Breast Cancer Prediction

1. Introduction

Breast cancer is a prevalent form of cancer among women worldwide, with early detection being crucial for effective treatment and improved survival rates. In this project, we aim to develop a machine learning model to predict breast cancer diagnosis based on various features extracted from diagnostic images.

Objective:

The primary objective of this project is to construct and assess machine learning models for predicting breast cancer diagnosis with high reliability.

Our specific goals include:

- Develop predictive models leveraging diagnostic data from breast cancer biopsies.
- Evaluate the effectiveness of various machine learning algorithms in accurately classifying breast cancer cases.
- Offer insights into the underlying trends and patterns discerned from the diagnostic features of breast cancer data.

Statement:

The project aims to construct models for precise breast cancer diagnosis, leveraging biopsy data to distinguish between benign and malignant cases, facilitating early detection and treatment.

2. Importing Libraries

Libraries such as pandas, NumPy, matplotlib, seaborn, and sklearn are imported for data manipulation, visualization, and machine learning tasks.

3. Data Collection and Preprocessing

We started by collecting breast cancer diagnostic data from the Breast Cancer Wisconsin (Diagnostic) Data Set.

- Encoding the diagnosis labels (Malignant: 1, Benign: 0) using Label Encoder.
- Checking for missing values (none found).
- Visualizing the distribution of diagnosis labels using a count plot.
- Exploring the correlation between features and the target variable.
- Dropping redundant columns to reduce multicollinearity and feature redundancy.

- Functions used:

  → pd.read_csv ("Breast_Cancer.csv")

  → LabelEncoder()

  → df.drop()

  → df.isna().sum()

  → df.info()

  → df.fillna()

  → df.head()

  → df.describe()

4. Exploratory Data Analysis (EDA)

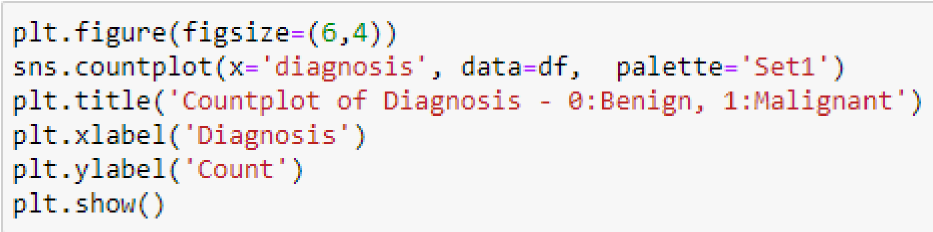
We conducted exploratory data analysis to gain insights into the relationship between features and diagnosis. This involved:

- Distribution of target values
- Boxplot visualization of feature distributions for malignant and benign cases.
- Heatmap visualization of feature correlations.
- Pairplot analysis to visualize relationships between different feature pairs.

4.1 Distribution of target values

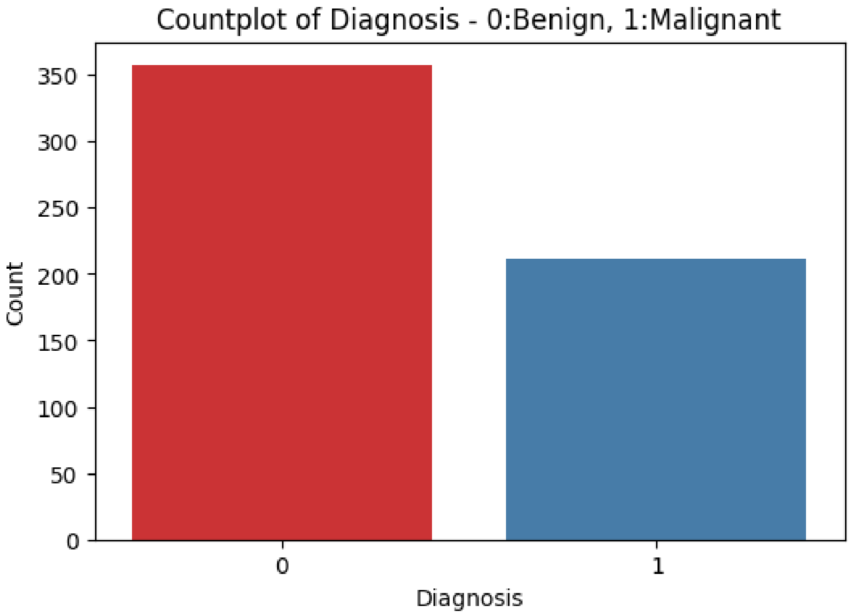
The distribution of target values (diagnosis) is visualized using a countplot.

Code:



```
plt.figure(figsize=(6,4))
sns.countplot(x='diagnosis', data=df,  palette='Set1')
plt.title('Countplot of Diagnosis - 0:Benign, 1:Malignant')
plt.xlabel('Diagnosis')
plt.ylabel('Count')
plt.show()
```

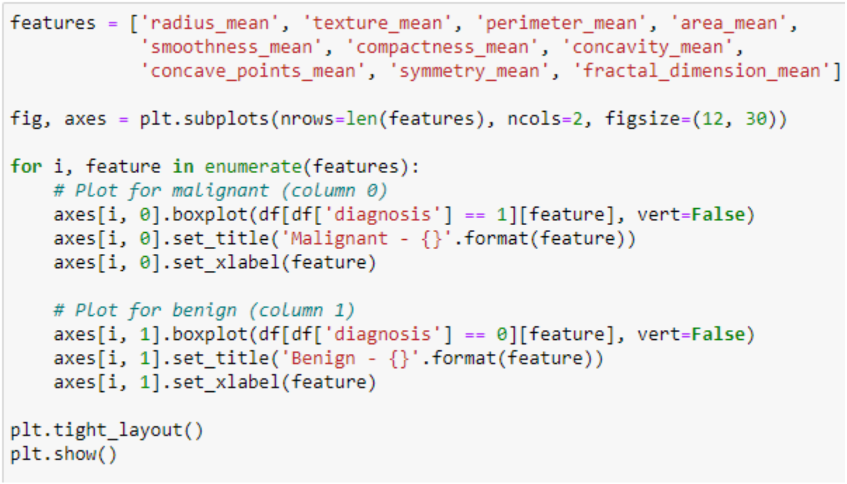
Output:



4.2 Boxplot to identify outliers

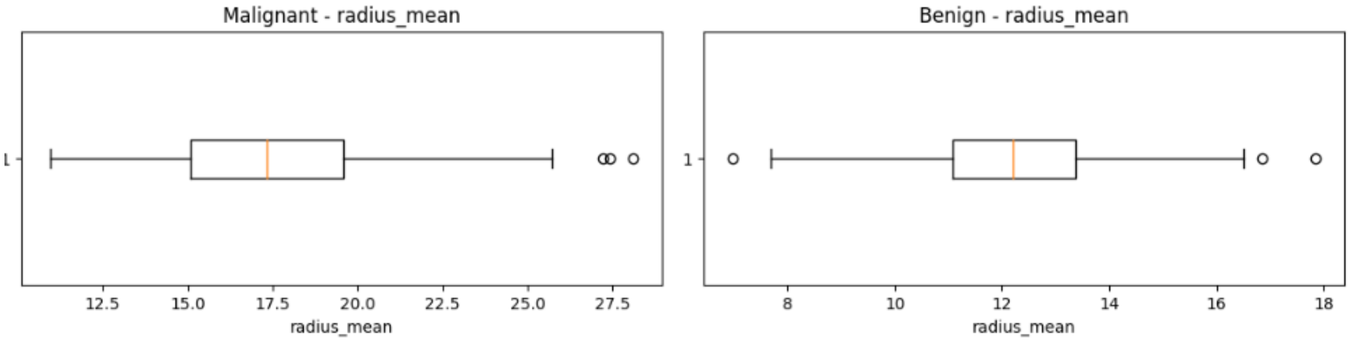
Boxplots are utilized to visualize the distribution of individual features between benign and malignant diagnoses in the breast cancer dataset.

Code:



```
features = ['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean',
            'smoothness_mean', 'compactness_mean', 'concavity_mean',
            'concave_points_mean', 'symmetry_mean', 'fractal_dimension_mean']

fig, axes = plt.subplots(nrows=len(features), ncols=2, figsize=(12, 30))

for i, feature in enumerate(features):
    # Plot for malignant (column 0)
    axes[i, 0].boxplot(df[df['diagnosis'] == 1][feature], vert=False)
    axes[i, 0].set_title('Malignant - {}'.format(feature))
    axes[i, 0].set_xlabel(feature)

    # Plot for benign (column 1)
    axes[i, 1].boxplot(df[df['diagnosis'] == 0][feature], vert=False)
    axes[i, 1].set_title('Benign - {}'.format(feature))
    axes[i, 1].set_xlabel(feature)

plt.tight_layout()
plt.show()
```

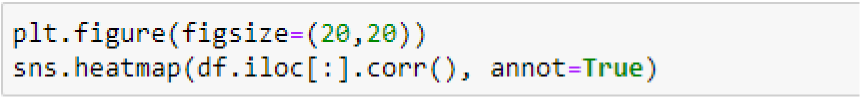
Output:



4.3 Heatmap to check multicollinearity

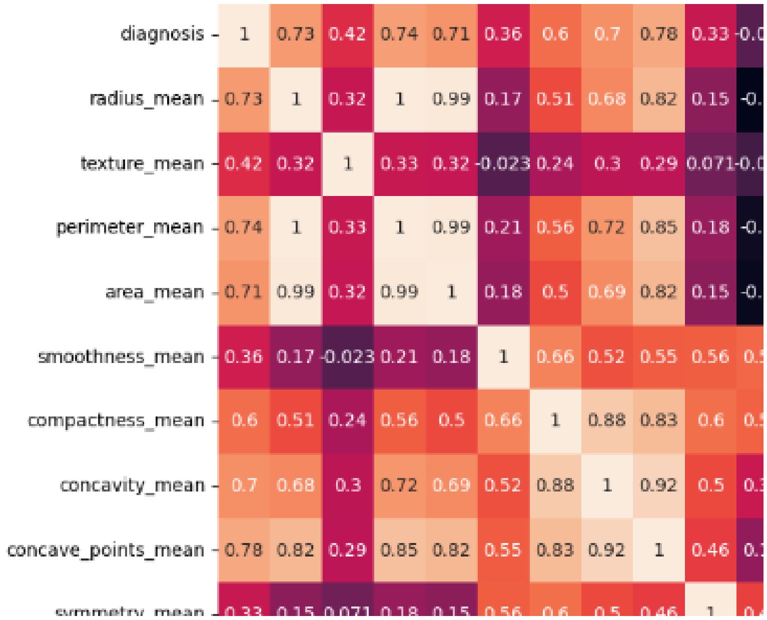
This heatmap provides a graphical representation of the pairwise correlation between different features in the dataset, helping to identify patterns and relationships between variables.

Code:



```
plt.figure(figsize=(20,20))
sns.heatmap(df.iloc[:].corr(), annot=True)
```

Output: (a small area of plot)

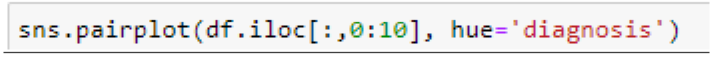


We can visualize multicollinearity present in radius, area, perimeter, compactness, concavity, concave points.

4.4 Pairplot analysis to choose model

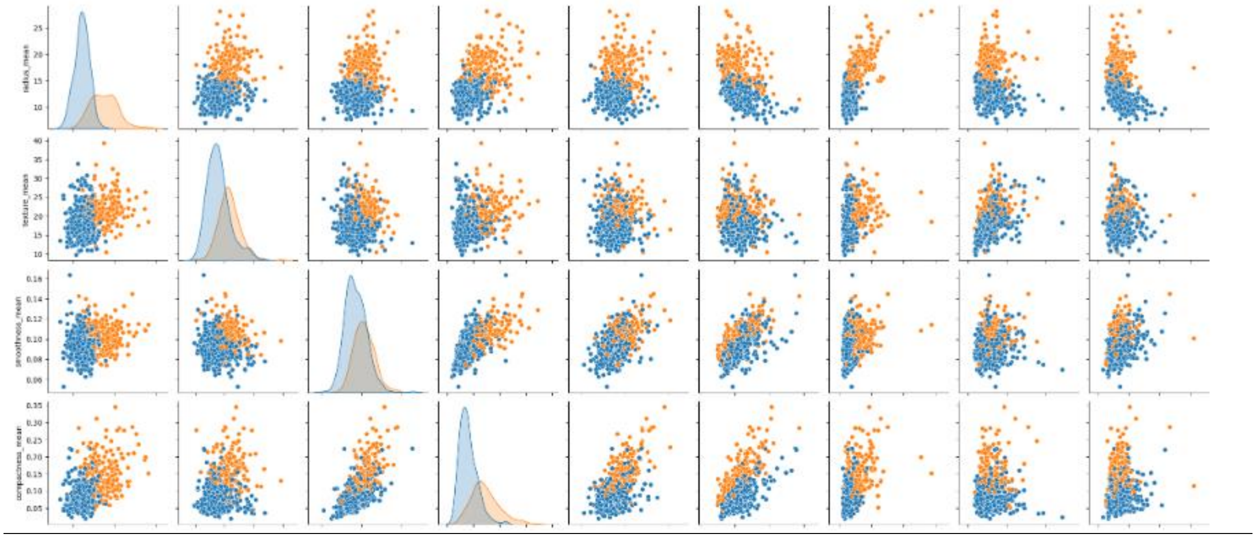
The pairplot provides a relationship between various pairs of features, so as to observe potential patterns and separability between malignant and benign tumor instances.

Code:



```
sns.pairplot(df.iloc[:,0:10], hue='diagnosis')
```

Output:



Support Vector Machines (SVM) could be chosen as they are effective in handling such separable data distributions by finding the optimal hyperplane that maximizes the margin between different classes.

5. Machine Learning Models

5.1 SVM (Support Vector Machine)

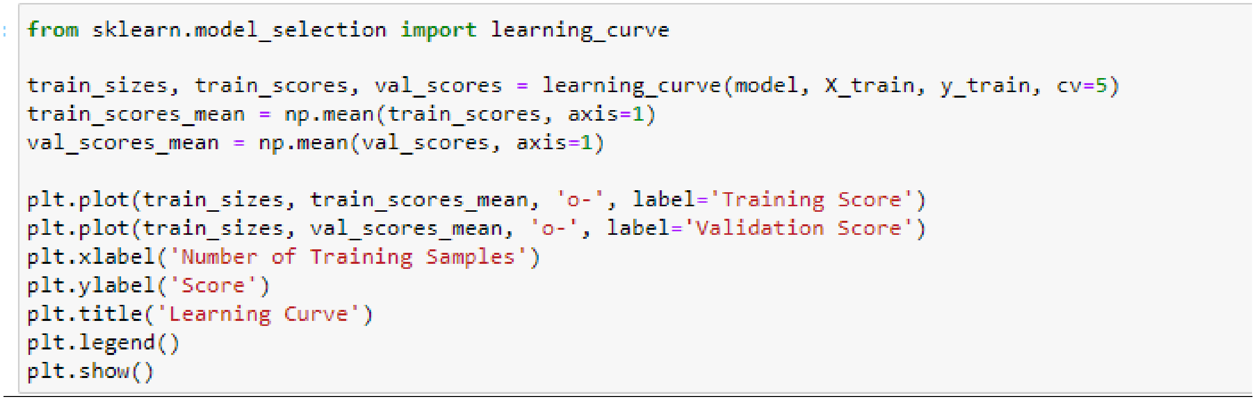
Explanation: Support Vector Machines (SVM) are a powerful class of supervised learning algorithms used for classification and regression tasks. In the context of this breast cancer prediction project, SVM is applied to classify tumor samples as benign or malignant based on various features extracted from breast cancer biopsies.

Application: SVM, chosen for its capability to handle high-dimensional data and non-linear decision boundaries, is applied to classify breast tumor samples as benign or malignant based on features from biopsies, aiming to accurately predict cancer presence.

For Breast Cancer prediction:

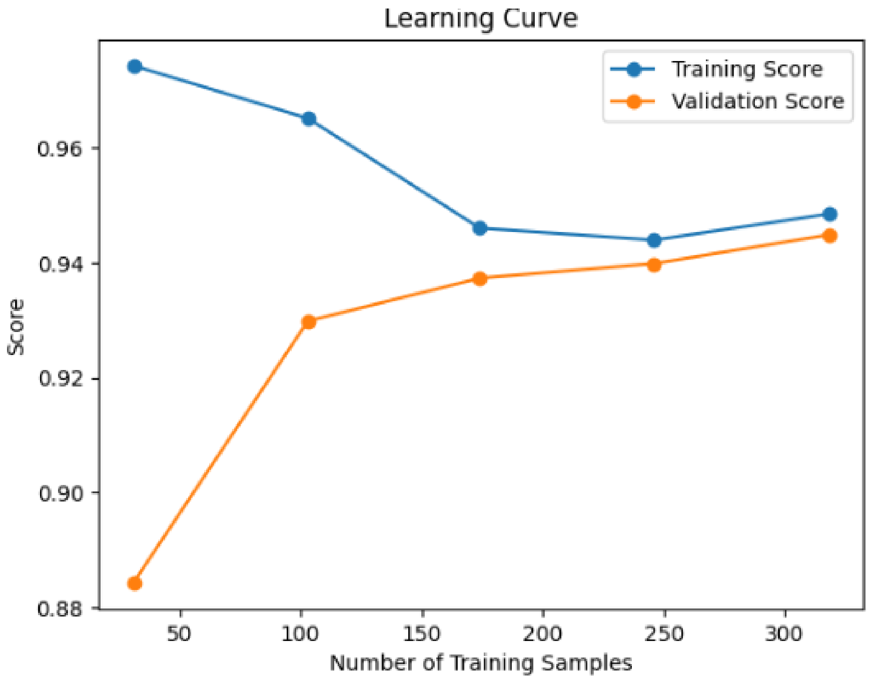
1. **Model Training and Evaluation:**
   Support Vector Machine (SVM) models are individually trained and meticulously evaluated for classifying breast tumor samples as benign or malignant based on the dataset's features.
2. **Performance Metrics Calculation:**
   Key metrics such as accuracy, precision, recall, and F1-score are computed for each SVM model. These metrics serve as quantitative measures of the model's performance in accurately classifying tumor samples.
3. **Visual Insights:**
   True vs Predicted plot, training scores vs validation scores plots are visualized.

Code:



```
from sklearn.model_selection import learning_curve

train_sizes, train_scores, val_scores = learning_curve(model, X_train, y_train, cv=5)
train_scores_mean = np.mean(train_scores, axis=1)
val_scores_mean = np.mean(val_scores, axis=1)

plt.plot(train_sizes, train_scores_mean, 'o-', label='Training Score')
plt.plot(train_sizes, val_scores_mean, 'o-', label='Validation Score')
plt.xlabel('Number of Training Samples')
plt.ylabel('Score')
plt.title('Learning Curve')
plt.legend()
plt.show()
```

Output:



6. Analysis of Model Performance

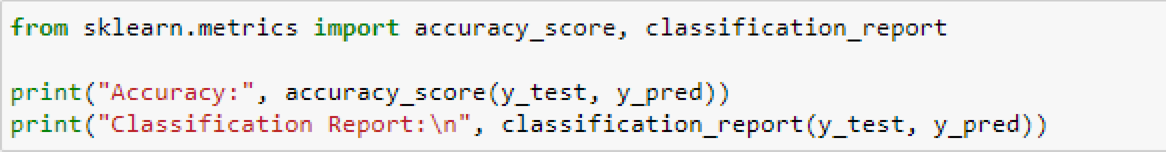
1. **Accuracy:** This metric measures the overall correctness of the model's predictions, calculated as the ratio of correctly classified samples to the total number of samples.

2. **Precision**: Precision quantifies the proportion of correctly predicted positive cases (malignant tumors) among all instances predicted as positive. It helps assess the model's ability to avoid false positives.

3. **Recall (Sensitivity)**: Recall calculates the proportion of correctly predicted positive cases among all actual positive cases. It helps evaluate the model's ability to capture all positive instances without missing any.

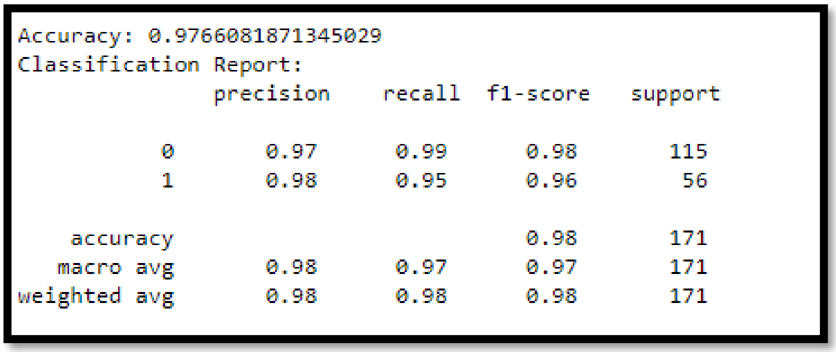
4. **F1-score**: The F1-score is the harmonic mean of precision and recall. It provides a balanced measure that considers both false positives and false negatives. F1-score is particularly useful when there is an imbalance between the classes.

Code:



```
from sklearn.metrics import accuracy_score, classification_report

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

Output:



```
Accuracy: 0.9766081871345029
Classification Report:
               precision    recall  f1-score   support

           0       0.97      0.99      0.98       115
           1       0.98      0.95      0.96        56

    accuracy                           0.98       171
   macro avg       0.98      0.97      0.97       171
weighted avg       0.98      0.98      0.98       171
```

7. Conclusion

In conclusion, the developed SVM model demonstrates promising performance in predicting breast cancer diagnosis based on diagnostic features. With an accuracy of 97.66%, the model shows high precision and recall for both benign and malignant classes.