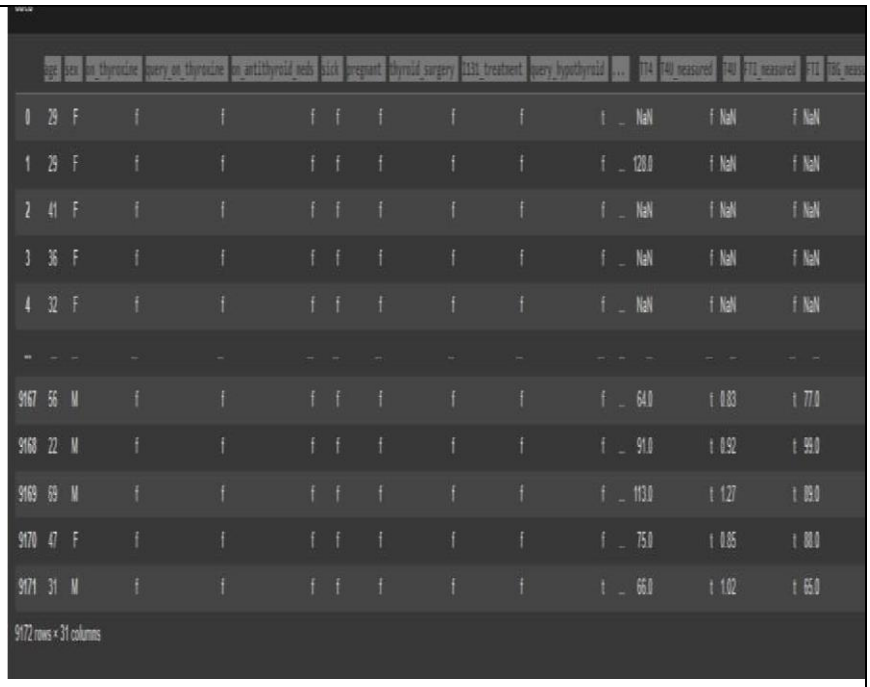
Data Collection and Preprocessing Phase

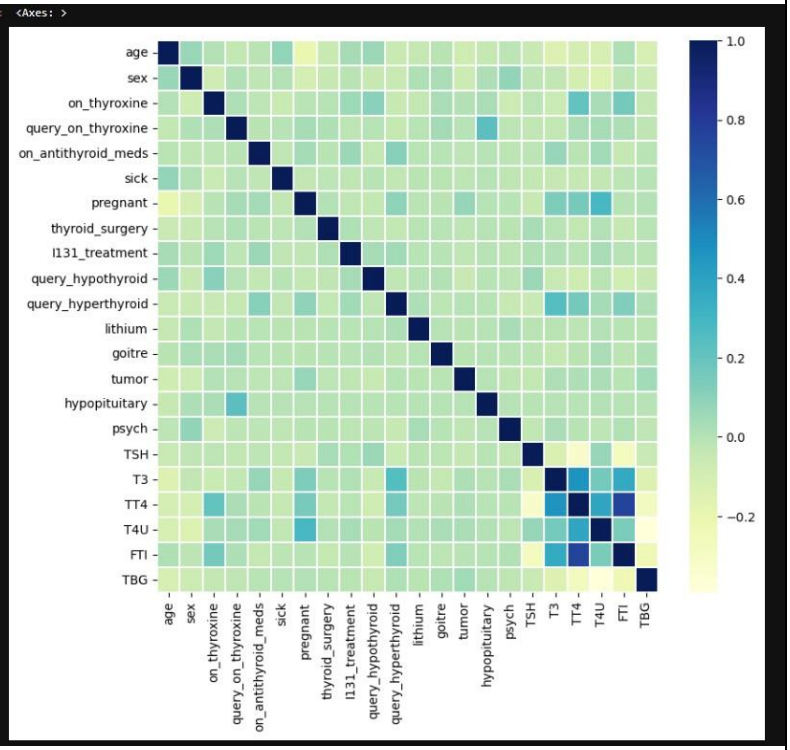
Date	15 March 2024
Team ID	SWTID1720437019
Project Title	Thyroid Classification
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

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

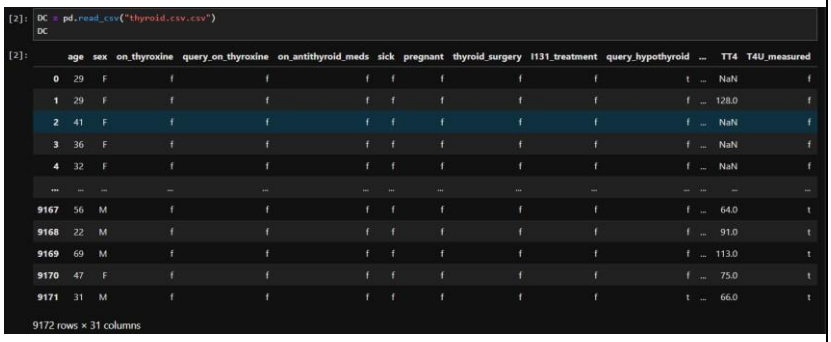
Section	**Description**
Data Overview	For our thyroid classification project, we analyzed a dataset with 9,172 rows and 31 columns, including attributes like age, sex, thyroid medication details, and various thyroid test results. We started by checking for null values and performing a descriptive analysis 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.

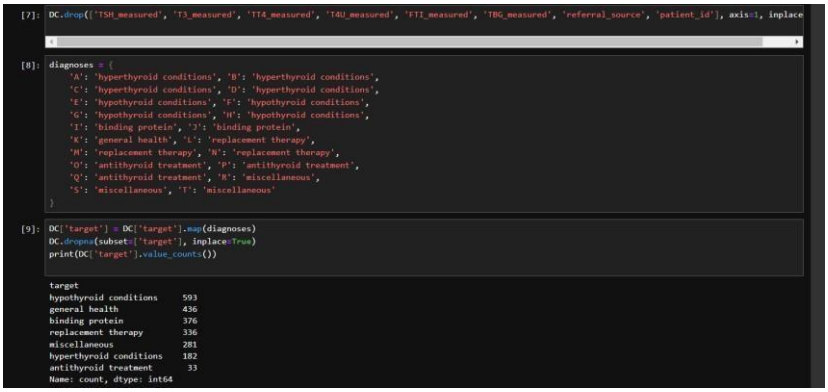
	
Univariate Analysis	we performed univariate analysis to understand the distribution and characteristics of individual features in the dataset. This involved examining each feature's summary statistics, such as mean, median, mode, standard deviation, and range, to identify central tendencies and variability. 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.

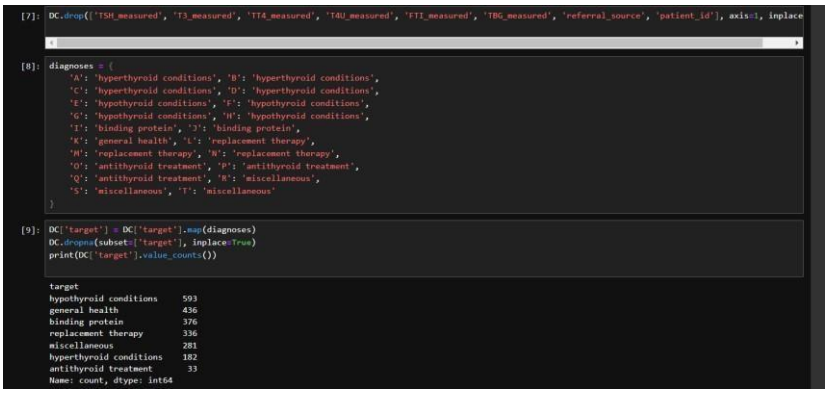
Bivariate Analysis	**Checking Correlation**
	[24]:
	``python
	import seaborn as sns
	corrmat = x.corr()
	
	f,ax = plt.subplots(figsize=(9,8))
	sns.heatmap(corrmat, ax = ax, cmap="YlGnBu", linewidths = 0.1)
	``
	[24]: <Axes: >
	 

Multivariate Analysis	
Outliers and Anomalies	-

Data Preprocessing Code Screenshots

Loading Data	 [2]: DC = pd.read_csv("thyroid.csv.csv") DC [2]: 9172 rows × 31 columns

Handling Missing Data	
	[7]: DC.drop(['TSH_measured', 'T3_measured', 'TT4_measured', 'T4U_measured', 'FTI_measured', 'TBG_measured', 'referral_source', 'patient_id'], axis=1, inplace
	[8]: diagnoses = {
	'A': 'hyperthyroid conditions', 'B': 'hyperthyroid conditions',
	'C': 'hyperthyroid conditions', 'D': 'hyperthyroid conditions',
	'E': 'hypothyroid conditions', 'F': 'hypothyroid conditions',
	'G': 'hypothyroid conditions', 'H': 'hypothyroid conditions',
	'I': 'binding protein', 'J': 'binding protein',
	'K': 'general health', 'L': 'replacement therapy',
	'M': 'replacement therapy', 'N': 'replacement therapy',
	'O': 'antithyroid treatment', 'P': 'antithyroid treatment',
	'Q': 'antithyroid treatment', 'R': 'miscellaneous',
	'S': 'miscellaneous', 'T': 'miscellaneous'
	}
	[9]: DC['target'] = DC['target'].map(diagnoses)
	DC.dropna(subset=['target'], inplace=True)
	print(DC['target'].value_counts())
	target
	hypothyroid conditions 593
	general health 436
	binding protein 376
	replacement therapy 336
	miscellaneous 281
	hyperthyroid conditions 182
	antithyroid treatment 33
	Name: count, dtype: int64

Data Transformation	
	[7]: DC.drop(['TSH_measured', 'T3_measured', 'TT4_measured', 'T4U_measured', 'FTI_measured', 'TBG_measured', 'referral_source', 'patient_id'], axis=1, inplace
	[8]: diagnoses = {
	'A': 'hyperthyroid conditions', 'B': 'hyperthyroid conditions',
	'C': 'hyperthyroid conditions', 'D': 'hyperthyroid conditions',
	'E': 'hypothyroid conditions', 'F': 'hypothyroid conditions',
	'G': 'hypothyroid conditions', 'H': 'hypothyroid conditions',
	'I': 'binding protein', 'J': 'binding protein',
	'K': 'general health', 'L': 'replacement therapy',
	'M': 'replacement therapy', 'N': 'replacement therapy',
	'O': 'antithyroid treatment', 'P': 'antithyroid treatment',
	'Q': 'antithyroid treatment', 'R': 'miscellaneous',
	'S': 'miscellaneous', 'T': 'miscellaneous'
	}
	[9]: DC['target'] = DC['target'].map(diagnoses)
	DC.dropna(subset=['target'], inplace=True)
	print(DC['target'].value_counts())
	target
	hypothyroid conditions 593
	general health 436
	binding protein 376
	replacement therapy 336
	miscellaneous 281
	hyperthyroid conditions 182
	antithyroid treatment 33
	Name: count, dtype: int64

Feature Engineering	Attached the codes in final submission.

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'))