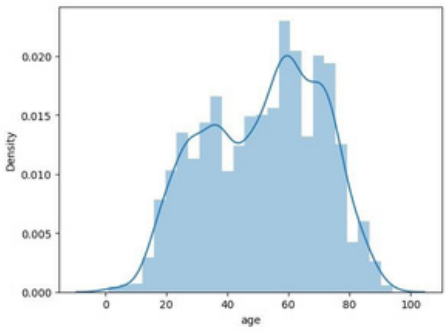
Data Collection and Preprocessing Phase

Date	3 July 2024
Team ID	866654
Project Title	Thyroid Classification using machine Learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

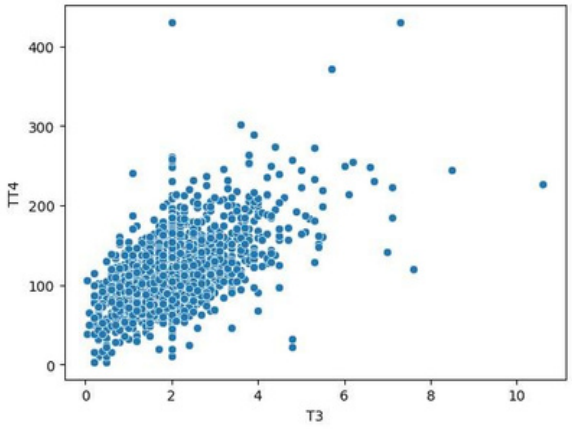
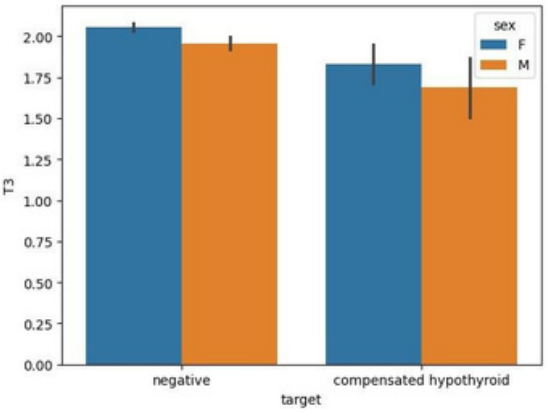
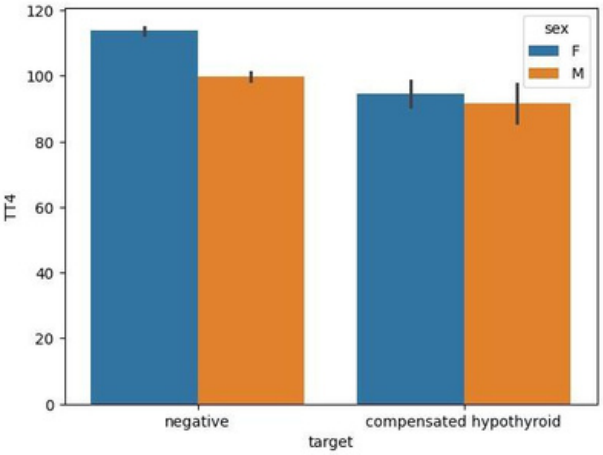
Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

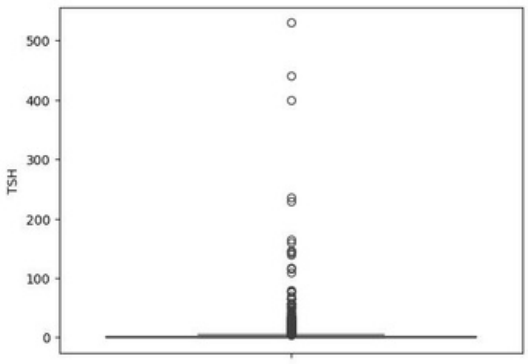
Section	Description
Data Overview	Dimensions: 4744 rows x 30 cols

Univariate Analysis	

	age	TSH	T3	TT4	T4U	FTI
count	4581.000000	4581.000000	4581.000000	4581.000000	4581.000000	4581.000000
mean	51.586335	3.084818	2.016608	108.987645	0.989697	111.248810
std	19.000420	14.920483	0.709480	32.830981	0.185445	29.344041
min	1.000000	0.005000	0.050000	2.900000	0.250000	2.800000
25%	36.000000	0.590000	1.700000	90.000000	0.890000	95.000000
50%	54.000000	1.300000	2.000000	104.000000	0.970000	107.000000
75%	67.000000	2.300000	2.200000	123.000000	1.060000	122.000000
max	94.000000	530.000000	10.600000	430.000000	2.320000	395.000000

Descriptive Analysis:

Bivariate Analysis	
Multivariate Analysis	 

Outliers and Anomalies	
Data Preprocessing Code Screenshots	
Loading Data	`df=pd.read_csv('/content/dataset123.csv')`
Handling Missing Data	`df.isnull().sum()` `age 0` `TSH 0` `T3 0` `TT4 0` `T4U 0` `FTI 0` `target 0` `sex_F 0` `sex_M 0` `on_thyroxine_f 0` `on_thyroxine_t 0` `on_antithyroid_medication_f 0` `on_antithyroid_medication_t 0` `sick_f 0` `sick_t 0` `pregnant_f 0` `pregnant_t 0` `thyroid_surgery_f 0` `thyroid_surgery_t 0` `I131_treatment_f 0` `I131_treatment_t 0` `query_on_thyroxine_f 0` `query_on_thyroxine_t 0` `query_hypothyroid_f 0` `query_hypothyroid_t 0` `query_hyperthyroid_f 0` `query_hyperthyroid_t 0` `lithium_f 0` `impute=SimpleImputer(strategy='most_frequent')` `impute1=SimpleImputer(strategy='median')` `df.replace('?',np.nan,inplace=True)` `df[['sex']]=impute.fit_transform(df[['sex']])` `df_values=['negative','compensated hypothyroid','primary thyroid']` `df= df[df['target'].isin(df_values)]` `df['target'].value_counts()` `df['age']=pd.to_numeric(df['age'],errors='coerce')` `df=df[df['age']!=455]` `mean_age=df['age'].mean()` `df['age']=df['age'].fillna(mean_age) #df['age']=impute1.fit_transform(df[['age']])` `df['age']=df['age'].round(0).astype('int')` `df['age'].unique()`

	`` df['TSH']=pd.to_numeric(df['TSH'],errors='coerce') df['T3']=pd.to_numeric(df['T3'],errors='coerce') df['TT4']=pd.to_numeric(df['TT4'],errors='coerce') df['FTI']=pd.to_numeric(df['FTI'],errors='coerce') df['T4U']=pd.to_numeric(df['T4U'],errors='coerce') ``` ``` df['TSH']=impute1.fit_transform(df[['TSH']]) df['TSH'] ``` ``` df['T3']=impute1.fit_transform(df[['T3']]) df['T3'].unique() ``` ``` df['TT4']=impute1.fit_transform(df[['TT4']]) df['TT4'].unique() ``` ``` df['T4U']=impute1.fit_transform(df[['T4U']]) df['T4U'].unique() ``` ``` df['FTI']=impute1.fit_transform(df[['FTI']]) df['FTI'].unique() ```
Data Transformation	-
Feature Engineering	-
Save Processed Data	-