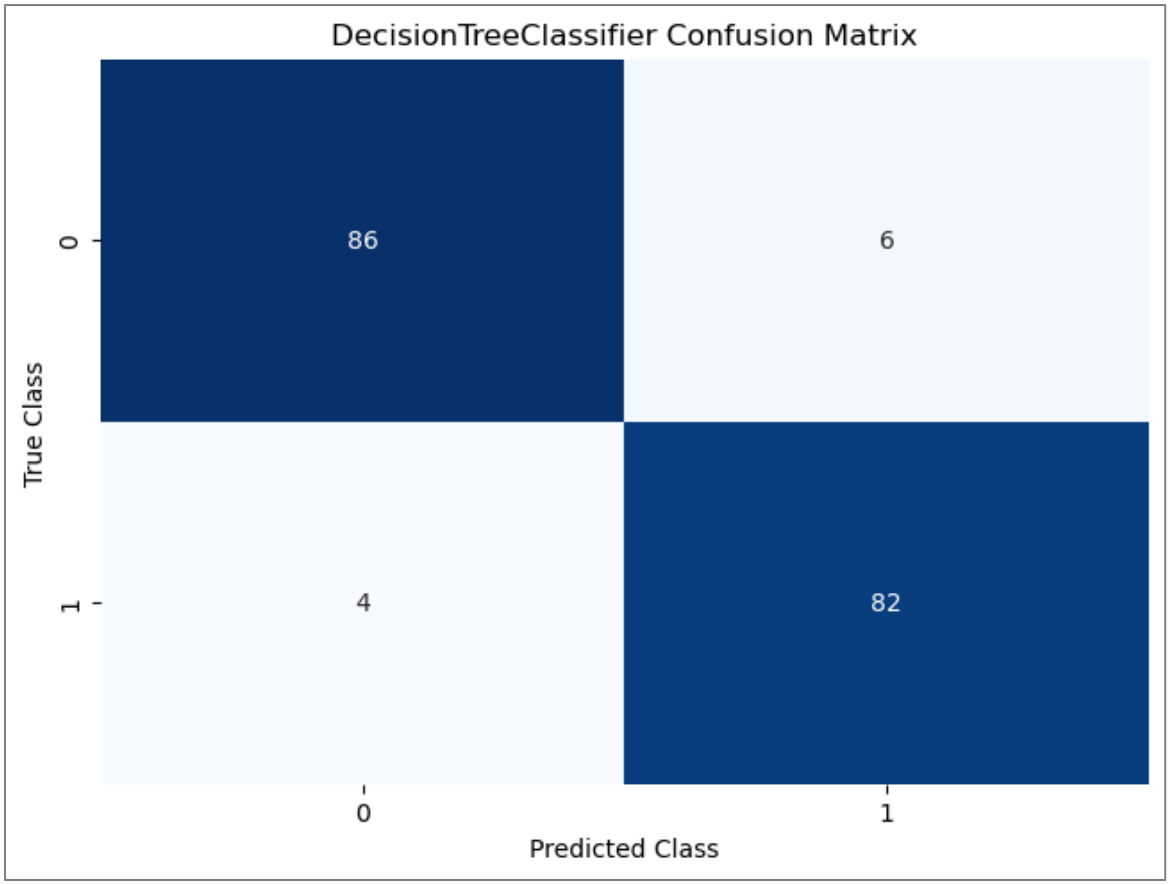


Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

BREAST CANCER PREDICTION

```
#Build the confusion matrix
matrix_sm = confusion_matrix(y_test_sm, y_pred_dtc)

print(matrix_sm)

# Create pandas dataframe
df_sm = pd.DataFrame(matrix_sm)

# Create a heatmap
sns.heatmap(df_sm, annot=True, cbar=None, cmap="Blues",fmt='.0f')
plt.title("DecisionTreeClassifier Confusion Matrix"), plt.tight_layout()
plt.ylabel("True Class"), plt.xlabel("Predicted Class")
plt.show()
```

```
[[86  6]
 [ 4 82]]
```



Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

```
#Print results
print(classification_report(y_test_sm, y_pred_dtc))
print(confusion_matrix(y_test_sm, y_pred_dtc))
print(f'ROC-AUC score : {roc_auc_score(y_test_sm, y_pred_dtc)}')
print(f'Accuracy score : {accuracy_score(y_test_sm, y_pred_dtc)}')
```

```
              precision    recall  f1-score   support

           2       0.96      0.93      0.95        92
           4       0.93      0.95      0.94        86

    accuracy                           0.94       178
   macro avg       0.94      0.94      0.94       178
weighted avg       0.94      0.94      0.94       178

[[86  6]
 [ 4 82]]
ROC-AUC score : 0.9441354903943378
Accuracy score : 0.9438202247191011
```

The model performs reasonably well for both classes, with relatively high precision, recall, and F1-score values. It correctly identifies around 96% of benign cases (class 2) and 93% of malignant cases (class 4). The confusion matrix indicates that there were 6 false positive predictions and 4 false negative predictions. While the model is performing well overall, these misclassifications should be considered in the context of the application. The ROC-AUC score of 0.944 and accuracy score of 0.944 demonstrate the model's effectiveness in distinguishing between benign and malignant cases.

KNN - KNeighbors Classifier

Initially, employing the default parameters for the KNeighborsClassifier led to a modest accuracy of approximately 60%. However, by employing GridSearchCV to determine the optimal value for the n_neighbors parameter, a substantial enhancement was achieved. Specifically, upon adopting the optimal n_neighbors value of 5, the model's accuracy experienced a notable surge to 95.5%.

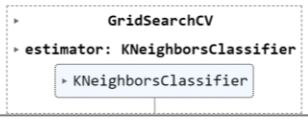
BREAST CANCER PREDICTION

KNeighborsClassifier

```
k_range = list(range(1, 31))
weight_options = ['uniform', 'distance']
metric_options=['minkowski','euclidean','manhattan','hamming']
param_grid = dict(n_neighbors=k_range, weights=weight_options, metric=metric_options)
print(param_grid)

knn = KNeighborsClassifier()

grid = GridSearchCV(knn, param_grid, cv=10, scoring='accuracy', return_train_score=False)
grid.fit(x, y)
```

```
{'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30], 'weights': ['uniform', 'distance'], 'metric': ['minkowski', 'euclidean', 'manhattan', 'hamming']}
```

6]:




```
print(grid.best_score_)
print(grid.best_params_)
```

```
0.9685137895812053
{'metric': 'hamming', 'n_neighbors': 5, 'weights': 'uniform'}
```

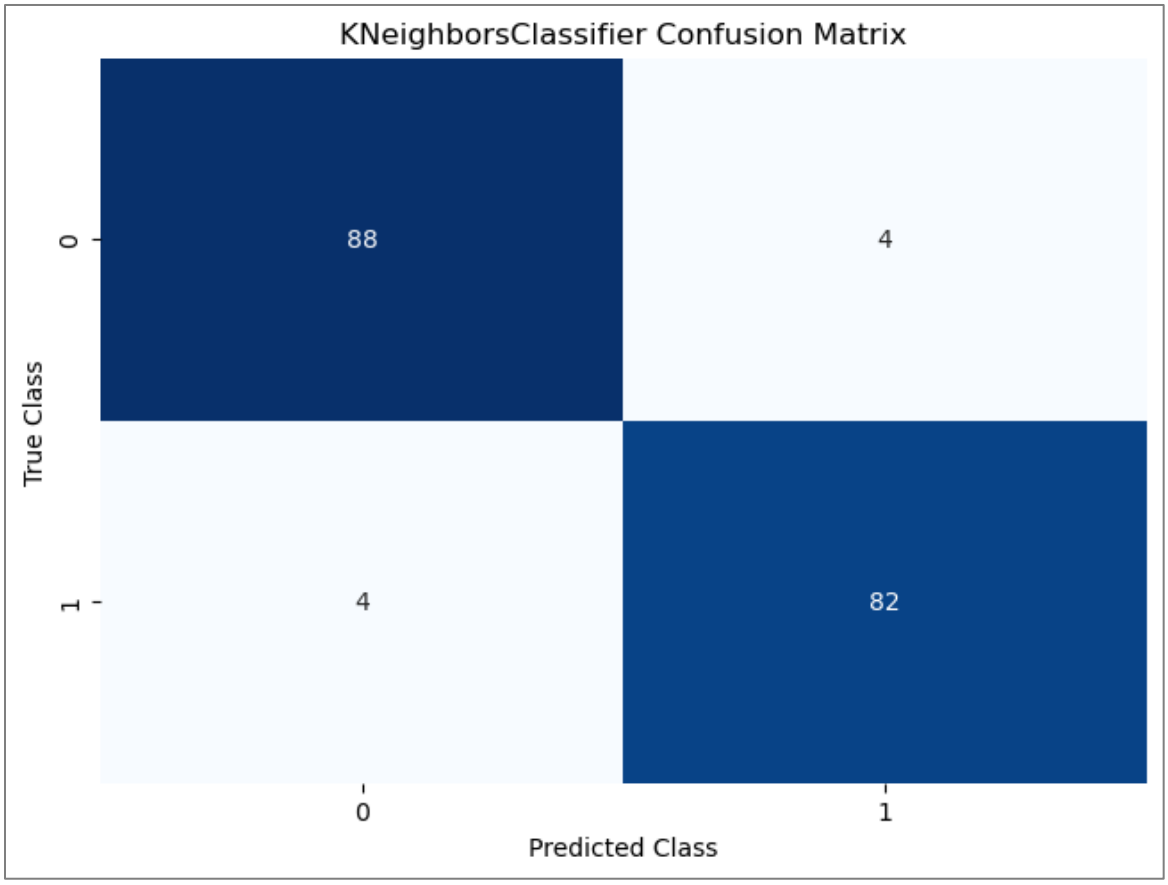
```
# Using n_neighbors = 5 for best model performance
neighbors = KNeighborsClassifier(n_neighbors=5, weights='uniform',metric= 'hamming')
neighbors.fit(x_train_sm, y_train_sm)
y_pred_knn = neighbors.predict(x_test_sm)
```

```
#Build the confusion matrix
matrix_sm = confusion_matrix(y_test_sm, y_pred_knn)

print(matrix_sm)

# Create pandas dataframe
df_sm = pd.DataFrame(matrix_sm)

# Create a heatmap
sns.heatmap(df_sm, annot=True, cbar=None, cmap="Blues",fmt='.0f')
plt.title("KNeighborsClassifier Confusion Matrix"), plt.tight_layout()
plt.ylabel("True Class"), plt.xlabel("Predicted Class")
plt.show()
```

```
[[88  4]
 [ 4 82]]
```

Date: 08/04/2023
Aarti Ramani
DSC630-T301 Predictive Analytics (2237-1)

BREAST CANCER PREDICTION




```
print(classification_report(y_test_sm, y_pred_knn))
print(confusion_matrix(y_test_sm, y_pred_knn))
print(f'ROC-AUC score : {roc_auc_score(y_test_sm, y_pred_knn)}')
print(f'Accuracy score : {accuracy_score(y_test_sm, y_pred_knn)}')
```

```
              precision    recall  f1-score   support

           2       0.96      0.96      0.96        92
           4       0.95      0.95      0.95        86

    accuracy                           0.96       178
   macro avg       0.96      0.96      0.96       178
weighted avg       0.96      0.96      0.96       178

[[88  4]
 [ 4 82]]
ROC-AUC score : 0.955005055611729
Accuracy score : 0.9550561797752809
```

The KNN model performs well for both classes, with high precision, recall, and F1-score values. It correctly identifies around 96% of benign cases (class 2) and 95% of malignant cases (class 4). The confusion matrix indicates that there were 4 false positive predictions and 4 false negative predictions. The ROC-AUC score of 0.955 and accuracy score of

0.955 demonstrate the model's effectiveness in distinguishing between benign and malignant cases.

Interpreting the Results

Considering we are dealing with a small dataset with not many features, the initial assumption was that KNN would have better accuracy over the Random Forest model. But as we can see, the Random Forest model performs better than the KNN with a 96.6% accuracy. This outcome emphasizes the importance of empirical evaluation and testing assumptions.

The Random Forest Classifier exhibited the highest accuracy of 96.6%, showcasing its proficiency in distinguishing between benign and malignant cases. It demonstrated robust precision, recall, and F1-scores for both classes, underscoring its overall effectiveness. In contrast, the Decision Tree achieved an accuracy of 94%, displaying slightly lower performance compared to the Random Forest. The KNN model, after fine-tuning its parameters, achieved an accuracy of 95.5%.

This outcome makes us rethink assumptions and shows how complex factors affect models. Random Forest's success suggests it's great for breast cancer prediction, challenging our original idea.

Initial Conclusion and Recommendation

Conclusion

The primary objective of this project is to analyze the breast cancer dataset and develop a predictive model for breast cancer. By constructing and comparing different models, we aim to identify the most effective approach for this prediction task. We implemented three models:

RandomForest, DecisionTree, and KNN, in order to determine the optimal model for breast cancer prediction.

The ROC-AUC score and accuracy score of the models demonstrate the model's effectiveness in distinguishing between benign and malignant cases. All three models perform well, with accuracy scores in the range of 94 to 97%, indicating strong predictive capabilities. The Random Forest and KNN models show slightly higher precision, recall, and F1-scores compared to the Decision Tree. The ROC-AUC scores for all models are above 94, indicating their capacity to discriminate between classes.

Overall, each model demonstrates effectiveness in predicting breast cancer based on the provided metrics. The Random Forest and KNN models particularly stand out with slightly higher performance, but the Decision Tree model is also competitive.

Recommendations

I would like to build an API/model that could be used by patients to input their symptoms and be able to predict the possibility of a benign or malignant tumor.

Although the Random Forest Classifier exhibited the highest level of accuracy, it may be beneficial to conduct a more comprehensive examination of feature importance and potential overfitting. Rigorous regression testing is crucial before deploying the API/model to minimize false positives and false negatives.

Limitations

The study relies on a limited and potentially outdated dataset. A more recent and comprehensive dataset could provide a more accurate representation of current trends and factors influencing breast cancer prediction. Additionally, the dataset's features might not capture all

relevant factors that contribute to breast cancer prediction. Additional clinical, genetic, or lifestyle-related features could enhance the model's accuracy.

Risk

Predictive models in the medical domain pose a risk of false positives (classifying benign as malignant) and false negatives (missing malignant cases). False positives could trigger unnecessary distress and invasive procedures, straining healthcare resources. False negatives may lead to delayed treatment, harming patient outcomes and intervention efficacy. Thorough testing is vital to minimize these risks and ensure patient well-being.

References

1. Pmotta. (2021, June 6). Breast cancer prediction.

   Kaggle https://www.kaggle.com/code/pmotta/breast-cancer-prediction/input

2. Patel, J., Patel , U., Patel, R., & Shah, P. (2019, April 28). *Breast Cancer Analysis*. https://rstudio-pubs-static.s3.amazonaws.com/491489_b86f191488ab4ed0a37e7a95c839a8f4.html