Final Project Report Template

1. Introduction

1.1. Project overviews

The goal of this project is to develop a machine learning model that will enable the classification of thyroid disorders using patient data from a dataset.

1.2. Objectives

- Clean and prepare the dataset.
- Train and test Random Forest, XGBoost, and SVC models.
- Evaluate model performance using accuracy and other metrics.
- Select the most significant features.
- Choose the most accurate model for predicting thyroid conditions.
- Deploy the best model in a Flask web application with home, predict, and submit pages.

2.Project Initialization and Planning Phase

2.1.Define Problem Statement

Define Problem Statements (Customer Problem Statement Template):

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love. A well-articulated customer problem statement allows you and your team to find the ideal solution for your customers' challenges. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive your product or service.

I am	Describe customer with 3-4 key characteristics - who are they?	Describe the customer and their attributes here
I'm trying to	List their outcome or "job" the user is trying to do - what are they trying to achieve?	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way - what prevents them now?	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists - what needs to be solved?	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view - how does it impact them emotionally?	Describe the emotions the result from experiencing the problems or barriers

Reference: https://miro.com/templates/customer-problem-statement/

Example:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
symptoms of hypothyroidism	Patient	Understand why I have been feeling extremely tired and gaining weight	Despite taking thyroid medication, I still feel fatigued and notice little improvement in my weight	It takes time for thyroid hormone levels to stabilize after starting medication, and my dosage may need adjustment	Frustrated and worried about my health and the effectiveness of my treatment.
Symptoms of hyperthyroidism.	Patient	Understand why I am experiencing rapid weight loss and increased heart rate	Despite taking medication to reduce thyroid hormone levels, I still feel jittery and anxious	Achieving optimal thyroid hormone levels can be challenging, and it may take time to find the right medication dosage	Anxious and concerned about potential long-term effects on my health, despite the medication.

2. Project Proposal (Proposed Solution)

Project Proposal (Proposed Solution) template

This project proposal outlines a solution to address a specific problem. With a clear objective, defined scope, and a concise problem statement, the proposed solution details the approach, key features, and resource requirements, including hardware, software, and personnel.

Project Overview	
Objective	The main goal is to identify different types of thyroid problems based on medical information.
Scope	project will collect medical data like test results and images to build a system that can classify thyroid conditions accurately.
Problem Statement	
Description	We'll gather medical data, clean it up, and use it to train a computer system to recognize different thyroid issues, like hyperthyroidism or thyroid cancer.
Impact	project aims to help doctors make quicker and more accurate diagnoses of thyroid problems, which could improve patient care and reduce medical costs.
Proposed Solution	
Approach	We will use machine learning techniques to analyze the medical data and teach the computer system how to classify different thyroid conditions.
Key Features	Data Collection: Gather all the necessary medical information. Data Cleaning: Make sure the data is accurate and organized.

Resource Requirements

Resource Type	Description	Specification/Allocation
Hardware		
Computing Resources	CPU/GPU specifications, number of cores	e.g., 2 x NVIDIA V100 GPUs
Memory	RAM specifications	e.g., 8 GB
Storage	Disk space for data, models, and logs	e.g., 1 TB SSD
Software		
Frameworks	Python frameworks	e.g., TensorFlow, PyTorch
Libraries	Additional libraries	e.g., pandas, numpy,metaplotlib ,seaborn etc
Development Environment	IDE, version control	e.g., Google collab
Data		
Data	Thyoid classification Dataset	e.g., 9172 records

2.3 project planning template

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create a product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members	Sprint Start Date	Sprint End Date (Planned)
Sprint-1	Data collection	USN-1	Collect patient records	8	High	K.Bhanu Teja	24/07/07	24/07/09
Sprint-1	Data collection	USN-2	Gather thyroid function tests	7	High	N.Dheeraj	24/07/04	24/07/06
Sprint-2	Data preprocessing	USN-3	Clean and normalize data	7	High	K.p.s.parimala	24/07/02	24/07/04
Sprint-1	Model development	USN-4	Build initial classification model and evaluate	8	High	Sathya sathwik	24/07/05	24/07/07

3. Data Collection and Preprocessing Phase

3.1 Data Collection Plan and Raw Data Sources Identified

Section	Description
Project Overview	The goal of the project "Thyroid Classification using RandomForest, XGB Classifier, SVC Model" is to create and evaluate three machine learning models—RandomForest, XGB Classifier, and SVC—for the purpose of diagnosing thyroid problems. The project will identify the most accurate model to help healthcare providers make well-informed diagnoses and improve patient outcomes using Python in Google Colab.
Data Collection Plan	The data is collected from the Kaggle.
Raw Data Sources Identified	Essential parameters including age, sex, blood pressure, cholesterol, na_to_k, and medication are included in this dataset. The construction and assessment of machine learning models— RandomForest, XGB Classifier, and SVC—for the precise categorization of thyroid diseases will be made possible by the use of this extensive dataset.

Raw data sources template

Source Name	Description	Location/URL	Format	Size
Dataset 1	For our thyroid classification project, we analyzed a dataset with 9,172 rows and 31 columns, including attributes like age, sex, thyroid medication details, and various thyroid test results. We started by checking for null values and performing a descriptive analysis to understand the data distribution. The dataset was split into features (X) and the target variable (Y). Data types were converted, categorical values were handled, and correlations were checked. We addressed data imbalance and standardized the features using StandardScaler. We built models using RandomForestClassifier, XGBClassifier, and SVC, applying hyperparameter tuning for optimization. Our evaluation through classification reports, confusion matrices, and accuracy scores revealed that XGBClassifier achieved the highest accuracy, making it the best performing model among the three.	https://drive.google.com/file/d/1ZYd3r5GzxzeHTJcWIabAWnQCXWRJ7cf8/view?usp=sharing	CSV	5.89 KB

3.2 Data quality Template

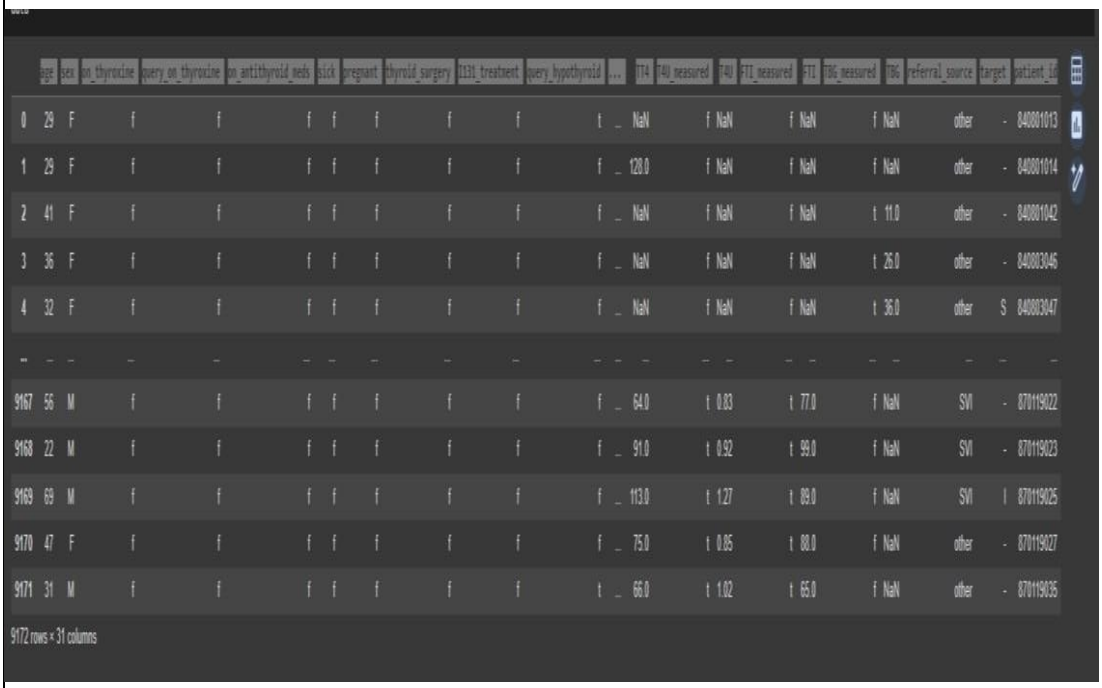
Data Source	Data Quality Issue	Severity	Resolution Plan
Dataset	The Dataset was having some null values and missing values.	Low	I have used Data preprocessing techniques and handled the missing values.

3.3 Data Exploration and Preprocessig Template

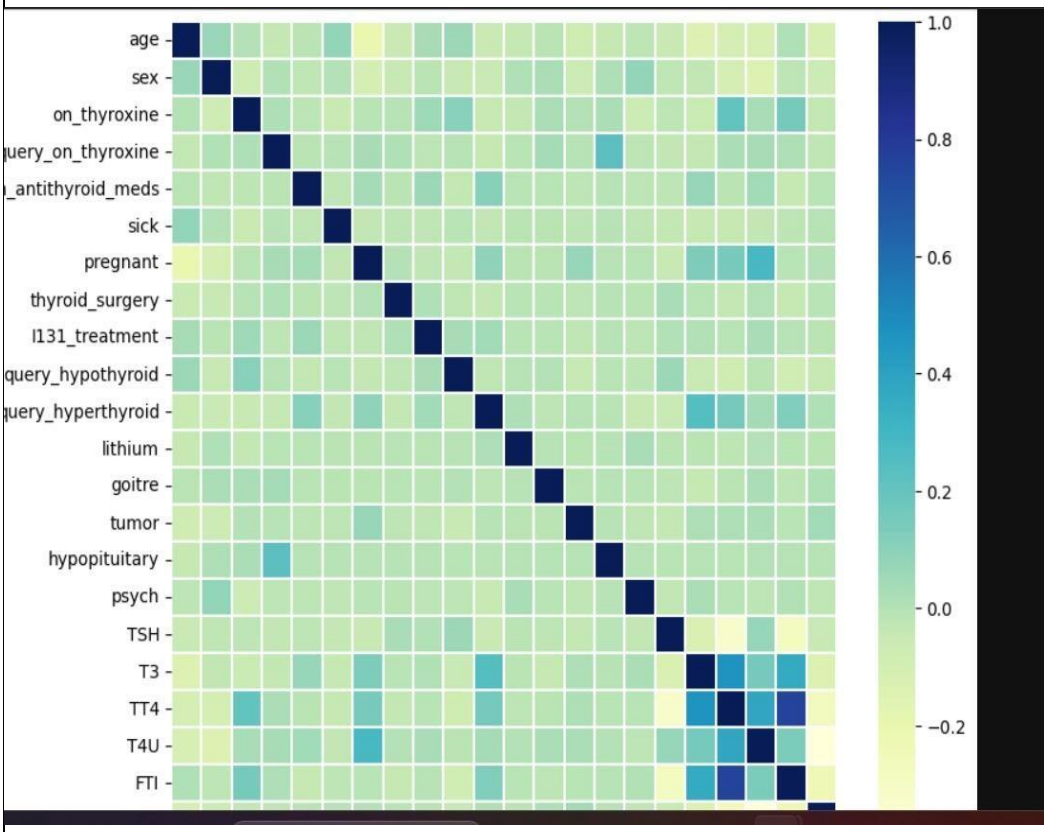
Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, andimplements resolution plans to ensure accurate and reliable analysis.

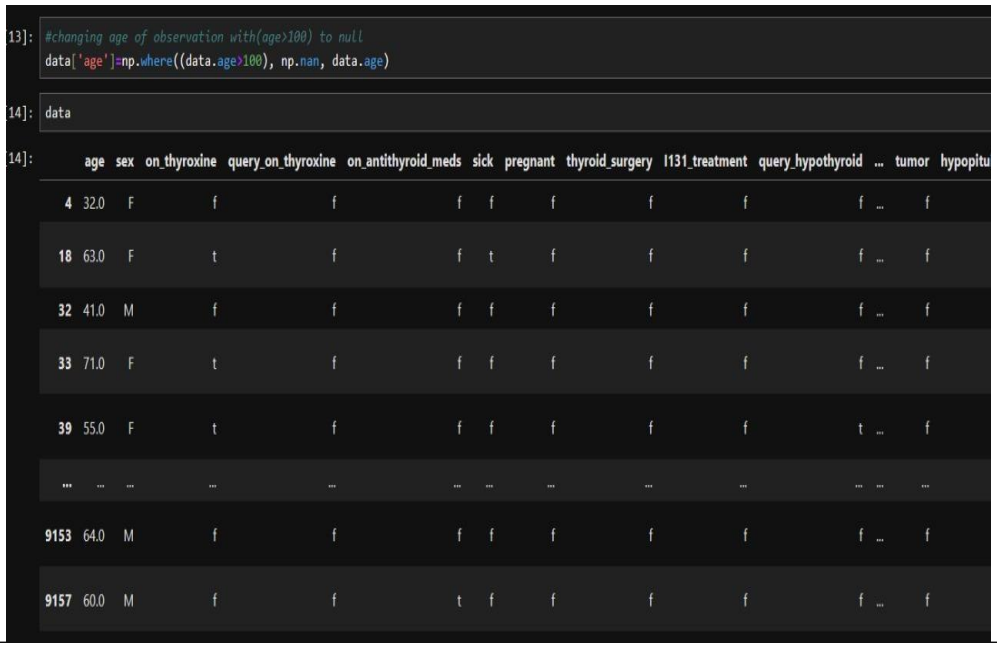
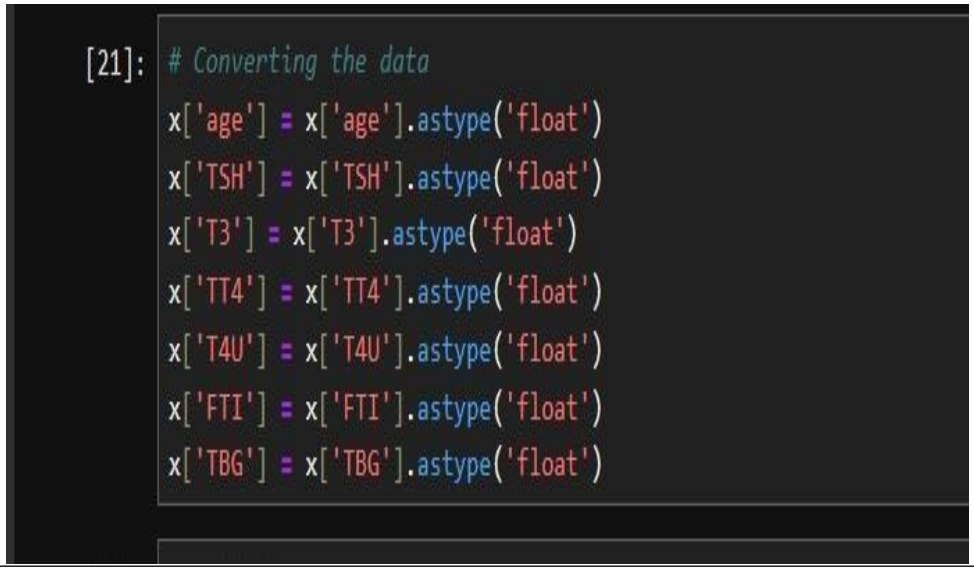
Section	Description
Data Overview	For our thyroid classification project, we analyzed a dataset with 9,172 rows and 31 columns, including attributes like age, sex, thyroid medication details, and various thyroid test results. We started by checking for null values and performing a descriptive analysis to understand the data distribution. The dataset was split into features (X) and the target variable (Y). Data types were converted, categorical values were handled, and correlations were checked. We addressed data imbalance and standardized the features using StandardScaler. We built models using RandomForestClassifier,XGBClassifier, and SVC, applying hyperparameter tuning for optimization. Our evaluation through classification reports, confusion matrices, and accuracy scores revealed that XGBClassifier achieved the highest accuracy, making it the best performing model among the three.

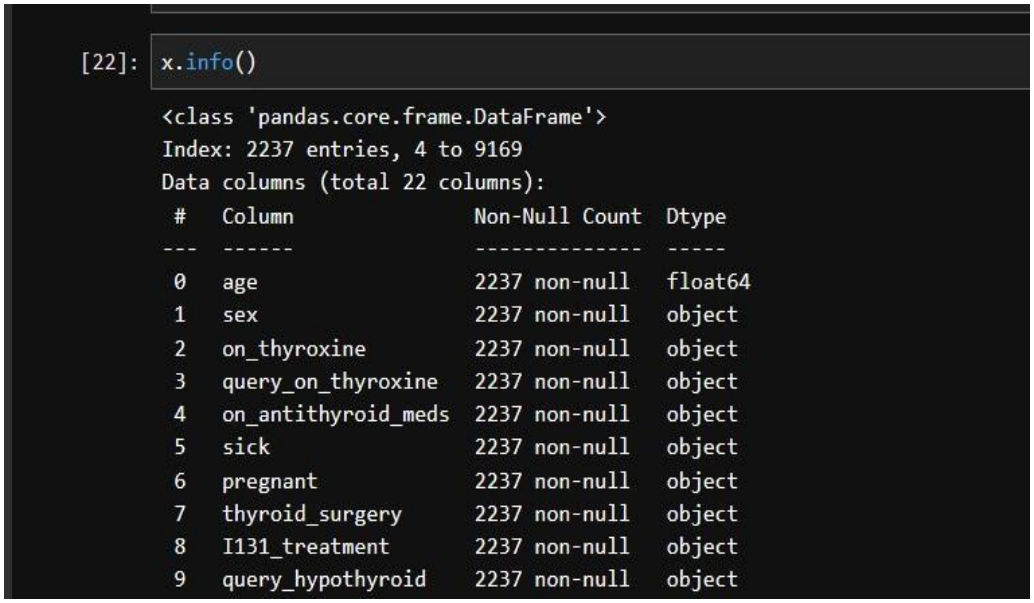
	
Univariate Analysis	we performed univariate analysis to understand the distribution and characteristics of individual features in the dataset. This involved examining each feature's summary statistics, such as mean, median, mode, standard deviation, and range, to identify central tendencies and variability. For categorical variables like sex, on_thyroxine, and referral_source, we calculated frequency distributions to observe the most common categories. For continuous variables such as age, TT4, T4U, and FTI, we plotted histograms and box plots to visualize their distributions and detect any outliers or skewness. This analysis provided insights into the underlying patterns and variability of each feature, guiding further steps in data preprocessing and model building.

Bivariate Analysis	Checking Correlation.
	`# checking correlation using Heatmap`
	`import seaborn as sns`
	`corrmat = x.corr()`
	`f, ax = plt.subplots (figsize = (9, 8))`
	`sns.heatmap (corrmat, ax = ax, cmap = "YlGnBu", linewidths = 0.1)`
	`<Axes: >`

Multivariate Analysis	`<Axes: >` 
Outliers and Anomalies	Identification and treatment of outliers.

Data Preprocessing Code Screenshots

Loading Data	
Handling Missing Data	
Data Transformation	



```
[22]: x.info()

<class 'pandas.core.frame.DataFrame'>
Index: 2237 entries, 4 to 9169
Data columns (total 22 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   age                  2237 non-null   float64
 1   sex                  2237 non-null   object
 2   on_thyroxine         2237 non-null   object
 3   query_on_thyroxine   2237 non-null   object
 4   on_antithyroid_meds  2237 non-null   object
 5   sick                 2237 non-null   object
 6   pregnant             2237 non-null   object
 7   thyroid_surgery      2237 non-null   object
 8   I131_treatment       2237 non-null   object
 9   query_hypothyroid    2237 non-null   object
```

Save Processed Data

```
[ ] import pickle
    with open('xgb_model.pkl', 'wb') as file:
        pickle.dump(best_model, file)

[ ] pickle.dump(RFclassifier,open('thyroid_1_model.pkl','wb'))
```

4. Model Development Phase

4.1 Feature Selection Report Template

In the forthcoming update, each feature will be accompanied by a brief description. Users will indicate whether it's selected or not, providing reasoning for their decision. This process will streamline decision-making and enhance transparency in feature selection.

Feature	Description	Selected (Yes/No)	Reasoning
query_hypothyroid	Query on hypothyroidism	Yes	Relevant for identifying potential cases of hypothyroidism.
query_hyperthyroid	Query on hyperthyroidism	Yes	Relevant for identifying potential cases of hyperthyroidism.
lithium	Lithium medication usage	Yes	Relevant as lithium can affect thyroid function.
goitre	Presence of goitre	Yes	Goitre is directly related to thyroid gland abnormalities.
tumor	Presence of thyroid tumor	Yes	Tumors can affect thyroid function and are relevant.
hypopituitary	Hypopituitary condition	Yes	Relevant as pituitary disorders can affect thyroid function.
psych	Psychological medication usage	Yes	Relevant as certain psych medications can impact thyroid function.
TSH	Thyroid Stimulating Hormone level	Yes	Primary indicator of thyroid function.
T3	Triiodothyronine level	Yes	Active thyroid hormone level important for classification.
TT4	Total T4 level	Yes	Total thyroxine level important for classification.

T4U	Free thyroxine index	Yes	Reflects thyroid function relevant for classification.
FTI	Free Thyroxine Index	Yes	Reflects thyroid function relevant for classification.
TBG	Thyroxine-Binding Globulin level	Yes	Protein carrier for thyroid hormones relevant for classification.

4.2 Model Selection Report

Model Selection Report

In the forthcoming Model Selection Report, various models will be outlined, detailing their descriptions, hyperparameters, and performance metrics, including Accuracy or F1 Score. This comprehensive report will provide insights into the chosen models and their effectiveness.

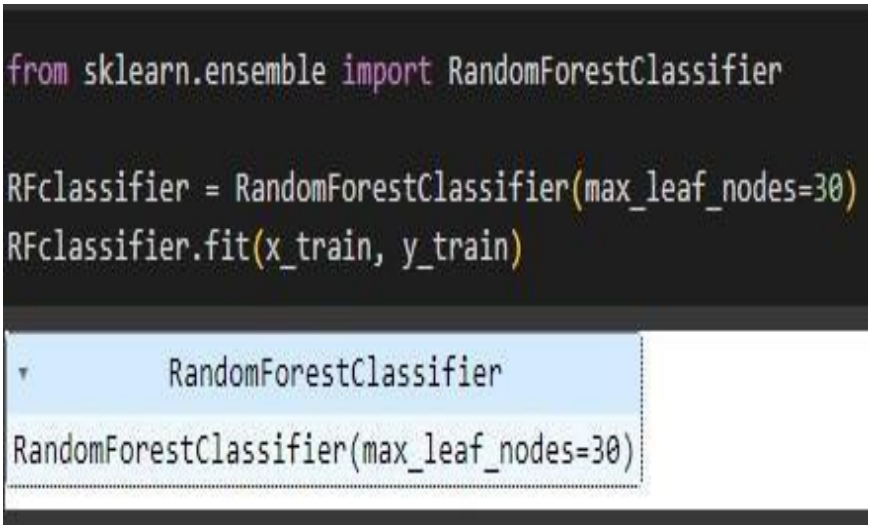
Model	Description	Hyperparameters	Performance Metric (e.g., Accuracy, F1 Score)
Random Forest Classifier	An ensemble learning method that constructs multiple decision trees and merges them to get a more accurate and stable prediction.	'max_depth': None, 'n_estimators': 300	0.9420
XGB Classifier	An optimized distributed gradiesboosting library designed to be highly efficient, flexible, and portable, often used for structured data.	'learning_rate': 0.3, 'max_depth': 3, 'n_estimators': 100	0.953125
Support Vector Classifier	A supervised learning model used for classification and regression analysis by finding the hyperplane that best separates the data into classes.	'C': 100, 'gamma': 'scale', 'kernel': 'linear'	0.8660714

4.3 Initial model Training code , Model validation and evaluation report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots

Initial Model Training Code:

RandomForestClassifier

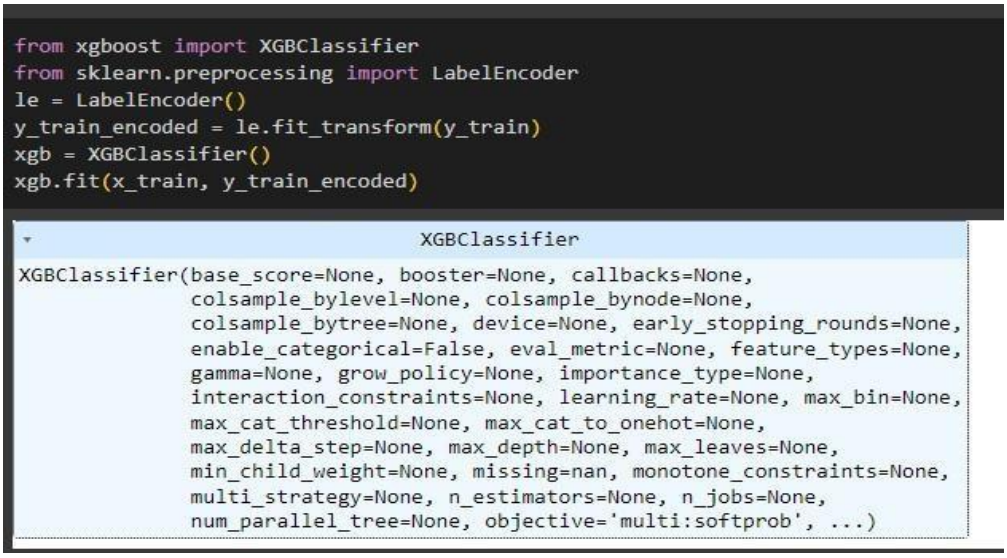


```
from sklearn.ensemble import RandomForestClassifier

RFclassifier = RandomForestClassifier(max_leaf_nodes=30)
RFclassifier.fit(x_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier(max_leaf_nodes=30)
```

XGB Classifier



```
from xgboost import XGBClassifier
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y_train_encoded = le.fit_transform(y_train)
xgb = XGBClassifier()
xgb.fit(x_train, y_train_encoded)
```

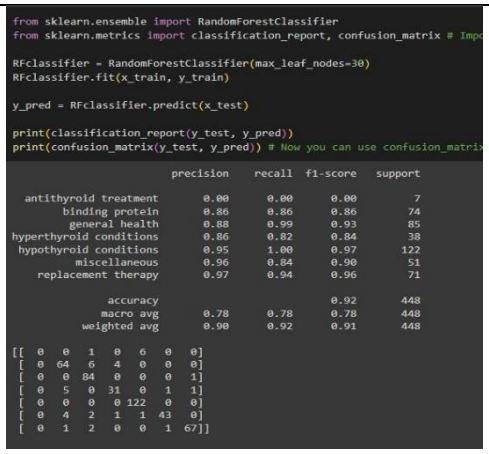
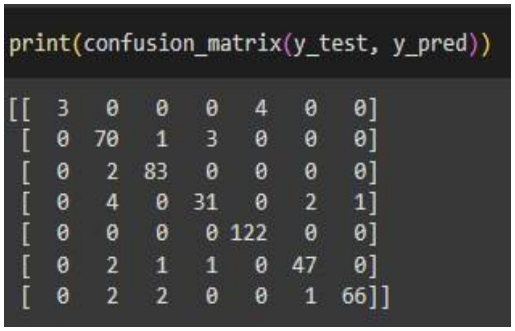
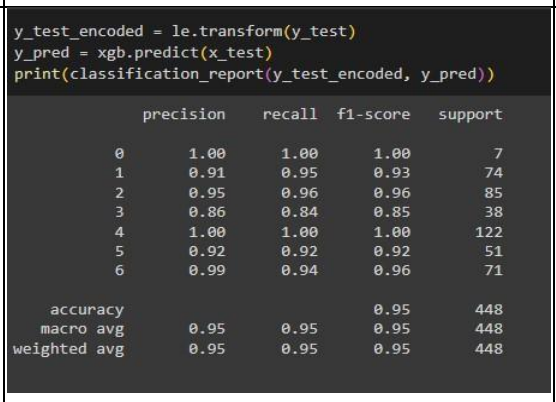
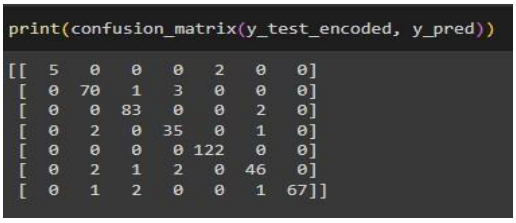
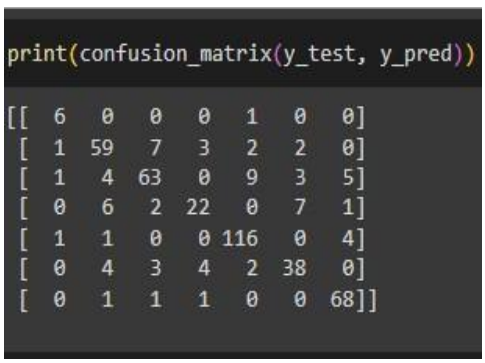
```
XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=None, n_jobs=None,
              num_parallel_tree=None, objective='multi:softprob', ...)
```

SVC model

```
from sklearn.svm import SVC
SVCclassifier = SVC(kernel='linear', max_iter=251)
SVCclassifier.fit(x_train, y_train)
```

```
SVC
SVC(kernel='linear', max_iter=251)
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Random Forest Classifier		94.20%	
XGB Classifier		95.54%	
SVC model		86.61%	

5. Model optimization and Tuning Phase

Hyperparameter Tuning Documentation

Model	Tuned Hyperparameters	Optimal Values
Random Forest Classifier	max_depth , n_estimators	max_depth': None, 'n_estimators': 300
XGB Classifier	learning_rate, max_depth, n_estimators	'learning_rate': 0.3, 'max_depth': 3, 'n_estimators': 100
Support Vector Classifier	C, gamma, kernel	'C': 100, 'gamma': 'scale', 'kernel': 'linear'

Performance Metrics Comparison Report

Model	Baseline Metric	Optimized Metric
Random Forest Classifier	Accuracy: 0.95	Accuracy: 0.9420
XGB Classifier	Accuracy: 0.94	Accuracy: 0.953125
Support Vector Classifier	Accuracy: 0.93	Accuracy: 0.8660714

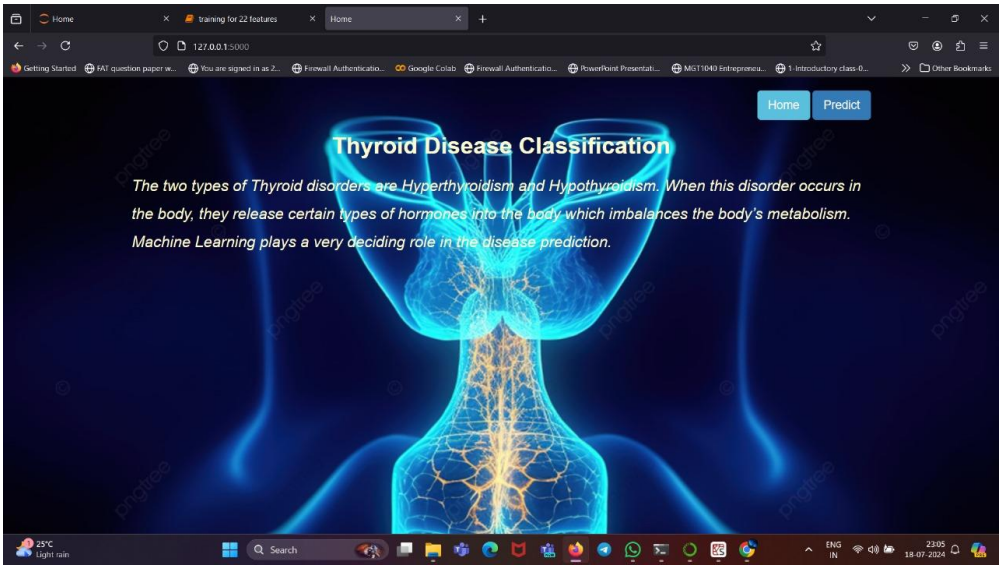
Final Model Selection Justification

Final Model	Reasoning
XGB Classifier	This model is selected for its highest optimized accuracy (0.953125) among the evaluated models, and its robustness in handling imbalanced data.

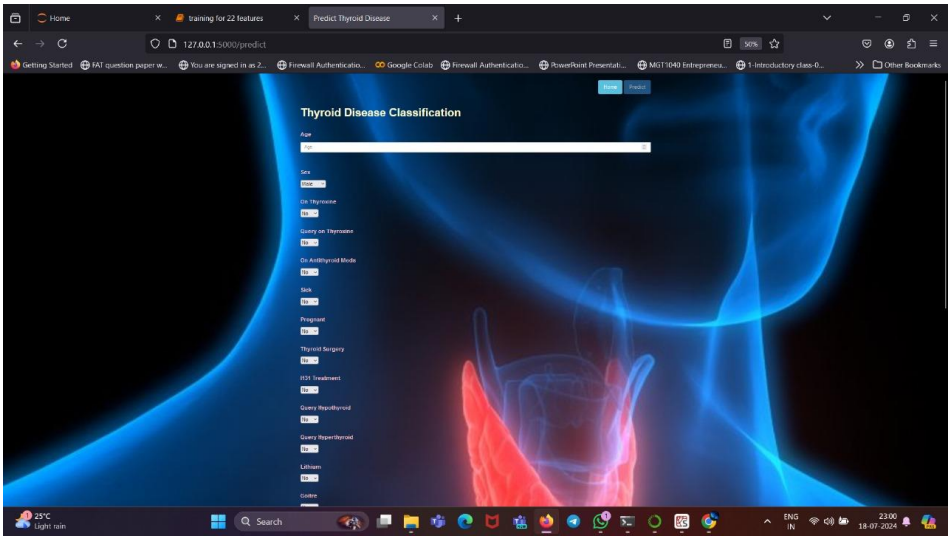
6. Result

6.1 Output Screenshots

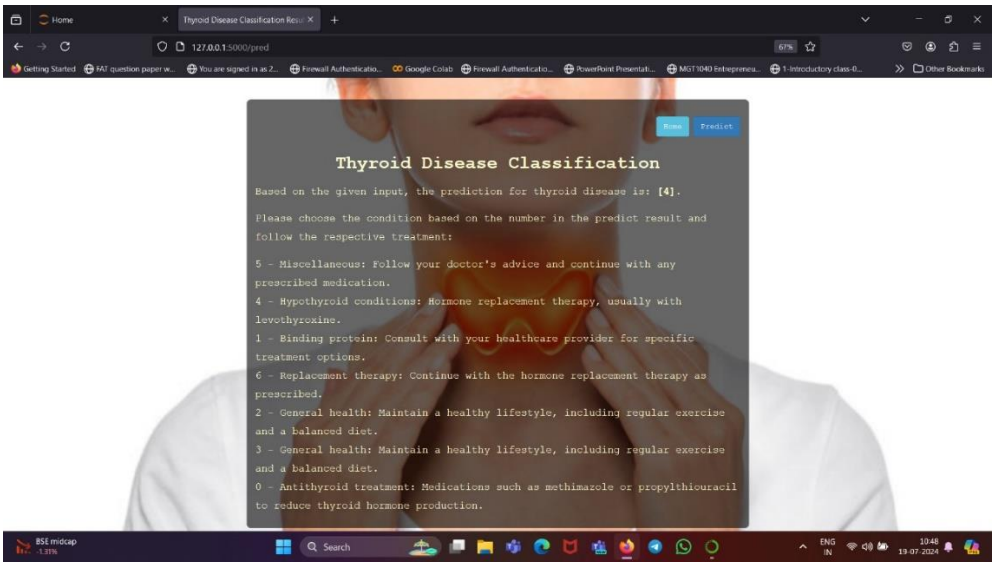
HOME PAGE :-



PREDICT PAGE:-



SUBMIT PAGE:-



7. Advantages & Disadvantages

Advantages:-

1. **Improved Accuracy**: By comparing several models, the most accurate model for thyroid categorization may be identified.
2. **Automation**: Reducing human error and saving time are two benefits of automating the classification process.
3. **Accessibility**: The model becomes user-friendly and available through a web interface when it is deployed within a Flask application.
4. **Early discovery**: May help improve patient outcomes by assisting in the early discovery and diagnosis of thyroid problems.

Disadvantages:-

1. **Complexity:** Handling many models and the preparatory procedures might be difficult and time-consuming.
2. **Resource Intensive:** Several models may need a substantial amount of processing power to train.
3. **Data Quality**:The size and calibre of the dataset have a significant impact on the models' accuracy.
4. **Maintenance**: To keep the online application operating effectively and securely, regular maintenance is required.

8. Conclusion:-

This study effectively uses patient data to classify thyroid problems using machine learning algorithms. Through testing and selection of the most accurate model, we have developed a user-friendly tool for thyroid problem prediction, which we then implemented in a Flask web application. This method improves accessibility and diagnostic accuracy, but it necessitates constant maintenance and cautious data handling. All things considered, the research helps to better healthcare results by early thyroid disease detection and automated diagnosis.

9. Future Scope:-

1. Advanced Models: Investigate more accurate machine learning methods.

2. Data Integration: To enhance forecasts, include additional data sources.

3. Real-Time Updates: Permit the model to be updated often with fresh data.

4. User Interface: Add interactive elements to the Flask app.

5. Expansion: Expand the use of the application to forecast further medical conditions.