**Thyroid Disease data analysis and prediction**

Abdul Basith Mohammed (T00700187)

IFSC 7370 Data Science Technology

Dr Elizabeth Pierce

Spring 2023

05/08/2023

**Contents**

[**Introduction** 3](#_Toc134536800)

[**Data** 4](#_Toc134536801)

[**Exploratory Data Analysis and Data Pre-Processing** 4](#_Toc134536802)

[**Clustering** 8](#_Toc134536803)

[**Feature Selection** 9](#_Toc134536804)

[**Model Training/Selection** 9](#_Toc134536805)

[**Conclusion / Results** 11](#_Toc134536806)

[Classification\_report for Cluster 2 || DecisionTreeClassifier() 12](#_Toc134536807)

[Classification\_report for Cluster 0 || RandomForestClassifier() 12](#_Toc134536808)

[Classification\_report for Cluster 1 || RandomForestClassifier() 13](#_Toc134536809)

[Classification\_report for Cluster 2 || RandomForestClassifier() 13](#_Toc134536810)

[**Execution Time:** 13](#_Toc134536811)

[**Future Work / Critique** 13](#_Toc134536812)

[**Critique:** 14](#_Toc134536813)

[**Certification** 14](#_Toc134536814)

[Brief Background about myself: 14](#_Toc134536815)

[Complete 2022 Data Science & Machine Learning Bootcamp Certificate: 15](#_Toc134536816)

[**Time spent for the Data Science project and certification** 15](#_Toc134536817)

# **Introduction**

Thyroid is one of the most crucial glands of human body. It is responsible for releasing hormones essential for regulating metabolism, blood pressure, body temperature, and heart rate (“Thyroid disorders”, 2022). Some of the important hormones released by Thyroid gland are triiodothyronine (T3), and T4 or thyroxine.

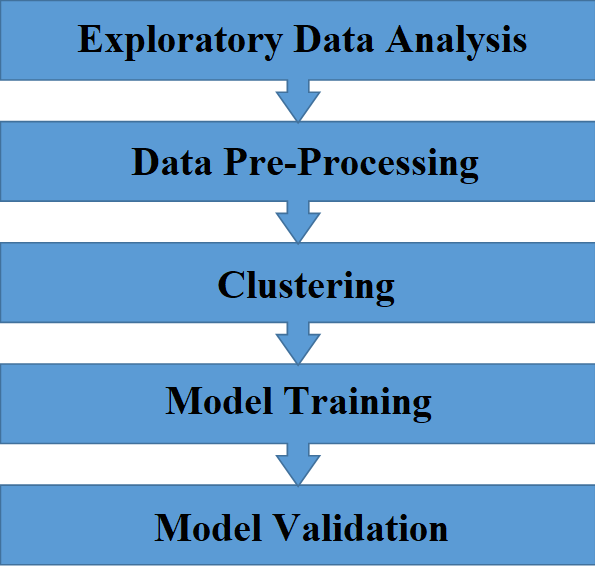
Unfortunately, like other organs/glands of the human body, it does gets sick / has diseases that affect its normal functioning. The reasons for thyroid diseases can be plenty including lifestyle, family genetics, nutrient deficiencies, or pregnancies. Primarily, there are two types of thyroid disorders/diseases.

1. Hyperthyroidism – Excess quantity of thyroid hormones produced.

2. Hypothyroidism – Inadequate thyroid hormones production.

**Problem Statement:**

In this project / report, I am going to use Machine Learning models for predicting the type of thyroid disease. This project is important and interesting to me because I too have hypothyroidism and curious to know more about it using analysis. This project will be interesting because I am using multiple machine learning algorithms such as K-means clustering and Random Forest for prediction. The flow of the project is illustrated by the below diagram.



**Process Flow Diagram**

# **Data**

The data for this project is taken from the University of California Irvine Machine Learning Repository (<https://archive.ics.uci.edu/ml/machine-learning-databases/thyroid-disease/>). There are many data files present in the source location. I am using the file named ‘allhypo.data’. After downloading the file, I changed its extension to .csv format. Added the headers using the ‘allbp.names’ text file. The dataset contains 2800 records and 30 columns. The target column is Class, it is a qualitative column having four unique values as mentioned below:

1. Negative
2. compensated hypothyroid
3. primary hypothyroid
4. secondary hypothyroid

This is a classification problem because we are predicting the qualitative column.

The columns/features of this dataset:

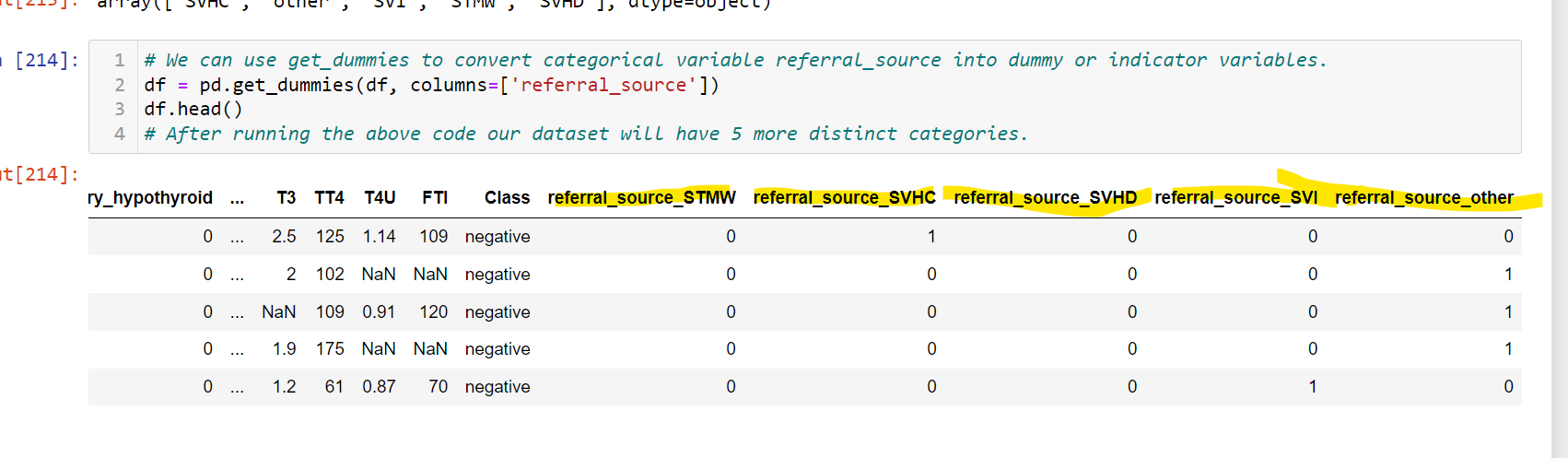
'age', 'sex', 'on\_thyroxine', 'query\_on\_thyroxine', 'on\_antithyroid\_medication', 'sick', 'pregnant', 'thyroid\_surgery', 'I131\_treatment', 'query\_hypothyroid', 'query\_hyperthyroid', 'lithium','goitre', 'tumor', 'hypopituitary', 'psych', 'TSH\_measured', 'TSH','T3\_measured', 'T3', 'TT4\_measured', 'TT4', 'T4U\_measured', 'T4U','FTI\_measured', 'FTI', 'TBG\_measured', 'TBG', 'referral\_source'.

* Thyroxine – (Called as T4), TSH, and T3 are thyroid hormones.
* TBG is a protein produced by liver helpful in distribution of T4.
* Hypopituitary – Condition of pituitary gland.
* Hypothyroid and Hyperthyroid are explained in the introduction.
* FTI means Free thyroxine index which is used for calculating the amount of Free Thyroxine in the bloodstream.

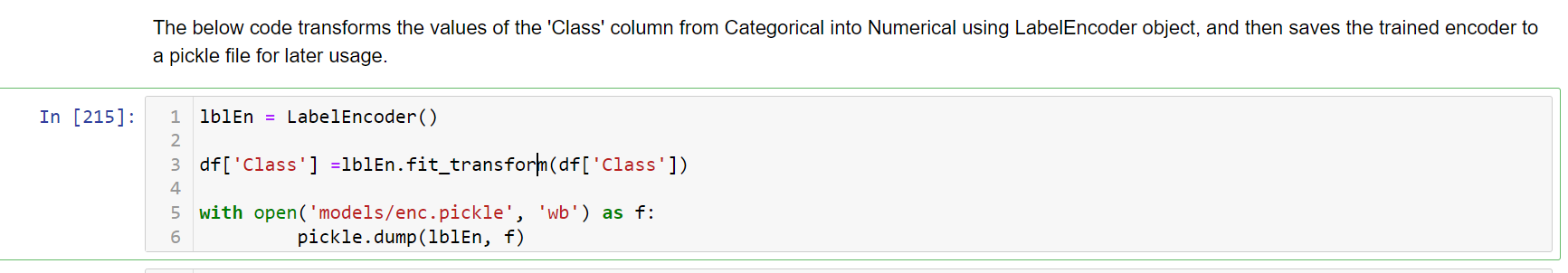
# **Exploratory Data Analysis and Data Pre-Processing**

Data cleaning is very important for the model training and accuracy. I have done EDA to have a thorough understanding of the dataset and cleaned the null values, duplicate values, dropped unnecessary columns.

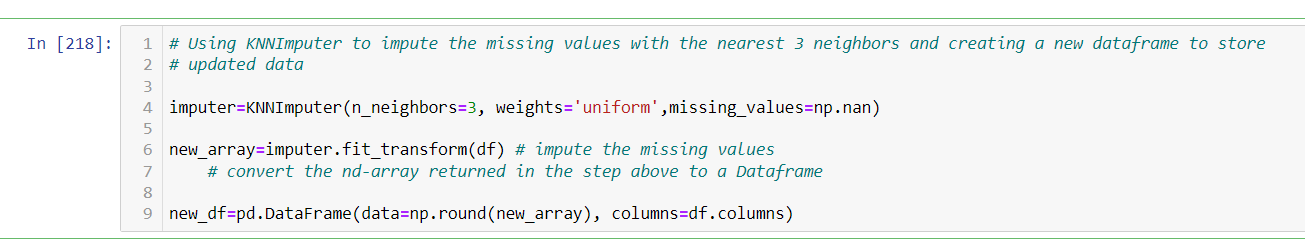
After carefully looking at the data it can be found that “?” have replaced the null values. So, I have replaced the? marks with the np.nan. There are few columns by name ‘TBG’, dummy values denoted with ‘measured’, I have dropped them. We have many categorical columns that need to be converted to numerical before we apply any imputation techniques. I have utilized the map function of python for mapping F and M to 0 and 1 respectively. Mapped the columns having f and m to 0 and 1 respectively. Used get\_dummies() to convert categorical variable referral\_source into dummy or indicator variables.



To convert the Class column from Categorical into Numerical used LabelEncoder object, and then saves the trained encoder to a pickle file for later usage.



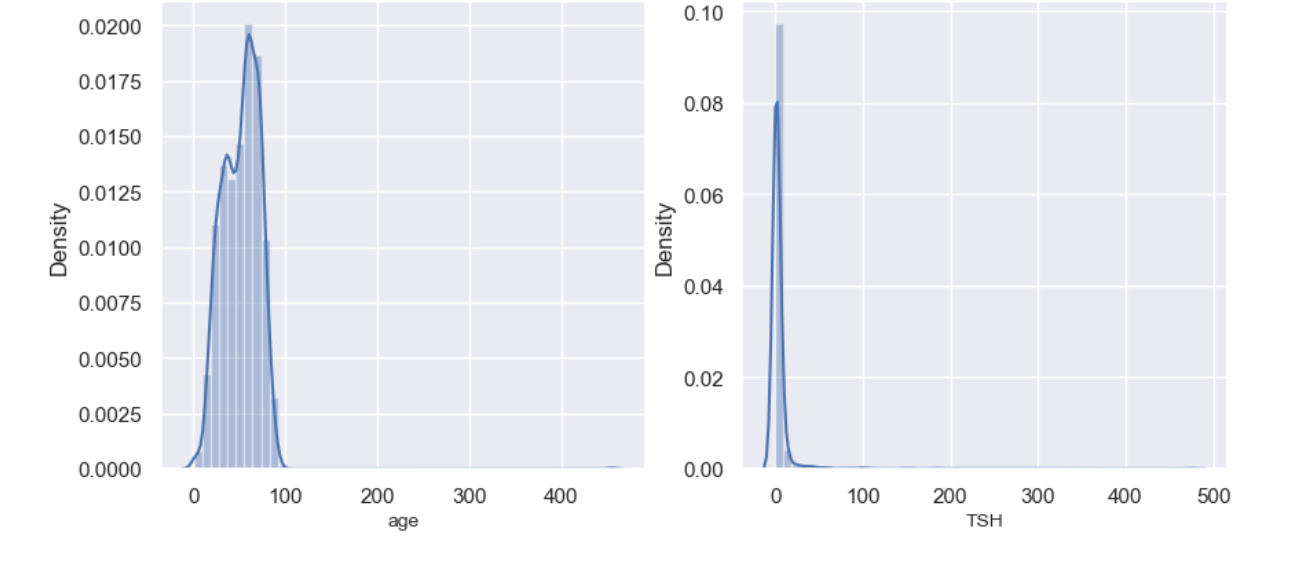
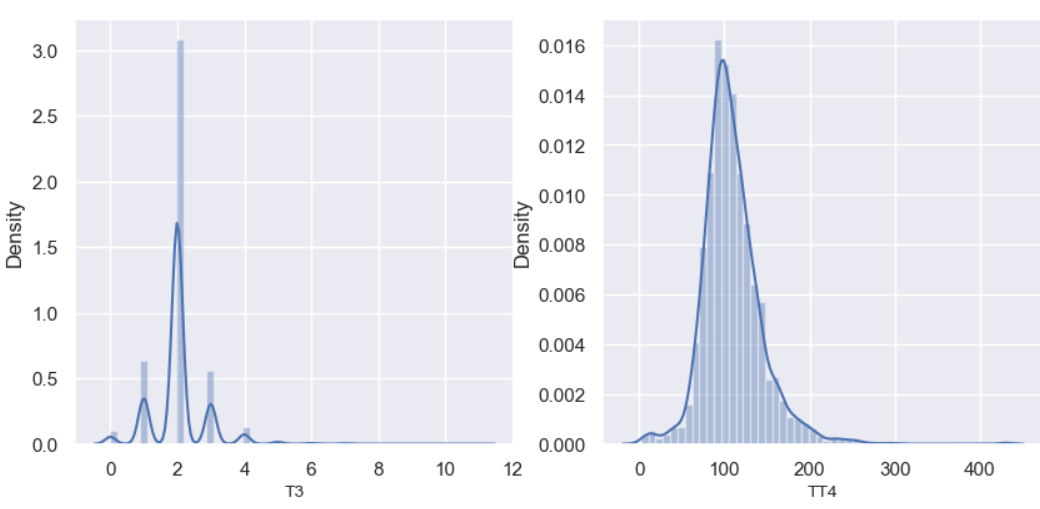
After converting all the categorical data into numerical data, used KNNImputer to impute the missing values.

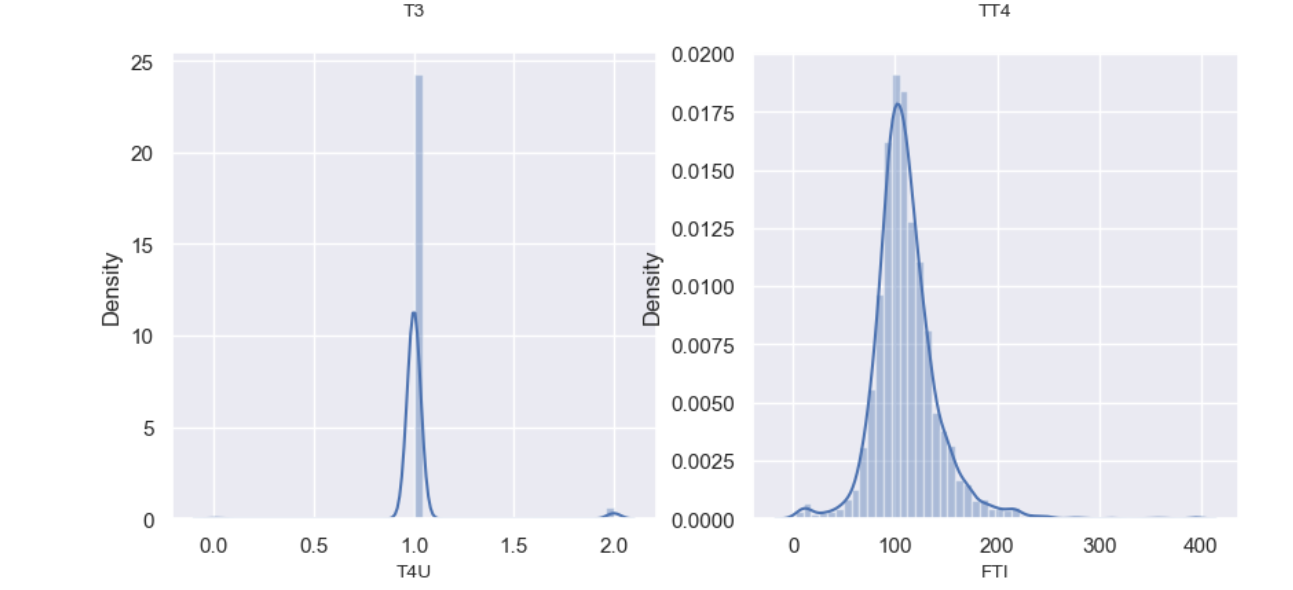


Now the data is cleaned, its ready for the visualizations.

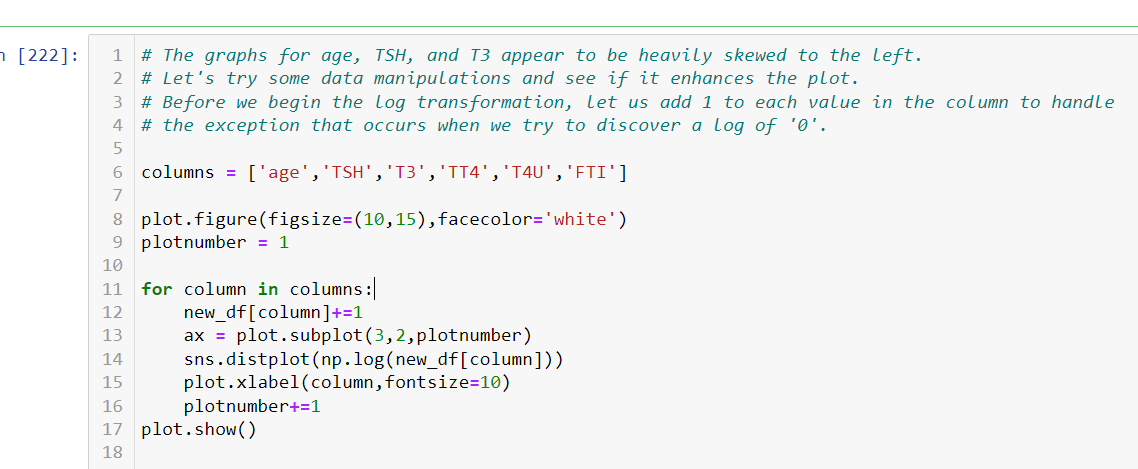
**Visualizations:**

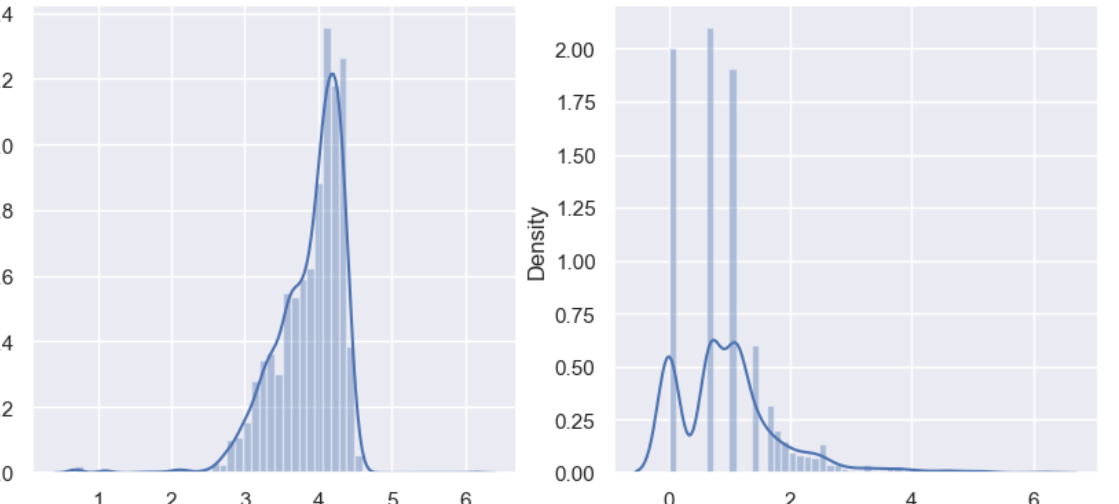
Used the seaborn distplot for plotting the overall distribution of the columns ['age', 'TSH' , 'T3', 'TT4' , 'T4U' ,'FTI']. This graphs looks skewed to either left or right.



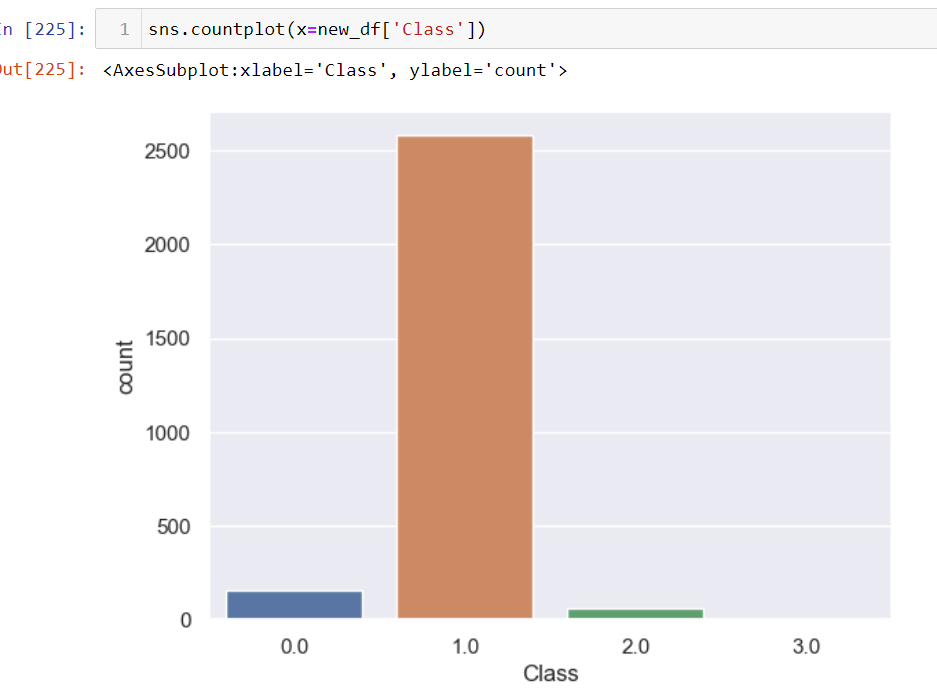


Applying the log transformation to correct it.

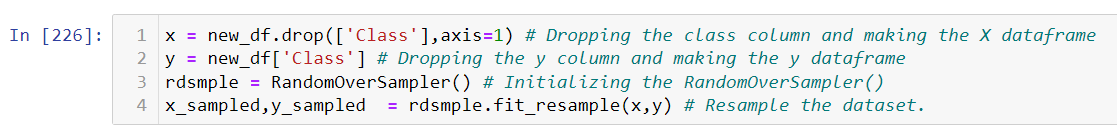


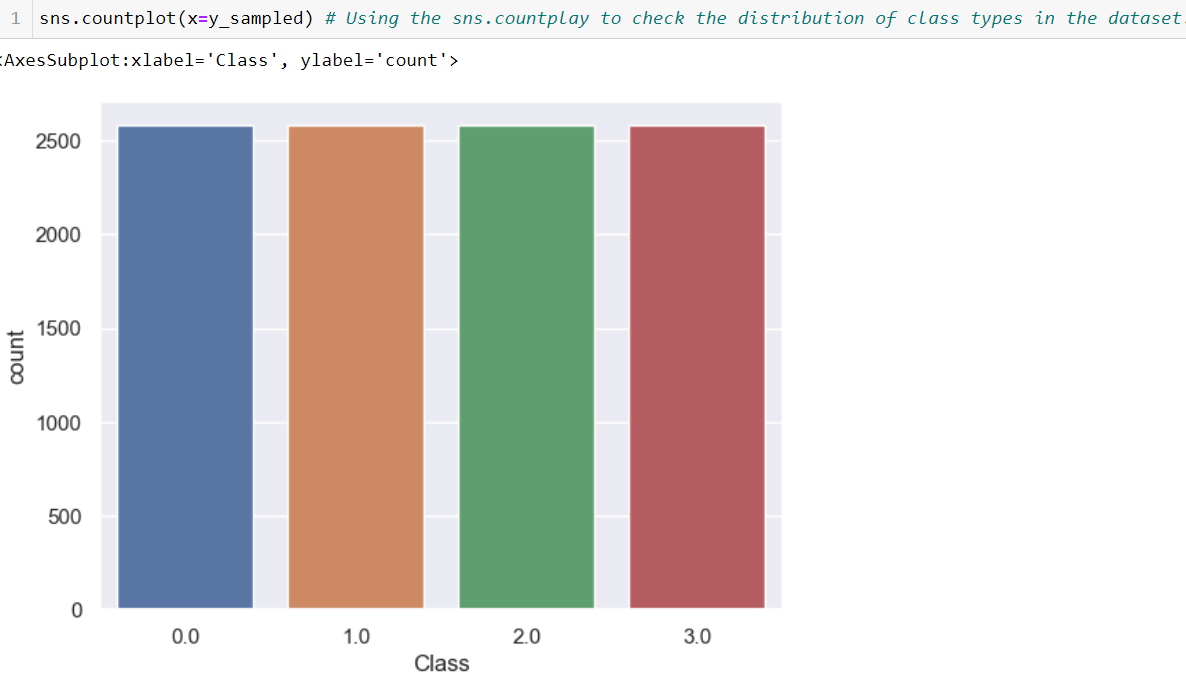


Dataset was heavily imbalanced i.e., the ‘Class’ column had nearly 3500 occurrences of the same value (negative). So, did some oversampling to balance the data.



The below piece of code does the oversampling and balances the data.

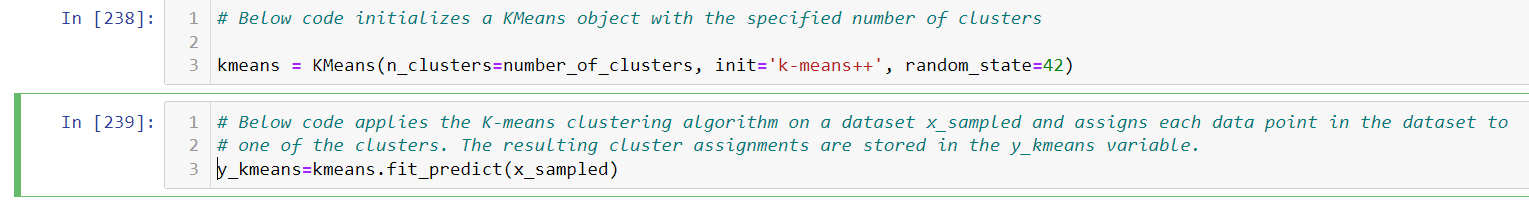




Now the data quality is improved and it can be clustered into different groups.

# **Clustering**

In this phase of my project, we are grouping our data into various clusters using K-means clustering algorithm. We are using the elbow plot and kneed libraries knee attribute to get the optimum number of clusters to split the data. After getting the K value, I have implemented the K Means algorithm onto the pre-processed dataset.



I am saving all the models that are generated during this training locally on the file system for future predictions. I have created a custom function called model\_saving() for it.

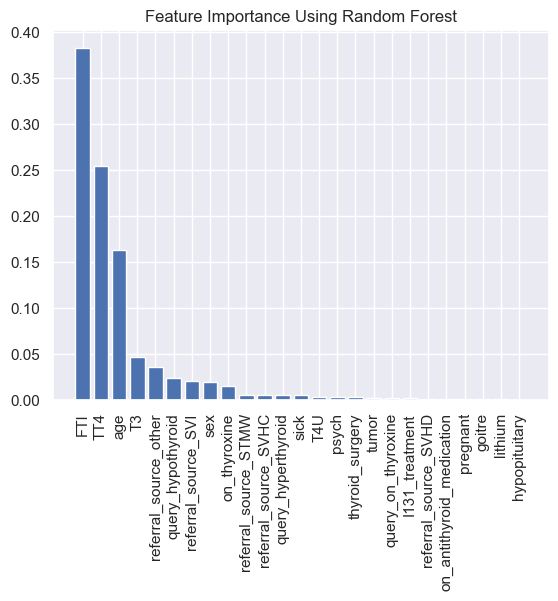
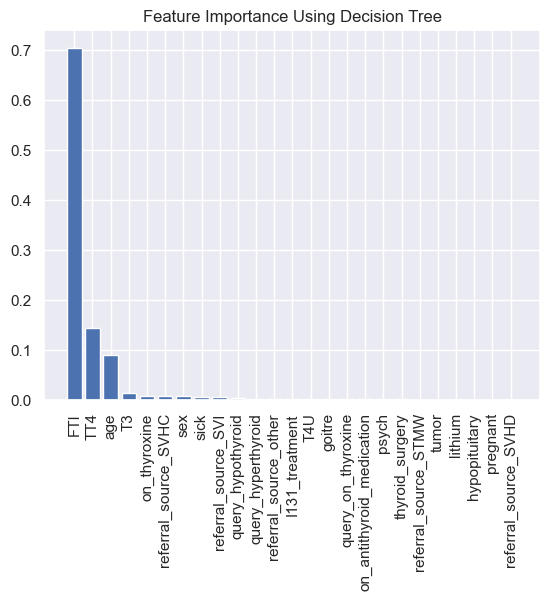
After creating the clusters, I am adding a label column to the data for the identification of the records with respect to the cluster numbers.

Now, that we have divided our dataset into clusters we can proceed with the model selection/training.

# **Feature Selection**

In this phase, I am selecting only the features that are most important for the prediction of thyroid disease. I am using the Decision Tree and Random Forest classifiers from the Scikit learn library to first shortlist the important features which will be further used in the Model Training/Selection phase.

Below are the graphs of the important features after applying Decision Tree and Random Forest.



After looking at the above figures, I have shortlisted only the below mentioned 9 features/predictors for better accuracy and less complexity.

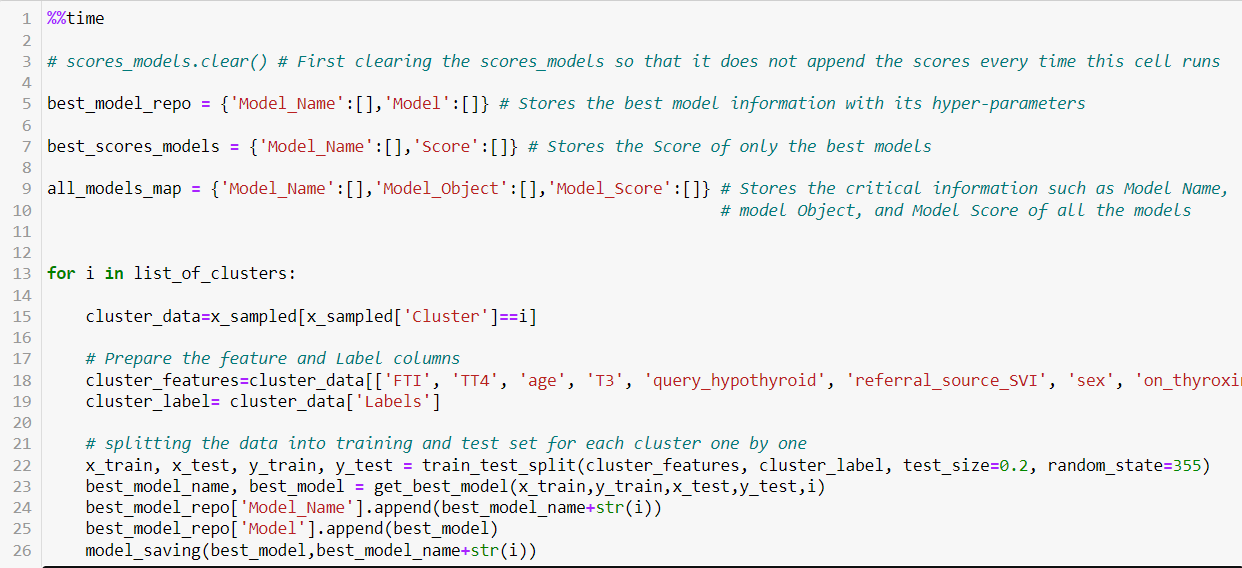
['FTI', 'TT4', 'age', 'T3', 'query\_hypothyroid', 'referral\_source\_SVI', 'sex', 'on\_thyroxine', 'referral\_source\_STMW'].

**From those features the top 3 predictors for the Thyroid disease are FTI, TT4, and age.**

# **Model Training/Selection**

This is the most crucial and exciting phase of the project. RandomForestClassifier and DecisionTreeClassifier are the two models used for the training purpose. I am doing hyperparameter tuning using GridSearchCV() to get the best parameters for increased accuracy and stability of the models. I have designed the code of the model training using modularity i.e., in small blocks. I have created separate functions calling each model and separated the model training. **Detailed code and its explanation is provided in the python notebook** that I have created for the Model Training named **thyroid\_ml\_project.ipynb.**

I will give the overview of each function and the data flow in brief here.



**Code block1:**

The above code uses the for loop to get each cluster’s data sequentially. It splits the cluster\_data into two parts in the ratio of 80:20 (80 % for training and 20 % for testing). It passes the clustered data to the function **get\_best\_model()** and gets the model (either DecisionTree or Random Forest) which is best for the respective cluster’s data.

**Method 2: get\_best\_model()**

The **get\_best\_model()** function has two functions **cc**() and **get\_best\_model\_for\_random\_forest**() which are implementing the Hypertuned DecisionTree and Random Forest models. The data is **given** to these both functions and a best DecisionTree and Random Forest Model is **returned** for each cluster. The accuracy of both the models is measured using different metrics and the best model is returned to the main code.

**Method 3: get\_best\_model\_for\_DecisionTree**()

The **get\_best\_model\_for\_DecisionTree** () takes the training data. Uses GridSearchCV() to find the best parameters to implement DecisionTreeClassifier(), fits the classifier and returns the DecisionTree model.

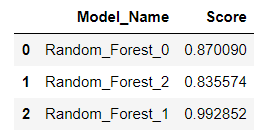
**Method 4**: **get\_best\_model\_for\_random\_forest**()

Similar to the get best model for knn, **get\_best\_model\_for\_random\_forest**() takes the training data. Uses GridSearchCV() to find the best parameters to implement RandomForestClassifier(), fits the classifier and returns the RandomForest model.

# **Conclusion / Results**

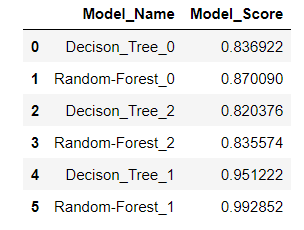
The below tables and output gives us the accuracy, recall and F-1 score of both the decision tree and random forest models for each clusters. By looking at this tables we can conclude that the RandomForestClassifiers are best suited for this dataset. The accuracy of the random forest for the clusters 1 and 2 is nearly 95 % which is pretty impressive. The accuracy of 98% for the cluster 0 suggests that the this model might be overfitted because of the data.

**Scores of best models an each cluster:**



The below table illustrates the scores of all the models (RandomForest and Decision Tree). The decision tree model scores are also impressive here.

**Scores of all the Models on each cluster:**



**Classification\_report for Cluster 0 DecisionTreeClassifier()**

I am passing each cluster data to the decision tree and calculating the precision, recall, f1-score using classification\_report.

precision recall f1-score support

0.0 0.62 0.35 0.44 81

1.0 0.79 0.93 0.86 255

2.0 1.00 0.44 0.61 16

accuracy 0.78 352

macro avg 0.81 0.57 0.64 352

weighted avg 0.76 0.78 0.75 352

Classification\_report for Cluster 1 || DecisionTreeClassifier()

I am passing each cluster data to the decision tree and calculating the precision, recall, f1-score using classification\_report

precision recall f1-score support

0.0 0.93 1.00 0.97 14

1.0 1.00 0.40 0.57 5

2.0 1.00 1.00 1.00 449

3.0 1.00 1.00 1.00 532

accuracy 1.00 1000

macro avg 0.98 0.85 0.88 1000

weighted avg 1.00 1.00 1.00 1000

## Classification\_report for Cluster 2 || DecisionTreeClassifier()

precision recall f1-score support

0.0 0.78 0.62 0.69 414

1.0 0.55 0.72 0.62 268

2.0 0.97 1.00 0.98 31

accuracy 0.67 713

macro avg 0.76 0.78 0.76 713

weighted avg 0.70 0.67 0.68 713

Cluster wise, the decision tree model performed pretty good on only Cluster # 1. It was not that good on the other two clusters.

## Classification\_report for Cluster 0 || RandomForestClassifier()

precision recall f1-score support

0.0 0.94 0.19 0.31 81

1.0 0.77 1.00 0.87 255

2.0 1.00 0.44 0.61 16

accuracy 0.78 352

macro avg 0.90 0.54 0.60 352

weighted avg 0.82 0.78 0.73 352

Score : 0.7840909090909091

## Classification\_report for Cluster 1 || RandomForestClassifier()

precision recall f1-score support

0.0 1.00 0.43 0.60 14

1.0 0.00 0.00 0.00 5

2.0 0.98 0.98 0.98 449

3.0 0.97 1.00 0.99 532

accuracy 0.98 1000

macro avg 0.74 0.60 0.64 1000

weighted avg 0.97 0.98 0.97 1000

Score : 0.977

## Classification\_report for Cluster 2 || RandomForestClassifier()

precision recall f1-score support

0.0 0.66 0.98 0.79 414

1.0 0.88 0.22 0.35 268

2.0 0.97 1.00 0.98 31

accuracy 0.69 713

macro avg 0.84 0.73 0.71 713

weighted avg 0.76 0.69 0.63 713

Score : 0.6942496493688639

The accuracy and recall of the clusters 0 and 1 is better than 2. But, the F1- score is higher for the cluster 2 RandomForestClassifier()

### **Execution Time:**

DecisionTreeClassifiers() were quicker than the RandomForest. DecisionTree took nearly 3 secs for running the cluster whereas RandomForest took almost 20 secs.

# **Future Work / Critique**

In this project I have created the trained models based on the training data from the csv files. I have created the below model files and saved them on the file system.

enc.pickle file -- This file holds the encoding information of the class column

kmeans\_clustering.sav – This file contains the ML Kmeans model created for the clustering

Decision\_Tree(<Cluster\_Number>) – The File with this type of name contains the decision tree model

RandomForest((<Cluster\_Number>) – The file with this type of name contains the RandomForest model.

By using the above files, I can build the pipelines that can take the raw files as the input, process the data like divide it into the clusters, apply hyper tuned machine learning models on the clusters and finally predict the outcomes.

The above system/pipeline can then be created into a Flask or any other kind of application or API deployed locally or cloud. This application can be consumed by front end clients such as a web app, http clients etc. They can also be used as a part of a much bigger system such as a multi disease diagnostic / detection system.

## **Critique:**

This was one of the most interesting and learning journey of my life. Machine learning, Data Science and other important concepts learned and implemented by me in this project and entire course will help me become a better professional. I strongly believe that nothing is perfect and there is always an improvement to it. The same applies to my project. This is a new system implemented by me, I am sure that it can be improved in its later versions. More testing I do on this project, the more I may be able to enhance it. Below are some learnings or shortcomings which I believe exist in my dataset or model:

1. The dataset has many columns that do not contribute to the prediction, I have excluded them.
2. The dataset had more of the qualitative variables, I am new to them and always have trouble with them. I have done good work on handling them, but surely knows that they can be handled much efficiently
3. I have implemented KNeighborsClassifier() on this dataset, but it was taking a lot of time to train it when compared to random forest and decision tree classifiers
4. I could have used some additional models such as SVM, logistic regression or else Naïve Bayes.

# **Certification**

## Brief Background about myself:

I have a bachelor’s degree in Electronics and Communications Engineering. But, due to my interest in the Information Technology I chose to work in the Information Technology industry. Before coming to Arkansas for my Masters, I was a Microsoft certified Azure Administrator. Even though, I was in IT for so many years, my knowledge about Data Science, Machine Learning and python was little. In my first semester (Fall 2022), I realized that without learnings Python, I cannot do nothing in this field. I quickly learned basic python to use in some very basic tasks such as visualization, or at least understanding what is going on in data field.

Now, coming to this semester (Spring 2023), I am very glad that I took this course. The quick mini courses provided by Dr Pierce from CognitiveClass.ai were helpful. Mr. Bruce Bauer’s python classes strengthened my python skills. I thought that doing a Data Science project for the final assignment is not a cup of tea for me and I might end up wasting time, so I thought of taking a certification course from Udemy called **Complete 2022** **Data Science & Machine Learning Bootcamp** (<https://www.udemy.com/course/python-data-science-machine-learning-bootcamp/>). I finished this course last week i.e., 1st week May 2023 and received a course completion certificate.

## Complete 2022 Data Science & Machine Learning Bootcamp Certificate:

I have attached the certificate as a pdf, the file name is ***DataScience\_Certificate\_Abdul\_Basith.pdf***. I learned many topics related to Data Science specifically Machine learning during this course.

Though this course was for 41.5 hours. I have skipped some python classes because I was taking classes from Mr. Bruce. Overall, I have spent almost 38 hours on this course. These are some of the topics that I have learned from this course.

1. Data Cleaning and pre-processing using python
2. Regression
   1. Linear Regression
   2. Regularization (Ridge and Lasso)
   3. Random Forest Regression
   4. Support Vector Regressor
3. Classification
   1. Multivariable Regression
   2. Logistic Regression
   3. Decision Tree
   4. KNN
   5. Support Vector Machines (SVM)
   6. XG-Boost
   7. ADA boost
   8. Random Forest Classifier
4. Clustering,
   1. K Means
   2. Hierarchical Clustering
   3. Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

I did work on numerous python packages namely Scikit Learn, Pandas, NumPy, SciPy, TensorFlow, keras.

# **Time spent for the Data Science project and certification**

As explained in the previous section about my background. Transition from an Administrative background to the Data Analytics and Data Science is not an easy task. But I have put some remarkable efforts in the right direction under the guidance of Dr. Pierce. Below is a table describing my time spent for the data science project/certification.

|  |  |  |
| --- | --- | --- |
| Type of Work | Activity | Approximate Time Spent (Hrs.) |
| Certification | Data Science & Machine Learning Bootcamp Certificate | 38 |
| Project | Problem Formulation and dataset search | 3 |
| Understanding the dataset | 2 |
| Data Cleaning (Learning Imputation and new techniques) | 5 |
| Learning to implement clustering to improve performance and use multiple models on different clusters | 4 |
| Coding involved in implementing the clusters and dealing with multiple classifierss | 3 |
| Learning label encoding | 1.5 |
| Learning about the Seaborn, Matplotlib visualizations | 1 |
| Solving coding errors | 2 |
| Commenting the code | 0.5 |
|  | Learning and implementing metrics of RandomForest and DecisionTree Classifiers | 1 |
|  | Code optimizations | 2 |
|  | Preparing the report and other documentation | 3 |
|  | Total | 68 |

****

**References:**

Thyroid disorders. (2022, October 28). Johns Hopkins Medicine, based in Baltimore, Maryland. <https://www.hopkinsmedicine.org/health/conditions-and-diseases/disorders-of-the-thyroid#:~:text=The%20thyroid%20is%20a%20butterfly,the%20body%20to%20other%20hormones>