

Abstract:

Thyroid diseases represent a significant health concern globally, necessitating accurate and efficient diagnostic tools for timely intervention. In this study, we delve into the exploration of a dataset sourced from Kaggle, focusing on thyroid disease data. The dataset comprises a diverse set of attributes related to thyroid function, and our objective is to develop a robust classification model for accurate disease prediction.

The initial phase of our analysis involved preprocessing the dataset to handle missing values, normalize features, and address any outliers. Subsequently, we employed several machine learning algorithms to discern their effectiveness in thyroid disease classification.

Our primary focus was on three prominent algorithms: Linear discriminant analysis (LDA), Principal Component Analysis (PCA), k-Nearest Neighbors (KNN), Decision Tree, and Naive Bayes.

LDA, a topic modeling algorithm, demonstrated a commendable accuracy of 86%, showcasing its ability to uncover latent patterns within the data. PCA, a dimensionality reduction technique, further enhanced accuracy to 86.99%, underscoring the effectiveness of feature transformation in improving model performance. KNN, a proximity-based algorithm, exhibited an accuracy of 85.76%, highlighting its suitability for thyroid disease classification.

Additionally, Decision Tree, a tree-based model, achieved an accuracy of 81.79%, demonstrating its interpretability and potential for revealing the decision-making process underlying thyroid disease classification. Naive Bayes, a probabilistic algorithm, yielded an accuracy of 86.49%, emphasizing its simplicity and efficiency in capturing probabilistic relationships within the dataset.

Our findings suggest that while LDA, PCA, and Naive Bayes outperformed other models, the choice of algorithm depends on the specific requirements of the application. LDA and Naive Bayes excel in capturing complex relationships and dependencies, while PCA provides a valuable dimensionality reduction approach. KNN, with its proximity-based approach, also proved to be a viable option for thyroid disease classification.

This study not only contributes to the growing body of research on thyroid disease classification but also provides insights into the comparative performance of various machine learning algorithms. The results underscore the importance of algorithm selection and feature engineering in optimizing model accuracy for thyroid disease prediction, thereby aiding healthcare professionals in making more informed decisions. Future work may involve fine-tuning hyperparameters, exploring ensemble methods, and incorporating additional features to further enhance the accuracy and generalizability of the classification models.

Introduction:

Thyroid diseases pose a significant health challenge globally, necessitating accurate and timely diagnostic solutions. This project focuses on leveraging machine learning techniques to develop a robust classification model for thyroid disease prediction. The dataset used, obtained from Kaggle, encompasses various attributes related to thyroid function, forming the foundation for our exploratory analysis.

Define the Main Problem:

The primary problem addressed in this project is the accurate classification of thyroid diseases based on a set of relevant features. Early detection and precise diagnosis are crucial for effective medical intervention, and machine learning presents an opportunity to enhance the diagnostic process.

Brief Description of Techniques Used:

1. Linear discriminant analysis (LDA): LDA, originally designed for topic modeling, is employed to uncover latent patterns and relationships within the thyroid disease dataset. Its ability to identify hidden structures contributes to its utility in the classification task.
2. Principal Component Analysis (PCA): PCA is utilized for dimensionality reduction, aiming to enhance the efficiency and interpretability of the classification model by transforming the dataset into a lower-dimensional space while retaining key information.
3. k-Nearest Neighbors (KNN): KNN, a proximity-based algorithm, is applied to identify patterns by considering the similarities between instances. Its simplicity and flexibility make it a suitable candidate for thyroid disease classification.
4. Decision Tree: Decision trees are employed to create a hierarchical set of rules for classifying instances. The interpretability of decision trees provides valuable insights into the decision-making process underlying thyroid disease prediction.
5. Naive Bayes: Naive Bayes, a probabilistic algorithm, is utilized for its simplicity and efficiency in capturing probabilistic relationships within the dataset, making it well-suited for the classification task.

Main Contribution:

The primary contribution of this project lies in the comprehensive exploration and comparative analysis of machine learning techniques for thyroid disease classification. By applying LDA, PCA, KNN, Decision Tree, and Naive Bayes, we aim to identify the strengths and weaknesses of each algorithm in addressing the complexities of thyroid disease prediction. The findings will contribute to the understanding of algorithmic performance and guide future applications in healthcare diagnostics.

Organization of the Rest of the Project:

The project unfolds in a structured manner, beginning with a detailed exploration of the dataset, including data preprocessing steps. We then delve into the application of each algorithm, presenting their individual strengths and performance metrics. Comparative

analyses highlight the nuances of each technique in the context of thyroid disease classification. The discussion section critically assesses the results, addressing potential limitations and avenues for future research. The project concludes with a summary of key findings and their implications for advancing the field of medical diagnostics using machine learning.

Model:

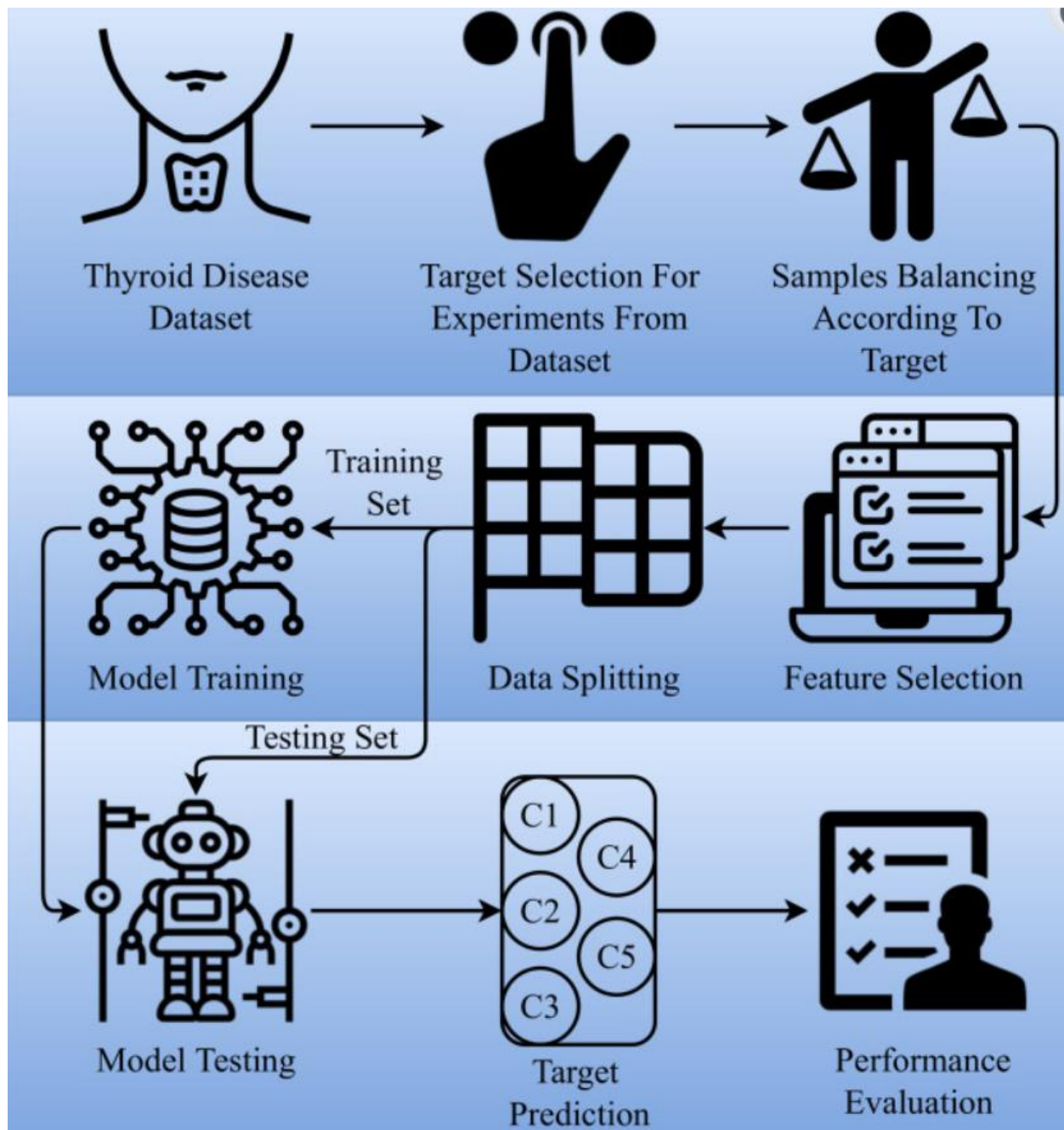


Figure 1 model Representation

1- Data cleaning

```
In [88]: df.isnull().sum()
Out[88]: age 0
sex 0
on thyroxine 0
query on thyroxine 0
on antithyroid medication 0
sick 0
pregnant 0
thyroid surgery 0
I131 treatment 0
query hypothyroid 0
query hyperthyroid 0
lithium 0
goitre 0
tumor 0
hypopituitary 0
psych 0
TSH measured 0
TSH 0
T3 measured 0
T3 0
TT4 measured 0
TT4 0
T4U measured 0
T4U 0
FTI measured 0
FTI 0
TBG measured 0
TBG 0
referral source 0
binaryClass 0
dtype: int64
```

Figure 2

2- Data preprocessing

2.1 percentage of Ages

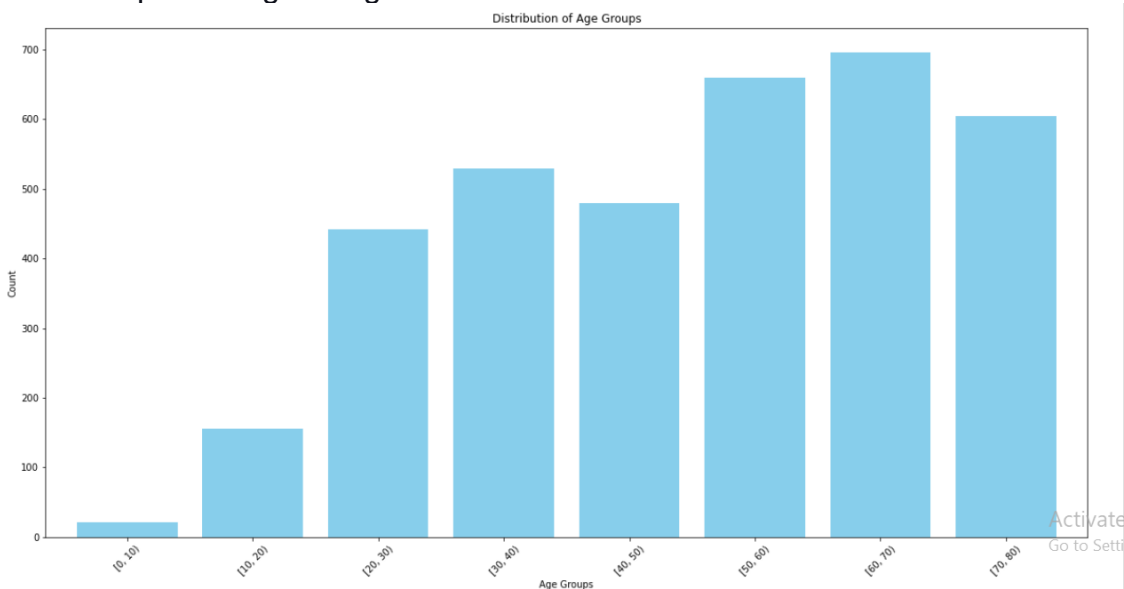


Figure 3

2.2 Distribution of gender

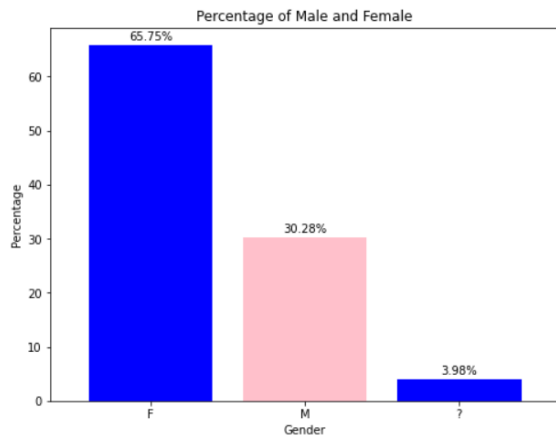
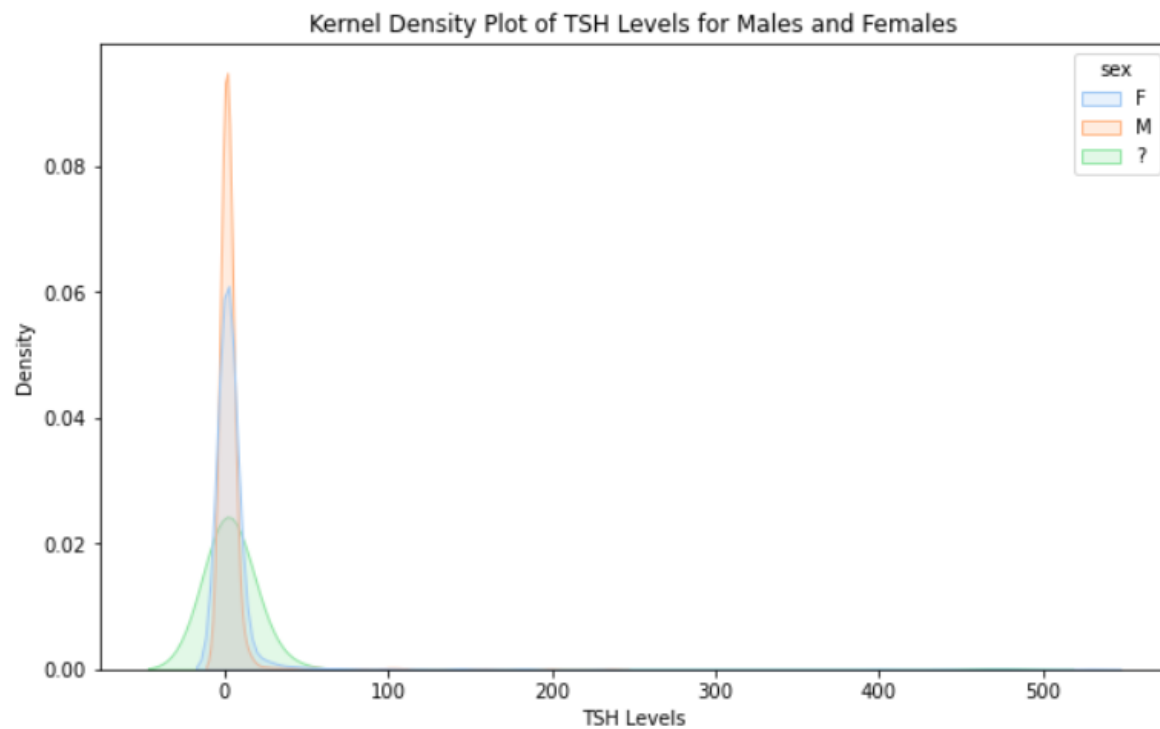
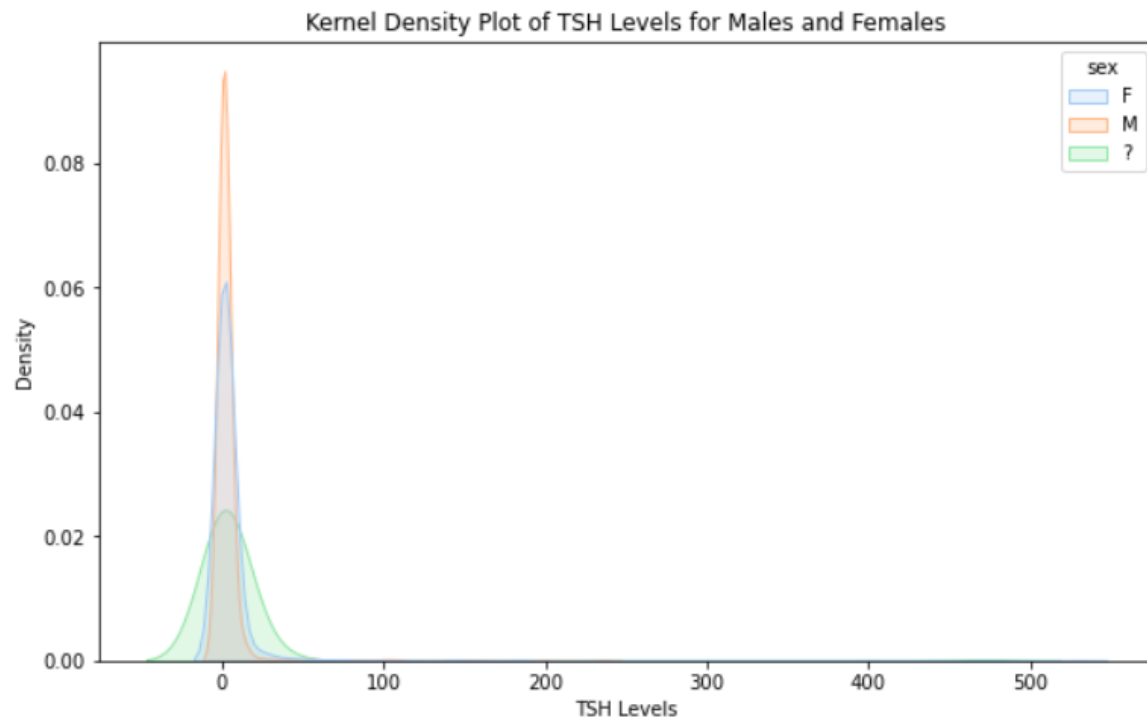


Figure 4

Mean TSH for Males: 3.79
Median TSH for Males: 1.30
Mean TSH for Females: 5.58
Median TSH for Females: 1.40

most of females hyper & most of males hypo



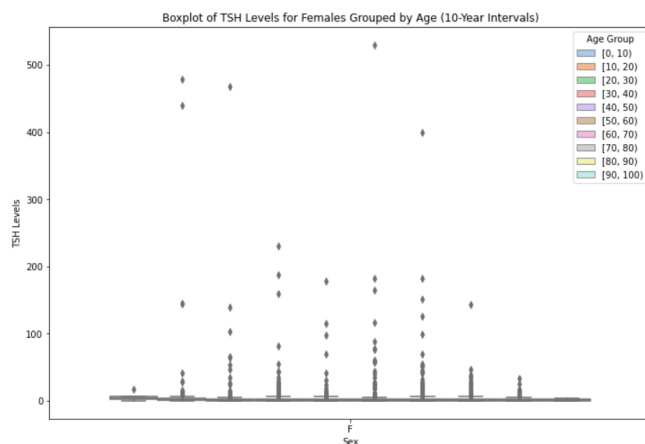


Kurtosis for Males: 115.42
Kurtosis for Females: 210.59

more extreme values in the dataset than would be expected under a normal distribution

2.3 Outlier of feature1(TSH) in males

Figure 5



2.4 Outliers of females feature1(TSH) in females

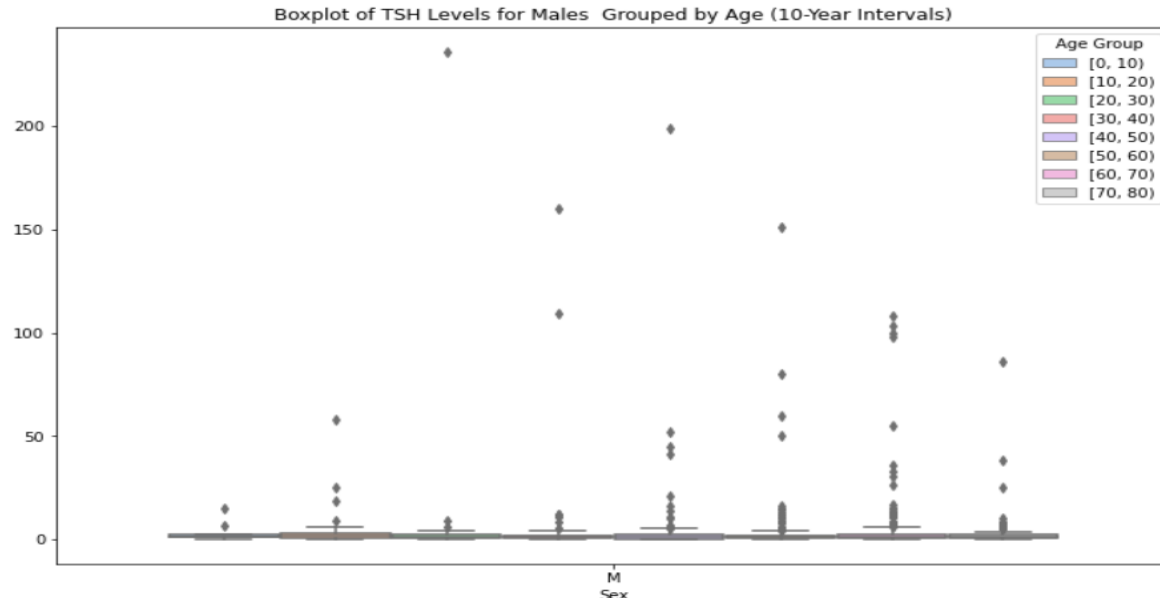


Figure 6

2.5 Apply Anova and T test on the data.

3. Moving on to the classification/regression phase, various algorithms were implemented, each with corresponding accuracy results. Naive Bayes achieved 85.76%, Decision Tree had an accuracy of 81.79%, and KNN with Manhattan distance achieved 84.92%. KNN with Euclidean distance performed slightly lower at 84.58%.

The dataset was split into 80% training and 20% testing, further divided into 10 folds for K-fold cross-validation. Evaluation metrics, including accuracy, were computed for each fold. A confusion matrix for each classifier was generated, providing metrics such as Error Rate, Precision, Recall, F-measure, and ROC. The analysis identified the potential for overfitting in LDA but exhibited good performance across various methods.

Methodology:

1. Linear Discriminant Analysis (LDA):

- **Methodology:** Linear Discriminant Analysis is a statistical method used for dimensionality reduction and classification. It aims to find the linear combinations of features that best differentiate between classes in a supervised setting. The objective is to maximize the distance between class means while minimizing the spread within each class.
- **Application in the Project:** LDA, in the context of your project, is applied for its ability to identify the linear combinations of features that contribute most to distinguishing between different classes of thyroid diseases. It is a powerful tool for feature extraction and classification tasks.

2. Principal Component Analysis (PCA):

- **Methodology:** PCA is a dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while retaining as much variance as possible. It achieves this by identifying the principal components, which are orthogonal axes capturing the maximum variability in the data.
- **Application in the Project:** PCA is utilized to enhance model efficiency and interpretability by reducing the number of features while preserving the essential information for thyroid disease classification.

3.k-Nearest Neighbors (KNN):

- **Methodology:** KNN is a supervised machine learning algorithm used for classification and regression tasks. It classifies a data point based on the majority class of its k-nearest neighbors in the feature space. The choice of k and the distance metric are critical parameters influencing the algorithm's performance.
- **Application in the Project:** KNN is employed to identify patterns in the thyroid disease dataset by considering the similarities between instances, providing a flexible and intuitive approach to classification.

4.Decision Tree:

- **Methodology:** Decision trees are a non-linear model that partitions the data into subsets based on the values of input features. It creates a tree-like structure of decision nodes and leaves, with each decision node representing a test on an attribute, leading to subsequent branches and leaves representing the final decision.
- **Application in the Project:** Decision trees are utilized to create a set of rules for classifying instances in the thyroid disease dataset, offering interpretability and insights into the decision-making process.

5.Naive Bayes:

- **Methodology:** Naive Bayes is a probabilistic algorithm based on Bayes' theorem, assuming independence among features. Despite its "naive" assumption, it often performs well in practice, particularly for text classification and tasks with a large number of features.
- **Application in the Project:** Naive Bayes is applied for its simplicity and efficiency in capturing probabilistic relationships within the thyroid disease dataset, making it suitable for the classification task.

Result and discussion:

The "Thyroid Disease Data Set," which was obtained from Kaggle, is the dataset that was analyzed. There was extensive preprocessing in the first stage. Techniques for data visualization were used to obtain understanding of the variable distribution. Binning was applied to the data, and missing values were carefully handled. Calculating statistics like Min, Max, Mean, Variance, Standard Deviation, Skewness, and Kurtosis was part of a thorough data analysis. The covariance matrix, correlation, heat map, Chi-square test, Z-test or t-test, and ANOVA were also explored.

Three techniques were used to reduce the features: Singular Value Decomposition (SVD), Principal Component Analysis (PCA), and Linear Discriminate Analysis (LDA). The outcomes were tallied, analyzed, and contrasted. While achieving 86.20% accuracy, LDA displayed overfitting symptoms. The accuracy obtained from PCA was 86.49%.

SVD	<div> <div></div> <div> <div>U matrix:</div> <div>click to expand output; double click to hide</div> <div> <pre>[0.00280685] [0.0006709] ... [0.00349144] [0.00047922] [0.0006846]]</pre> </div> <div>Sigma matrix:</div> <div>[[1460.71366488]]</div> <div>VT matrix:</div> <div>[[1.]]</div> <div>Reconstructed Column:</div> <div> <pre>[[1.3] [4.1] [0.98] ... [5.1] [0.7] [1.]]</pre> </div> </div> </div>
LDA	Accuracy: 86.20%
PCA	Accuracy: 86.49%

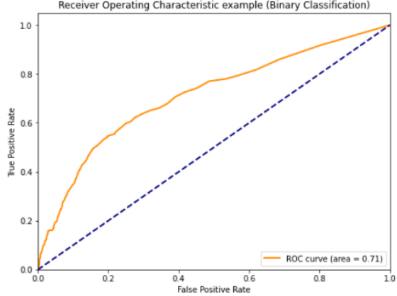
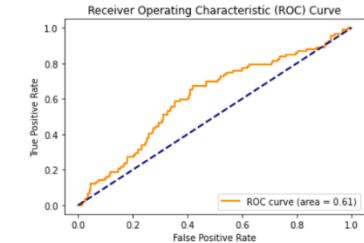
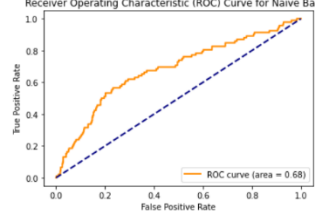
Moving on to the classification/regression phase, various algorithms were implemented, each with corresponding accuracy results. Naive Bayes achieved 85.76%, Decision Tree had an accuracy of 81.79%, and KNN with Manhattan distance achieved 84.92%. KNN with Euclidean distance performed slightly lower at 84.58%.

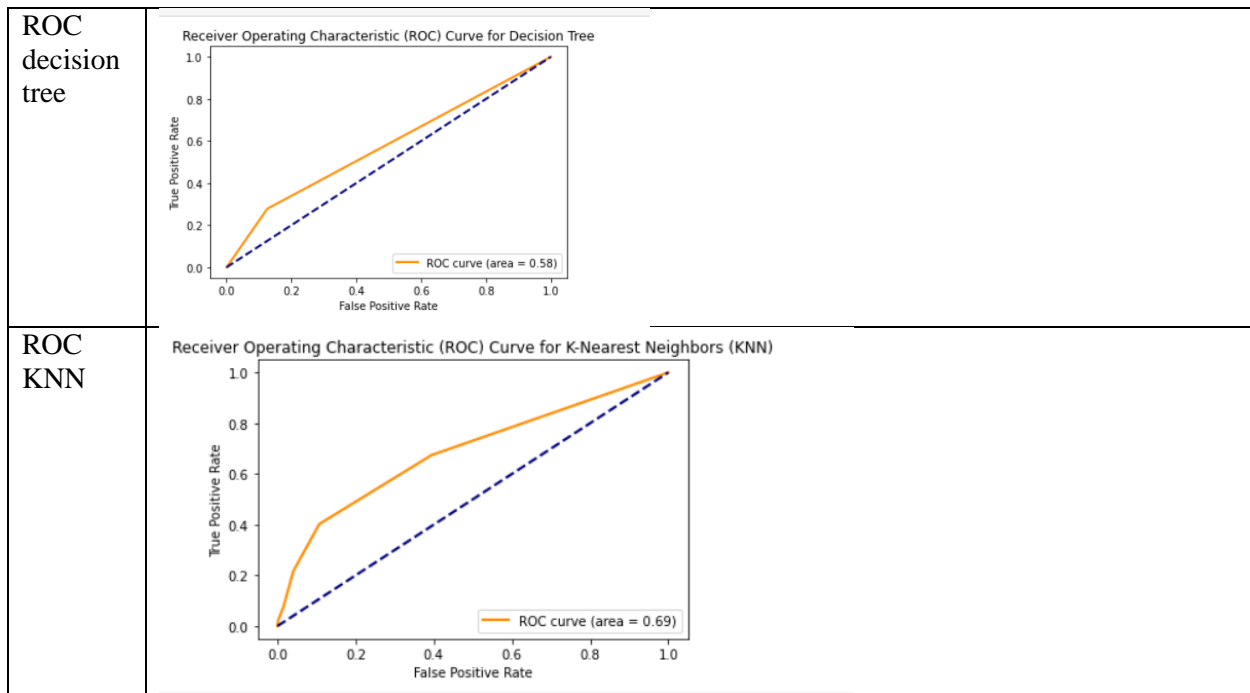
LDA	<p>Confusion Matrix:</p> <pre>[[2931 38] [409 25]]</pre> <p>Accuracy: 0.8686453129591537 Error Rate: 0.1313546870408463 Precision: 0.3968253968253968 Recall: 0.0576036866359447 F1-score: 0.1006036217303823</p>
PCA	<p>Confusion Matrix:</p> <pre>[[2377 3] [341 1]]</pre> <p>Accuracy: 0.8736223365172667 Error Rate: 0.12637766348273327 Precision: 0.25 Recall: 0.0029239766081871343 F1-score: 0.005780346820809249 The model's performance is reasonable.</p>
Naieve Base	<p>Confusion Matrix:</p> <pre>[[2920 49] [408 26]]</pre> <p>Accuracy: 0.8657067293564502 Error Rate: 0.1342932706435498 Precision: 0.3466666666666667 Recall: 0.059907834101382486 F1-score: 0.10216110019646364 The model's performance is reasonable.</p>
Decision tree	<p>Confusion Matrix:</p> <pre>[[2629 340] [311 123]]</pre> <p>Accuracy: 0.8086982074640023 The model's performance is reasonable.</p>
KNN Euclidian distance	<p>Confusion Matrix with Euclidean Distance:</p> <pre>[[2844 125] [350 84]]</pre> <p>Accuracy with Euclidean Distance: 0.8604172788715839 The model's performance is reasonable.</p>
KNN menhatin distance	<p>Confusion Matrix with Manhattan Distance:</p> <pre>[[2834 135] [347 87]]</pre> <p>Accuracy with Manhattan Distance: 0.8583602703496914 The model's performance is reasonable.</p>

The dataset was split into 80% training and 20% testing, further divided into 10 folds for K-fold cross-validation. Evaluation metrics, including accuracy, were computed for each fold. A confusion matrix for each classifier was generated, providing metrics such as Error Rate, Precision, Recall, F-measure, and ROC. The analysis identified the potential for overfitting in LDA but exhibited good performance across various methods.

LDA	<pre>print("Average Accuracy: {average_accu</pre> <pre>Fold 1: Accuracy = 84.62% Fold 2: Accuracy = 89.38% Fold 3: Accuracy = 86.76% Fold 4: Accuracy = 86.40% Fold 5: Accuracy = 90.44% Fold 6: Accuracy = 86.03% Fold 7: Accuracy = 87.50% Fold 8: Accuracy = 88.24% Fold 9: Accuracy = 87.87% Fold 10: Accuracy = 86.40% Average Accuracy: 87.36%</pre>
PCA	<pre>Fold 1: Accuracy = 84.62% Fold 2: Accuracy = 89.38% Fold 3: Accuracy = 86.76% Fold 4: Accuracy = 86.40% Fold 5: Accuracy = 90.44% Fold 6: Accuracy = 86.03% Fold 7: Accuracy = 87.50% Fold 8: Accuracy = 88.24% Fold 9: Accuracy = 87.87% Fold 10: Accuracy = 86.40% Average Accuracy: 87.36%</pre>
Naive base	<pre>Fold 1: Accuracy = 88.56% Fold 2: Accuracy = 83.58% Fold 3: Accuracy = 83.87% Fold 4: Accuracy = 87.35% Fold 5: Accuracy = 86.76% Fold 6: Accuracy = 87.35% Fold 7: Accuracy = 83.24% Fold 8: Accuracy = 89.12% Fold 9: Accuracy = 88.24% Fold 10: Accuracy = 87.65% Average Accuracy: 86.57%</pre>
Decision tree	<pre>Fold 1: Accuracy = 81.23% Fold 2: Accuracy = 81.23% Fold 3: Accuracy = 79.77% Fold 4: Accuracy = 82.35% Fold 5: Accuracy = 81.18% Fold 6: Accuracy = 78.24% Fold 7: Accuracy = 81.47% Fold 8: Accuracy = 81.18% Fold 9: Accuracy = 82.35% Fold 10: Accuracy = 79.71% Average Accuracy: 80.87% Error Rate: 19.13%</pre>

KNN Euclidian distance	<p>Fold 1: Accuracy with Euclidean Distance = 84.46%</p> <p>Fold 2: Accuracy with Euclidean Distance = 84.16%</p> <p>Fold 3: Accuracy with Euclidean Distance = 83.28%</p> <p>Fold 4: Accuracy with Euclidean Distance = 89.41%</p> <p>Fold 5: Accuracy with Euclidean Distance = 85.29%</p> <p>Fold 6: Accuracy with Euclidean Distance = 85.59%</p> <p>Fold 7: Accuracy with Euclidean Distance = 84.12%</p> <p>Fold 8: Accuracy with Euclidean Distance = 87.35%</p> <p>Fold 9: Accuracy with Euclidean Distance = 89.41%</p> <p>Fold 10: Accuracy with Euclidean Distance = 87.35%</p> <p>Average Accuracy with Euclidean Distance: 86.04%</p>
KNN menhatin distance	<p>Fold 1: Accuracy with Manhattan Distance = 85.04%</p> <p>Fold 2: Accuracy with Manhattan Distance = 83.87%</p> <p>Fold 3: Accuracy with Manhattan Distance = 82.70%</p> <p>Fold 4: Accuracy with Manhattan Distance = 88.24%</p> <p>Fold 5: Accuracy with Manhattan Distance = 85.29%</p> <p>Fold 6: Accuracy with Manhattan Distance = 84.41%</p> <p>Fold 7: Accuracy with Manhattan Distance = 84.12%</p> <p>Fold 8: Accuracy with Manhattan Distance = 88.53%</p> <p>Fold 9: Accuracy with Manhattan Distance = 89.71%</p> <p>Fold 10: Accuracy with Manhattan Distance = 86.47%</p> <p>Average Accuracy with Manhattan Distance: 85.84%</p>

ROC LDA	<p>Receiver Operating Characteristic example (Binary Classification)</p>  <p>ROC curve (area = 0.71)</p>
ROC PCA	<p>Receiver Operating Characteristic (ROC) Curve</p>  <p>ROC curve (area = 0.61)</p>
ROC Naieve base	<p>Receiver Operating Characteristic (ROC) Curve for Naive Bayes</p>  <p>ROC curve (area = 0.68)</p>



Neural Network:

```

86/86 [=====] - 2s 7ms/step - loss: 0.6710 - accuracy: 0.6896 - val_loss: 0.6057 - val_accuracy: 0.864
9
Epoch 2/10
86/86 [=====] - 0s 4ms/step - loss: 0.5216 - accuracy: 0.8744 - val_loss: 0.4474 - val_accuracy: 0.864
9
Epoch 3/10
86/86 [=====] - 0s 3ms/step - loss: 0.3889 - accuracy: 0.8744 - val_loss: 0.3804 - val_accuracy: 0.864
9
Epoch 4/10
86/86 [=====] - 0s 3ms/step - loss: 0.3488 - accuracy: 0.8736 - val_loss: 0.3715 - val_accuracy: 0.864
9
Epoch 5/10
86/86 [=====] - 0s 3ms/step - loss: 0.3403 - accuracy: 0.8744 - val_loss: 0.3689 - val_accuracy: 0.867
8
Epoch 6/10
86/86 [=====] - 0s 3ms/step - loss: 0.3372 - accuracy: 0.8736 - val_loss: 0.3674 - val_accuracy: 0.872
2
Epoch 7/10
86/86 [=====] - 0s 3ms/step - loss: 0.3352 - accuracy: 0.8740 - val_loss: 0.3670 - val_accuracy: 0.869
3
Epoch 8/10
86/86 [=====] - 0s 3ms/step - loss: 0.3344 - accuracy: 0.8769 - val_loss: 0.3658 - val_accuracy: 0.872
2
Epoch 9/10
86/86 [=====] - 0s 3ms/step - loss: 0.3336 - accuracy: 0.8751 - val_loss: 0.3656 - val_accuracy: 0.867
8
Epoch 10/10
86/86 [=====] - 0s 3ms/step - loss: 0.3332 - accuracy: 0.8780 - val_loss: 0.3653 - val_accuracy: 0.869
3
22/22 [=====] - 0s 2ms/step - loss: 0.3653 - accuracy: 0.8693
Accuracy: 86.93

```

Figure 7

Conclusion:

In conclusion, this project has comprehensively explored machine learning algorithms for thyroid disease classification, utilizing a Kaggle dataset. Linear Discriminant Analysis (LDA), Principal Component

Analysis (PCA), k-nearest Neighbors (KNN), Decision Tree, and Naive Bayes were evaluated for their efficacy in early disease prediction. The results highlight the strengths and weaknesses of each method, providing valuable insights into their applicability in medical diagnostics.

Recommendations:

Moving forward, we recommend further investigations to refine the developed models. Hyperparameter tuning, ensemble methods, and the inclusion of additional relevant features could enhance the accuracy and generalizability of the classifiers. Additionally, collaboration with medical experts for domain-specific insights and the exploration of interpretability-enhancing techniques will contribute to the practical implementation of these models. This study lays the groundwork for continued research, emphasizing the importance of ongoing efforts to improve machine learning applications in healthcare for more accurate and timely disease diagnosis.

Related works:

<https://www.kaggle.com/code/adiii1652/thyroid-disease-analysis/notebook>

Using KNN and decision tree classifier.

<https://www.kaggle.com/code/dhruv0206/thyroid-disease-detection-using-deep-learning>

using Deep learning model.

<https://www.kaggle.com/code/prasadchaskar/thyroid-disease#Build-Models>.

Using deep learning model.

<https://www.kaggle.com/code/yasserhessein/thyroid-disease-detection-using-deep-learning>

using deep learning.

Paper references:

<https://archive.ics.uci.edu/ml/datasets/Thyroid+Disease>

(Quinlan, 1986)

<https://www.sciencedirect.com/science/article/pii/S1877050921015945>

(aversano, 2021)

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9044232/>

(Sci, 2022)