Springboard Data Science Career Track

Capstone Project I Final Report

Thyroid Classification.

Rahul Ambati

May 2018

1. Introduction.
2. Approach.
    a. Data Acquisition and Wrangling.
    b. Data Exploration and Inferential Statistics.
    c. Modeling and Data Analysis.
3. Findings.
4. Ideas for Further Research.
5. Client Recommendations.
6. Resources.

1. Introduction.

Definition:
Thyroid disease is a medical condition affecting the function of the thyroid gland. The symptoms of the disease vary depending on the type of thyroid disease.

Intention:
A physician needs to know the demographics associated with individuals suffering from thyroid disease and find what sector of people can be focused on so that they get admitted and get prior treatment.

Client:
Physicians who want to understand what kind of demographics, medication, etc. to consider while treating patients with thyroid disease so that the right group of people can get the proper care and treatment.

2. Approach.

The data science problem associated with the business problem is that of finding a model that can be used to classify a data point according to three classes: hyperthyroidism, hypothyroidism, and neither one of these. This section elaborates on the various parts of the approach that was followed.

a. Data Acquisition and Wrangling.

The data has been acquired from UCI ML dataset Thyroid disease and, in particular, this project will be focusing on the dataset known as ANN, which provides the information for characterizing thyroid disease.

The dataset consists of demographics (age, sex), about medication, current conditions (sick, tumor, goiter, etc.) some relevant measurements (TSH, T3, TT4, T4U, FTI) and disease category.

The dataset from the repository is clean, and not much cleaning or wrangling had to be done.

b. Data Exploration and Inferential Statistics.

The dataset contains information about three categories of the disease ***hyperthyroidism*** *(class 1),* ***hypothyroidism*** *(class 2) and* ***normal*** *(class 3).* ***Normal*** category means person does not suffer from the disease.

Exploring the data, much of the demographics does not suffer from the disease, as one would expect (see Figure 1).

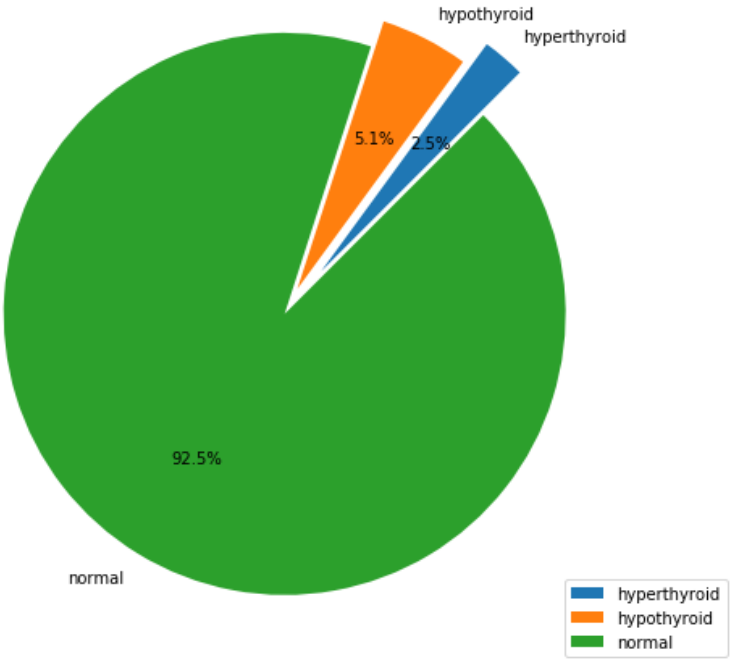


Figure 1. Proportion of classes in the ANN dataset

From the dataset a strip plot (see Figure 2) is made to view the dataset clustered according to the classes, age group and gender.

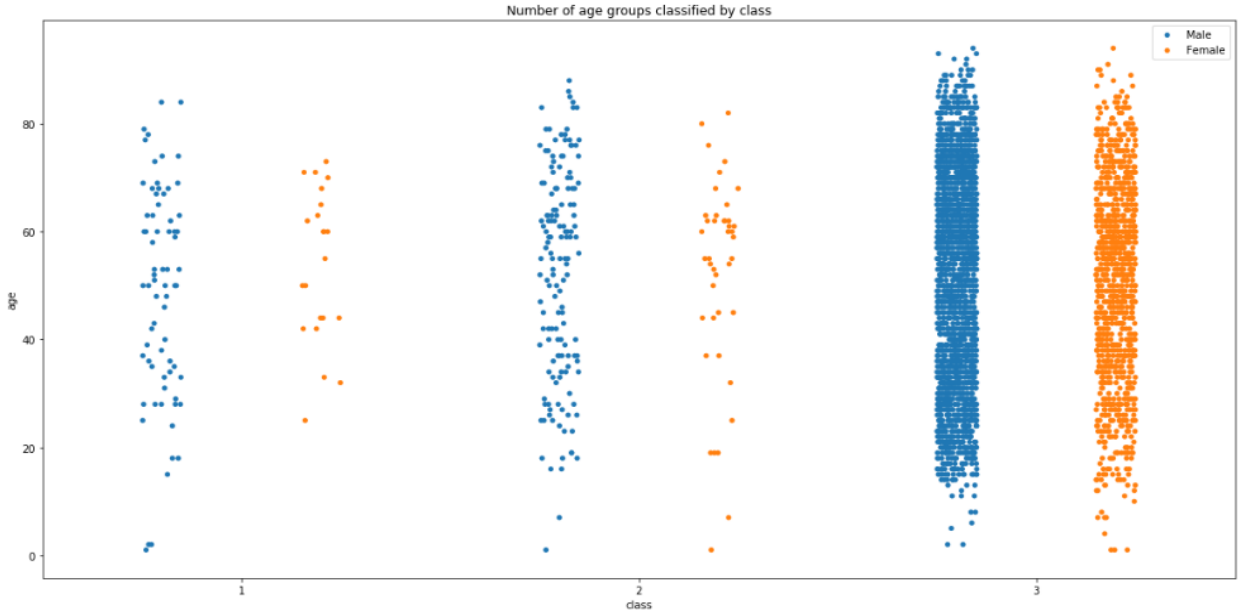


Figure 2. Strip plot showing the distribution of classes per age group and gender

Density plots (see Figure 3) show TSH, T3, TT4, T4U, FTI measurements and age.

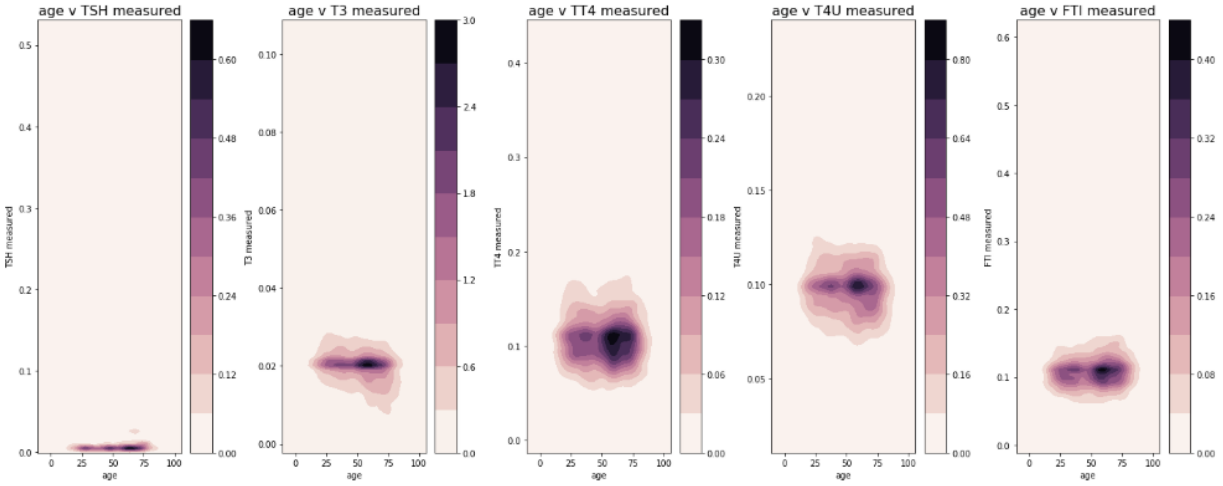


Figure 3. Density plots for various thyroid-related measurements and age

Hyperthyroid class was extracted from the data set and overlapped on the measurements to see where it lies in the distribution (Figure 4).

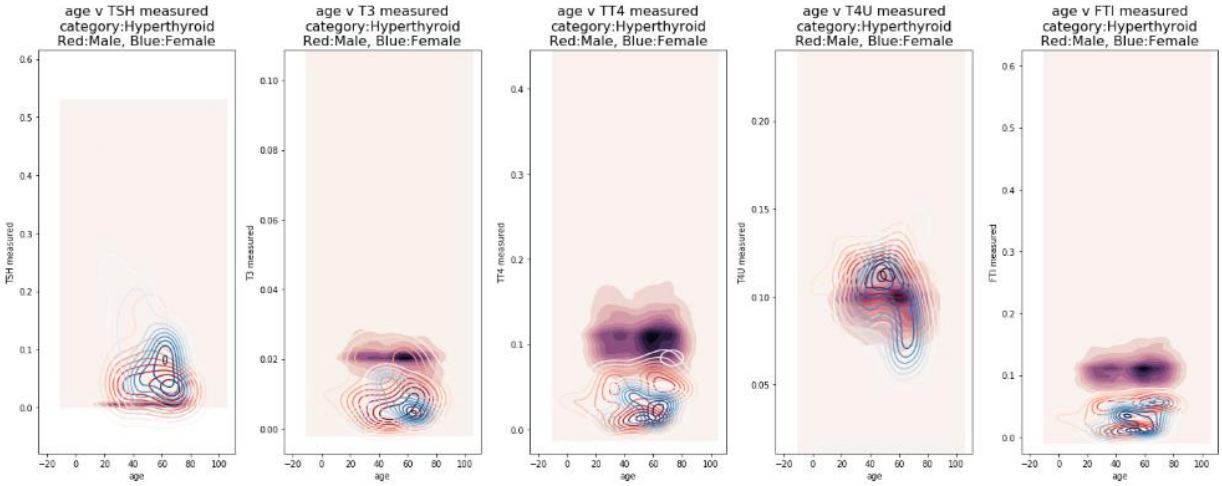


Figure 4. Density plot for various thyroid-related measurements, age and classes

A clear outlier can be seen where the ***hyperthyroid*** lies and we can make some inferences that it might lie outside the normal distributions.

A correlation heatmap shows how different variables are connected (Figure 5).

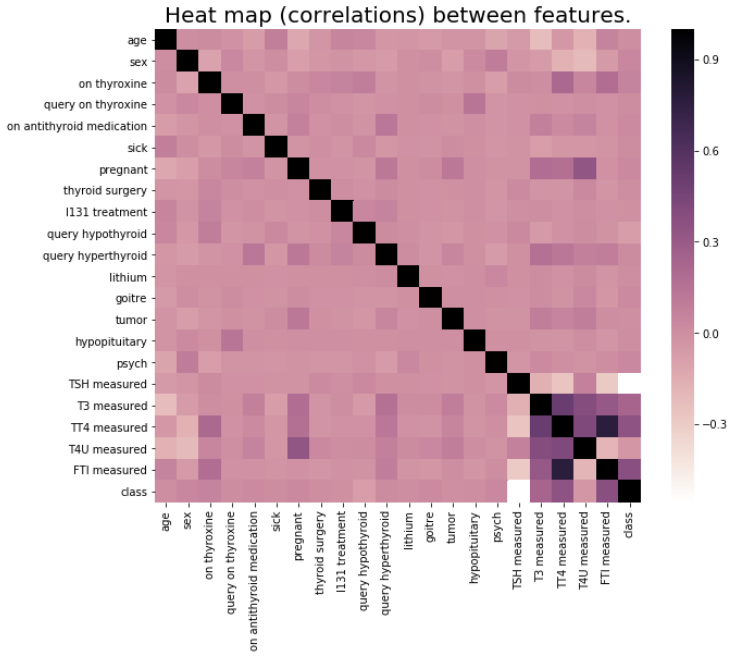


Figure 5. Correlation heatmap for some variables

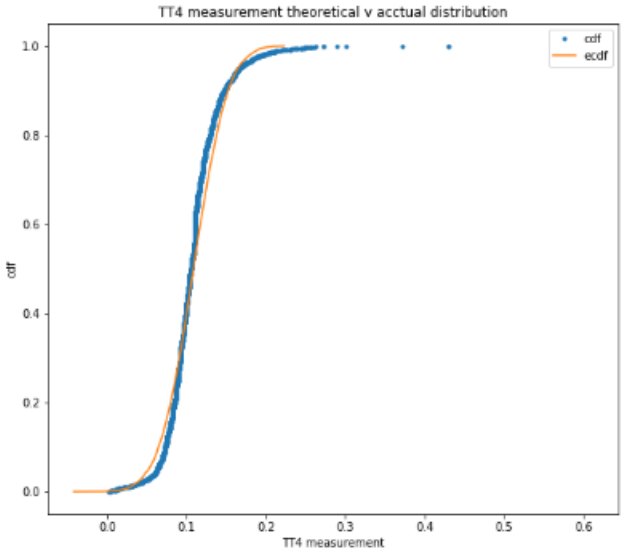
Looking at the correlation map (Figure 5) the measurements have a varied level of influence.

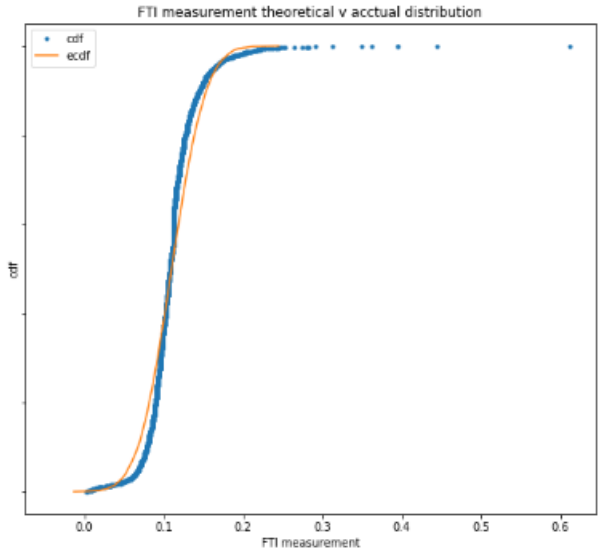
Statistical inference was used to determine if correlation between TT4 and FTI measurements is statistically significant or not.

Null hypothesis: Correlation between TT4 measured and FTI measured is zero.

Alternative hypothesis: Correlation between TT4 measured and FTI measured is greater than zero.

For the statistic to be significant we observe the p-value and if p-value is less than 0.05 (5%) null hypothesis is rejected.





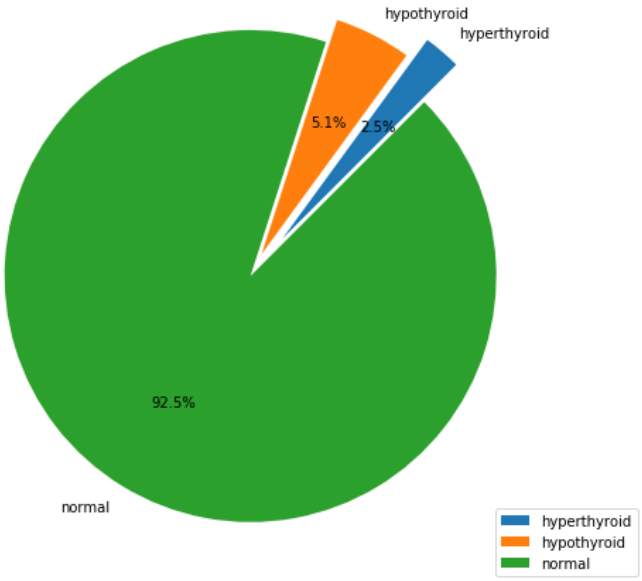
The above graphs show TT4 and FTI actual vs theoretical distributions and the actual distribution does not lie on the theoretical distribution. Also, a normal test has been performed to verify whether the distributions is normal or not and distributions are not normal.

A t-test was performed between TT4 and FTI to get the p-value, and after the t-test p-value is 0.017 (1.7%) which is below 0.05(5%), so the null hypothesis is rejected.

And after rejecting the null hypothesis, we can say that (alternative hypothesis) correlation between TT4 measured and FTI measured is greater than zero.

c. Modeling and Data Analysis.

From Figure 1, we see there are 3 classes, we can use supervised machine learning techniques to build a predictive model. Moreover, the classes are defined and we want to classify whether the person is suffering hyperthyroid or hypothyroid or none, it's a multi-class classification problem with three classes.



From the above figure we see a little percentage of people suffering from thyroid disease. It is a highly imbalanced data set. Most of the standard learning techniques are not well suited with imbalanced data set for training, because the model might be biased towards the majority class.

For imbalanced data sets we need to apply some re-sampling techniques during the training phase i.e., on the training data set. Re-sampling might be needed and may involve under-sampling the majority class (e.g., by usingRandomUnderSampler) or over-sampling the minority class (e.g., by using SMOTE – Synthetic Minority Oversampling Technique) or a combination of both (e.g., by using SMOTEENN – SMOTE and Edited Nearest Neighbors). The techniques mentioned here are implemented in the Python package "imbalanced-learn".[1]

And we have two separate data sets one for training and one for testing. We will be applying re-sampling techniques on the training data set.

Logistic Regression.

I have selected logistic regression as a base model for classification and after tuning the regularization hyper-parameter for L1 and L2 norms, the models did not classify as expected on the test data and the metrics (Table 1.a.) were low for class 1 and 2 as expected (has a bias towards majority class by looking at the class 3 metrics).

[1] http://contrib.scikit-learn.org/imbalanced-learn/stable/install.html

Since the model has poor performance, SMOTE was applied on training data to enhance the minority classes so that the performance might be improved. After the model got trained on the re-sampled data and tested on the test-data, the model performance (metrics) for the minority classes were improved significantly.

CLASS 1 CLASSIFICATION REPORT ON TEST-SET DATA			
CLASSIFIER	PRECISION	RECALL	F-SCORE
LogisticRegression C=1000 L1 regularization	75.949%	82.192%	78.947%
LogisticRegression C=1000 L1 regularization using SMOTE	64.356%	89.041%	74.713%
LogisticRegression C=1000 L2 regularization	79.730%	80.822%	80.272%
CLASS 2 CLASSIFICATION REPORT ON TEST-SET DATA			
CLASSIFIER	PRECISION	RECALL	F-SCORE
LogisticRegression C=1000 L1 regularization	75.862%	24.859%	37.447%
LogisticRegression C=1000 L1 regularization using SMOTE	70.732%	98.305%	82.270%
LogisticRegression C=1000 L2 regularization	70.000%	15.819%	25.806%
CLASS 3 CLASSIFICATION REPORT ON TEST-SET DATA			
CLASSIFIER	PRECISION	RECALL	F-SCORE
LogisticRegression C=1000 L1 regularization	95.746%	99.150%	97.418%
LogisticRegression C=1000 L1 regularization using SMOTE	99.903%	96.853%	98.354%
LogisticRegression C=1000 L2 regularization	95.293%	99.371%	97.289%

Table 1.a. Logistic Regression Classifier's metrics.

Over-fitting model analysis.

Models suffer from over-fitting, when the model performs well in training and performs poorly on test data it means the model over-fitted. From Table 1.b. we see how the two logistic regression models performed when one trained on training data and the other using SMOTE.

Initially the first model did perform well during training but after testing we can clearly the difference in precision and the difference between training and test is very much pronounced. Clearly over-fitted itself. Moreover Class 2 recall is very poor, it means it did a poor job in predicting Class 2.

The first model is not a good model, but the second model recall for Class 2 improved after training using SMOTE. Plus, over all recall is very good.

Still by looking at the training and test for the second model the difference in precision is pronounced. It clearly over-fitted itself for Class 1 during training.

LogisticRegression C=1000 L1 regularization

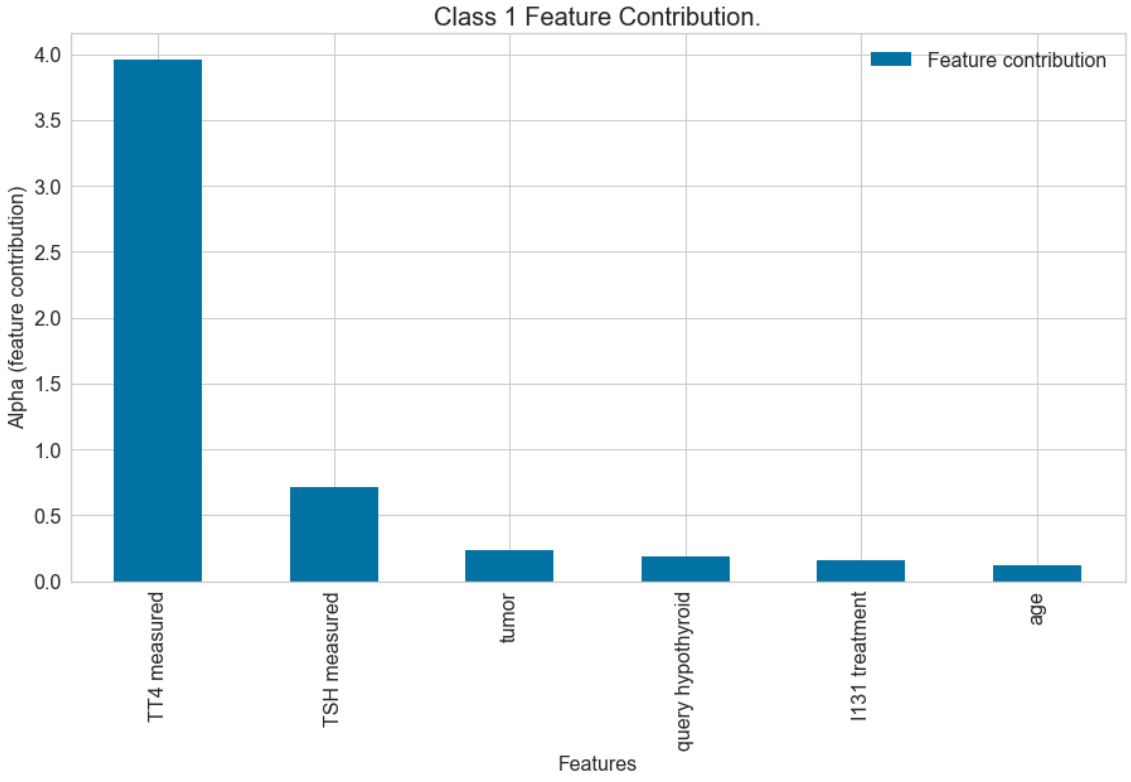
CLASS		PRECISION		RECALL		F-SCORE	
		TRAINING	TEST	TRAINING	TEST	TRAINING	TEST
CLASS 1	OVER-FITTING	87.234%	75.949%	88.172%	82.192%	87.701%	78.947%
CLASS 2		94.000%	75.862%	24.607%	24.859%	39.004%	37.447%
CLASS 3		95.948%	95.746%	99.799%	99.150%	97.836%	97.418%

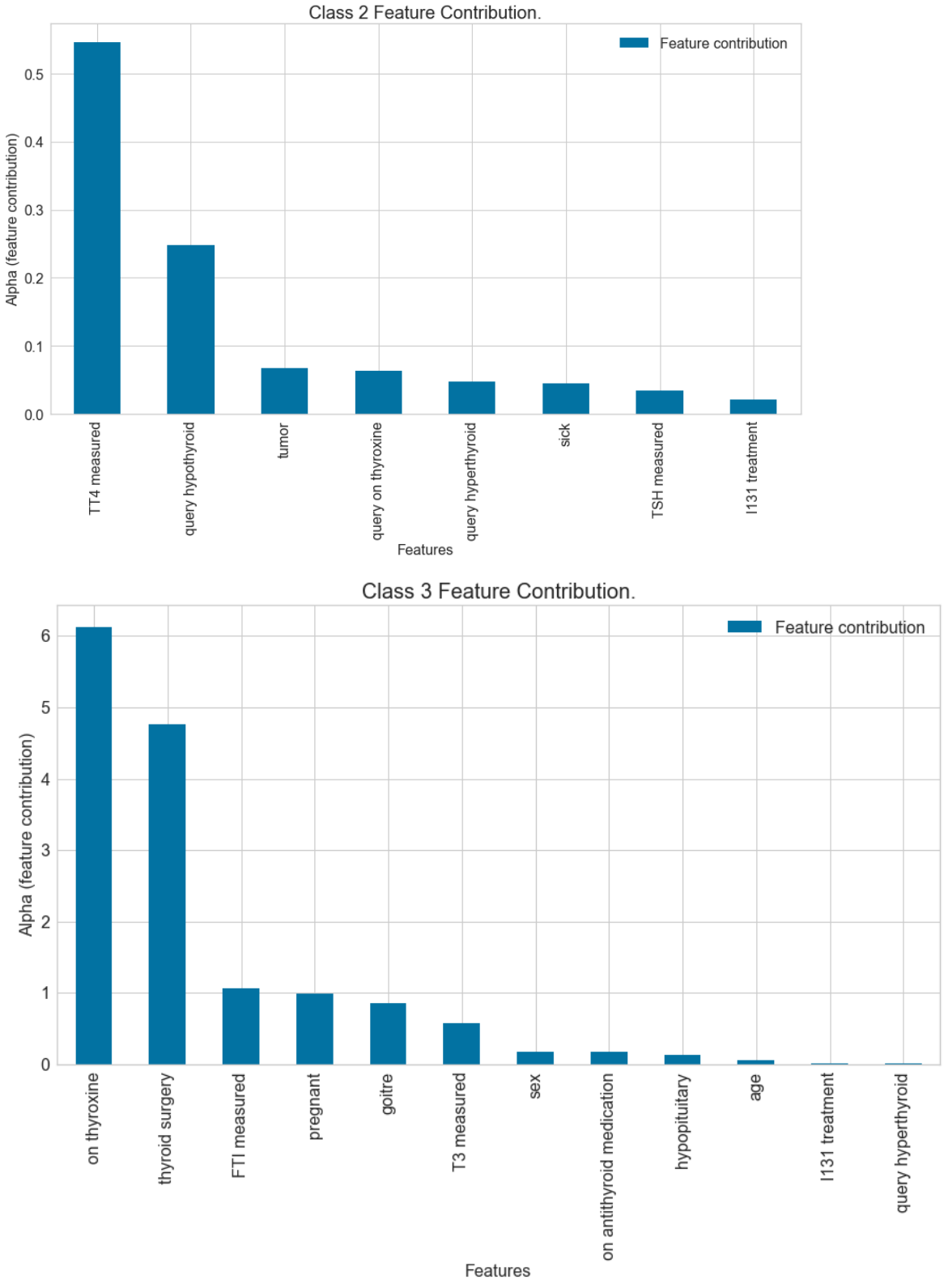
LogisticRegression C=1000 L1 regularization using SMOTE

CLASS		PRECISION		RECALL		F-SCORE	
		TRAINING	TEST	TRAINING	TEST	TRAINING	TEST
CLASS 1	OVER-FITTING	81.081%	64.356%	96.774%	89.041%	88.235%	74.713%
CLASS 2		77.593%	70.732%	97.906%	98.305%	86.574%	82.270%
CLASS 3		99.912%	99.903%	97.964%	96.853%	98.929%	98.354%

Table 1.b. Logistic regression model over-fitting analysis.

Feature importance.





Figure(s) 6. Logistic Regression classifier feature importance for each class.

Logistic regression using SMOTE is used to figure out the features enabled for prediction.

From Figure 6. We can see the which features (positive significance) contributed in predicting classes.

Random Forest.

After making a base model for reference, RandomForestClassifier from scikit-learn[2] was used.

[2] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

CLASS 1 CLASSIFICATION REPORT ON TEST-SET DATA

CLASSIFIER	PRECISION	RECALL	F-SCORE
RandomForestClassifier	86.747%	98.630%	92.308%
RandomForestClassifier after tuning hyperparameters using SMOTE	86.905%	100.000%	92.994%
RandomForestClassifier using RandomUnderSampler (under-sampling)	48.026%	100.000%	64.889%
RandomForestClassifier using SMOTE (over-sampling)	87.952%	100.000%	93.590%
RandomForestClassifier using SMOTEENN	82.955%	100.000%	90.683%

CLASS 2 CLASSIFICATION REPORT ON TEST-SET DATA

CLASSIFIER	PRECISION	RECALL	F-SCORE
RandomForestClassifier	90.576%	97.740%	94.022%
RandomForestClassifier after tuning hyperparameters using SMOTE	93.651%	100.000%	96.721%
RandomForestClassifier using RandomUnderSampler (under-sampling)	64.706%	99.435%	78.396%
RandomForestClassifier using SMOTE (over-sampling)	92.670%	100.000%	96.196%
RandomForestClassifier using SMOTEENN	87.624%	100.000%	93.404%

CLASS 3 CLASSIFICATION REPORT ON TEST-SET DATA

CLASSIFIER	PRECISION	RECALL	F-SCORE
RandomForestClassifier	99.841%	99.087%	99.463%
RandomForestClassifier after tuning hyperparameters using SMOTE	100.000%	99.276%	99.637%
RandomForestClassifier using RandomUnderSampler (under-sampling)	100.000%	94.525%	97.185%
RandomForestClassifier using SMOTE (over-sampling)	100.000%	99.245%	99.621%
RandomForestClassifier using SMOTEENN	100.000%	98.741%	99.367%

Table 2. Random Forest Classifier's metrics.

From Table 2. Random forest classifier with default parameters and training on train data did have a better prediction on the test data than the base model (logistic regression).

Since the data set is imbalanced SMOTE was applied on the train data and used it to train the classifier, and after testing on the test data we get a 100% recall on class 1,2 and a relatively good precision (**green**).

Furthermore, various re-sampling techniques were used to train the classifier (under-sampling of majority class, mix of under and oversampling classes- SMOTEENN). After using SMOTE (over-sampling) data for training the model performed really good on test data.

The next step was to increase the precision for the classifier, so hyper-parameters were tuned for the classifier and trained on the SMOTE data. After testing on the test data, we see a similar prediction (**blue**) to that of the default parameter random forest classifier. Moreover, we see the metrics match closely for each class.

From Table 2. We observe that the model with default parameters trained on SMOTE data have a good recall (100%) for minority classes and really high f-scores. This model is a good candidate for predicting disease amongst random forest classifiers.

Feature analysis.

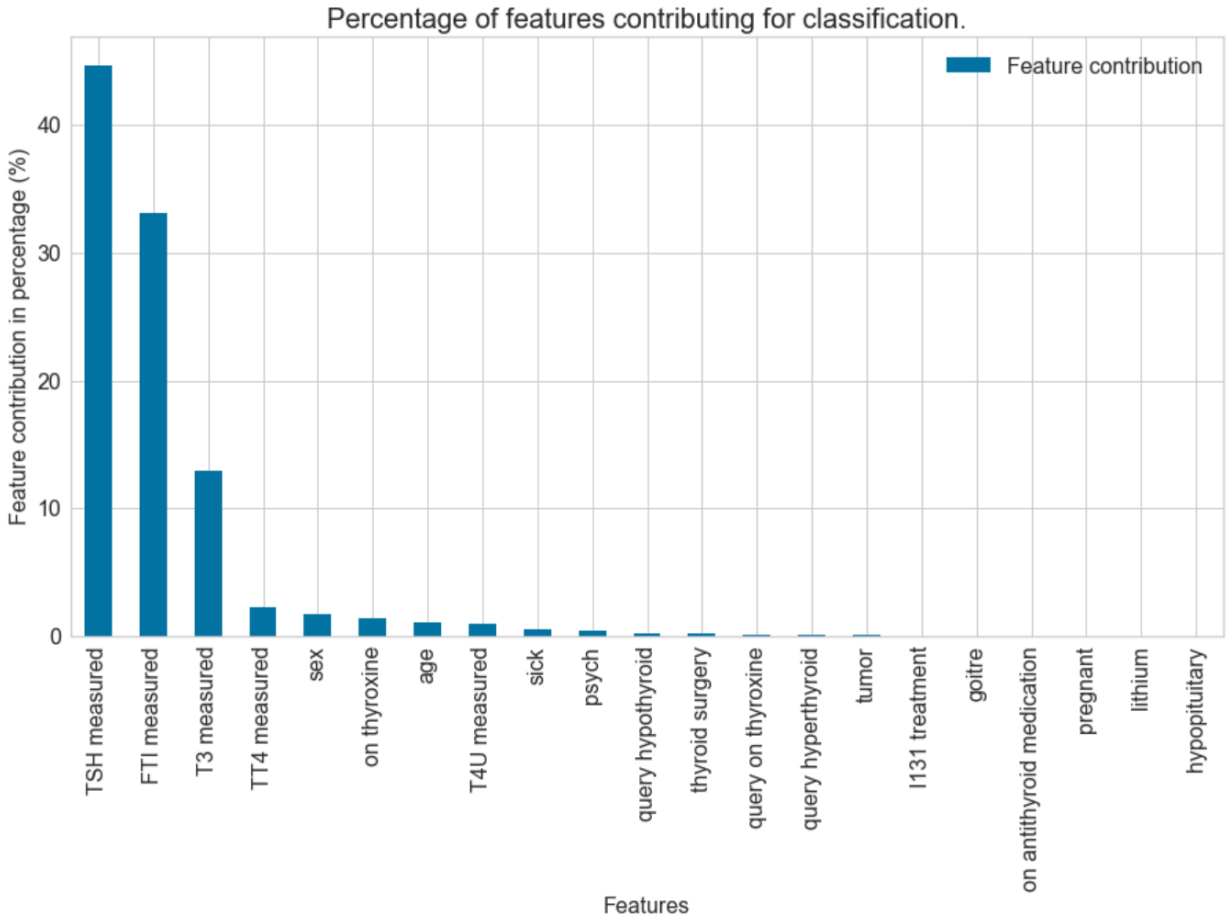


Figure 7. Random forest classifier feature importance.

From Figure 7. we see which features contributed the most in predicting different classes. TSH, FTI, T3 measurements alone contributed towards 90% of the prediction.

Model Comparison and Recommendations.

We made a base model and after evaluating the model we created new models to predict whether the person suffers from any thyroid disease or not.

CLASS 1 CLASSIFICATION REPORT ON TEST-SET DATA

CLASSIFIER	PRECISION	RECALL	F-SCORE
LogisticRegression C=1000 L1 regularization	75.949%	82.192%	78.947%
RandomForestClassifier using SMOTE (over-sampling)	87.952%	100.000%	93.590%

CLASS 2 CLASSIFICATION REPORT ON TEST-SET DATA

CLASSIFIER	PRECISION	RECALL	F-SCORE
LogisticRegression C=1000 L1 regularization	75.862%	24.859%	37.447%
RandomForestClassifier using SMOTE (over-sampling)	92.670%	100.000%	96.196%

CLASS 3 CLASSIFICATION REPORT ON TEST-SET DATA

CLASSIFIER	PRECISION	RECALL	F-SCORE
LogisticRegression C=1000 L1 regularization	95.746%	99.150%	97.418%
RandomForestClassifier using SMOTE (over-sampling)	100.000%	99.245%	99.621%

Table 3. Base model vs best model metrics.

The above Table 3. Shows how the base model stacks up with the best model (green showing the best model, yellow showing the base model metrics). A lot has improved w.r.t base model. Moreover, looking at the Recall being 100% meaning it's predicting right classes, but there are few false positives looking at the precision.

Random forest classifier model with default parameters trained using SMOTE data is the best model to recommend and predict whether the person suffers from hyperthyroid or hypothyroid or not.

3. Findings.

We explored the data set to understand what kind of diseases can be categorized and their share. From data exploration we see only a slight percentage of people suffer from thyroid diseases and the rest do not. It made the project interesting in identifying those extreme cases. It is similar to predicting fraud transactions or flight cancellations, where the probabilities are very low.

From initial exploration looking at TSH, FTI, T3, TT4, T4U measurements we were able to make some predictions where the edge cases might lie, and also by looking into correlations to understand it.

We used supervised classification algorithms in predicting the diseases. Since the samples are imbalanced, re-sampling of the data was done to improve the performance of the classifier. Re-sampling provided with more additional data points (synthetic data) for the edge cases and it maximized the edge cases so that the model can learn it.

A Random forest classifier model with training on synthetic (re-sampling) data performed the best among the classifiers on test data, and also gave insights into which factors are affecting them the most.

We can recommend this model to the physicians to predict or classify whether the patient has thyroid disease or not.

4. Ideas for Further Research.

This dataset has limited features, no feature engineering was done. However, if more data or features were to be added we can enhance the power of the model.

Here are some of the possibilities that can be incorporated for further research.

1. If new data comes with existing features and additional features such as height, weight or other measurements, we can use those additional features to verify its contribution, correlation, etc.

2. If patient history is maintained or tracked, we can add more data and engineer more features.

3. After treatment is administered, measurements are to be noted before and after so to understand the effectiveness of the treatment, which can provide additional features for different types of treatments.
4. With more data and features we are not only trying to make better predictions, but understanding connections between the features.

5. Client Recommendations.

1. From Figure 7 we see few features that are dominating the prediction in classifying thyroid disease, physicians need to understand why those measurements are high is it because of genetics, heredity, dietary, life style or other factors. Thus, physicians can focus on those measurements and provide some treatment.

2. The model can aid the physician as a preliminary diagnosis tool in decision making but still physicians need to look at the test results to avoid any false positives or false negatives.

3. Model was built on data provided, there could be other unknown factors affecting the patients which was not accounted for or might not considered. I would recommend the physicians to look more into the diagnosis i.e. looking at their blood pressure, heavy metals in blood stream, hormones and many more.

6. Resources.

Tools:

1. Anaconda
2. Jupyter Notebook
3. GitHub

Language:

4. Python 3.6

Packages:

5. Numpy
6. Pandas
7. LogisticRegression
8. RandomForestClassifier
9. DecisionTrees
10. SMOTE
11. RandomUnderSampler
12. SMOTEENN
13. Supervised learning.
14. Model selection.

Packages for visuals:

15. Matplotlib
16. Seaborn
17. OOB plot for ensemble.
18. http://www.scikit-yb.org/en/latest/api/classifier/confusion_matrix.html
19. http://www.scikit-yb.org/en/latest/api/classifier/classification_report.html

Blogs:

20. Building a logistic regression model.
21. Statistics stack exchange.
22. Data science stack exchange.
23. Analytics Vidya.