**Project Title: Prediction model by the utilization of distinctive features of patients with thyroid disease**

**Understanding of Data:**

1) Data is sourced from keel machine learning repository.

2) The dataset has a total of 7200 instances.

3) There are 21 features totally.

4) Out of 21, 6 are continuous variables.

1. Remaining 15 variables are categorical in nature.

**EDA and data Pre-Processing:**

1) First we checked for null values, there are no null values in the dataset.

2) Next, we checked for duplicated values. There are total of 71 duplicated columns.

3) The duplicated rows are dropped from the data-frame. We are left with 7129 instances now.

4) The data\_types of features are checked. "Age" is on a scale from 0 to 1. It is converted to integer scale.

5) The number of instances of each class is analyzed.

1. The classes are labeled as 1,2,3 & by using Label encoding, we convert them to 0,1,2 for classification.

**Balanced or unbalanced classification ?**

The number of samples with Hyper-thyroidism = 166

The number of samples with Hypo-thyroidism = 368

The number of samples with Negative case of Thyroidism = 6595

Therefore the number of instances of the minority classes are much lesser than that of the majority class (almost 1:2:40 Proportion.

**Therefore the problem is very close to being a highly unbalanced classification.**

**Tackling Unbalanced issue ?**

Balancing techniques are not applied here as the data in a real worl application is naturally highly unbalanced.

Therefore, through this problem we seek to find the best algorithm that can generate the highest accuracy, while dealing with the highly unbalanced case.

**Problem 1: Regression**

**Target Audience / Beneficiary:** Medical testing facilities & labs.

**Aim :** Find the missing values in medical report

**Target Variable :** To find the value of TT4 (Total thyroxine) by using other variables such asT3, T4 and TSH.

**Benefits :**

Validating the existing medical data to check for possible errors or outliers in medical reports.

Use of regression to predict and generate the total thyroxine level in blood using the other variables

**Correlation** :

Identification of the Input features for the regression problem:

The values of FTI, T3 and T4U have high correlation with TT4 (Total Thyroxine Values)

Therefore are selected as input variables for regression analysis.

**primary meteric for evaluation of best regression model - R^2 Value for test datasets.**

Linear Regression :

R^2 score for test dataset = 0.9179

Decision Tree Regressor :

R^2 score for test dataset = 0.9589

Bagging Regressor :

R^2 score for train dataset = 0.9968

R^2 score for test dataset = 0.9819

Random Forest Regressor:

R^2 score for train dataset = 0.997

R^2 score for test dataset = 0.9825

For the Regression algorithms, Random forest regressor and Bagging regressor both have almost identical R^2 scores, therefore are best algorithms.

**The regression problem is solved with a very high accuracy rate .**

**Problem 2** : Classification Algorithm

**Target Audience / Beneficiary**: End users - Patients

**Aim :** To use classification algorithms for Classify the patients into 3 classes based on the thyroid medical data set.

**Target Classes:**

“0” - Hyperthyroidism : Hyperthyroidism is an overactive thyroid (when it produces too much thyroid hormone).

“1” - Hypothyroidism : Hypothyroidism is an under-active thyroid (when it does not produce enough). Almost 4.6% of US population is suffering from it.

“2” - Negative.

**Solution:** Problem - 2 Classification.

**Best Performing Algorithm:**

* Random Forest Classifier with tuned hyper-parameters.
* The random forest algorithm initially resulted in an f-1 score of .86 for the minority class '1' in the test instance, and 1.0 for the train instance.
* However, after tuning hyper-parameters the variance was eliminated and the f-1 score of the worst performing minority class was increased to .90 for both test and train instances.

**Next best performing algorithm :**

* Decision tree classifier with parameters from GridSearchCV:
* Initially, the default parameters of decision trees causes a high variance problem with over-fitting.
* The f-1 scores of train and test for minority class "1" is 1.0 and 0.71 respectively.
* This problem was solved with GridSearchCV and passing the resultant parameters.
* This was able to improve performance and greatly decrease the high variance problem:
* The f-1 scores of train and test for minority class are now .87 and .83

**Analyzing the Performance of SVM :**

* First one we used was SVM with default parameters. Resulted in an f-1 score of 0,0 for the two minority classes.
* Then, implementing Kernel SVM - resulted in f1 score of 74 for the first minority class (o).
* However, minority class (1) still had f-1 score 0.
* Then used Scikit-learn’s GridSearchCV.
* Grid search was able to improve the score of the second minority class from zero to 70 !
* That too with minimal variance.
* However, as the continuous variables (thyroid hormones and indexes) impact varies from individual to individual patient, there is an overlap of features.
* Therefore, as SVM is a geometric based algorithm, The hyperplane fails to accurately divide the features for classification. Hence, SVM fails to achieve an f-1 score that competes with Random Forests.

**Analyzing the Performance of probability based models:**

* All probability based models including logistic regression, LDA, QDA and Gaussian Naive Bayes show very poor f-1 scores for the minority class, sometimes tends to zero.
* The major reason behind this could be the highly imbalanced data and few instances of the minority classes.
* As probability based algorithms require combination of input features for the various outputs, there are not enough instances in the train subset of the model.
* Additionally, the assumptions of Naive Bayes about the independence of the feature variables does not hold true.
* Therefore the above mentioned models do not perform well.

**Analysis : Why Random Forest Performed Best ?**

* Random forests, with tuning the hyper-parameters shows the greatest improvement in the result.
* Accuracy of 90% and 90% for the minority classes - eliminating any variance overall accuracy of the model stands at 98.5%
* Random forests use the Collective outcome of multiple decision trees.
* Prevention of over-fit for better model performance.
* Great with higher dimensional data, here we use all 6 features. Ability to handle outliers - binning and reducing error - by down-sampling majority class and growing trees on balanced datasets

**Learning and Recommendations:**

* Grid search is an amazing tool, use it. Saves time and prevents monotony.
* Choose algorithms based on the data you use.
* Try solving the problem with the simplest algorithms and gradually increase complexity. Prioritize the concept of Occam's Razor.
* What to consider when selecting the best model ? performance evaluation metrics and Also consider how important the accuracy is, what difference does an improvement of 0.1% make ?
* In our case, we chose a bio-informatics problem, therefore an accuracy improvement of even 0.1% will result in accurate classification of 10 patients in a dateset of 10,000 cases.
* Therefore the requirement of the computational resources are not considered when choosing the best model in our case.
* Specifically in-case of highly imbalanced classification problems, consider the minority classes.
* In Highly imbalanced case - overall accuracy has very less weight-age compared to the individual accuracy of the minority classes. For instance, LDA was able to achieve an overall accuracy of 94% but it is of no use as the f-1 score for minority class '1' is ZERO !|

**References:**

Code references from the DATA 602 Lecture notes of Dr. Masoud Soroush, Associate GPD MPS Data Science Program, UMBC.

**Other Project Contributors:**

Sruthi Batchala, MPS Data Science Graduate Student, UMBC.

Avanti Dasare, MPS Data Science Graduate Student, UMBC.

Luis Galvan, MPS Data Science Graduate Student, UMBC.