Diabetes Prediction Over Telephonic Health Survey

Name:	**Aditya Kishore**
Registration No./Roll No.:	21011
Institute/University Name:	IISER Bhopal
Program/Stream:	Data Science and Engineering
Problem Release date:	August 17th, 2023
Date of Submission:	November 19th, 2023

1 Introduction

The aim of this project is to develop a model that accurately predicts diabetes Classes(0,1,2), where class labels are 0 (no diabetes or only during pregnancy), 1 (prediabetes), 2 (diabetes) from the data collected over Telephonic Health Surveys. The Data set Contains 228312 data Points in the training data. The Number of Data Points per each class is as follows: **Class(0): 153866, Class(1): 3334, Class(2): 25449** and 25368 data points in the test data. There are no missing values in the training dataset So, No Imputation is Required. On Initial Analysis of the Problem statement, the target variables for this Problem are Discrete, Thus, we have to predict the category/class for a new data point/instance. Therefore it is a Classification Problem. In this Project we will be using different feature engineering techniques and classification models to accurately classify Diabetes class into one of the 3 classes. The Figure Below Represents the distribution of number of instances over the classes.

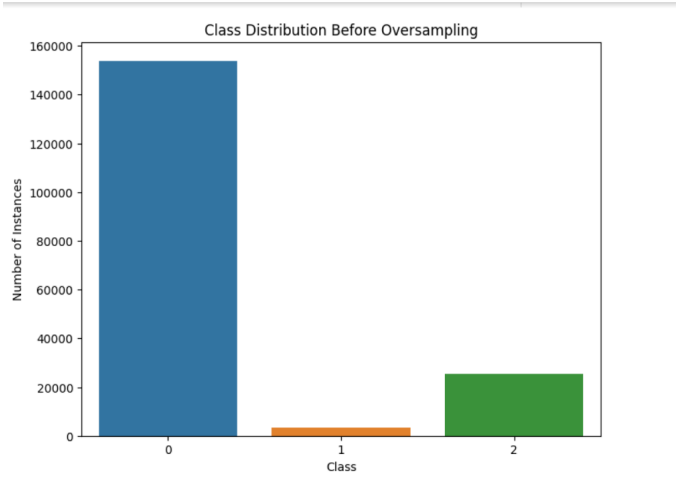

Figure 1: Overview of Data Set

2 Methods

The proposed method of approaching the problem includes Train-test split, oversampling, feature selection based on SelectKBest and correlation analysis along with appropriate classification techniques enhanced with hyperparametric tuning and displays a heat map which helps to visualize the correlation matrix between the features. The dataset was splitted into training and testing sets for the evaluation of the model. 20 percent of the dataset used for testing phase and the remaining 80 percent of the dataset is used for training. The random state parameter sets the random seed for reproducibility.

Setting a random seed ensures that every time the code is executed, the same random split will be generated.Here in this problem the random state is setted to 42. Stratified sampling is used as stratified sampling ensures that the distribution of class labels in the original dataset is maintained in both the training and testing sets. The SelectKBest method is further used to select top K features which are important, SelectKBest is a method for univariate feature selection. It selects the top k features based on univariate statistical tests, and chi2 (chi-squared) is the specific statistical test is used.As we can see from the above Data visulaization from the figure 1 that there is a significant class imbalance is present in the dataset. The code comprises of RandomOverSampler from the imbalanced-learn library to rectify the issue of class imbalance,The RandomOverSampler is initialized with a random state for reproducibility. It takes the training data (training data) and corresponding labels (training cat) as input and oversamples the minority class to match the majority class.After the application of oversampling the class distribution is again printed using Counter. which is visualized in the figure 2. The code uses Logistic Regression classifier to address the challenge of classification, logistic regression predicts the probability that a given instance belongs to a particular category. The output of logistic regression is transformed using the logistic function (also known as the sigmoid function), which maps any real-valued number to the range between 0 and 1. The model is then trained on oversampled training data. During the training process, an optimization algorithm(Gradient Descent) is used to adjust the coefficients, and the logistic function is applied to transform the linear combination of predictor variables into a probability. The trained model was then evaluated on the test set with the use of metrics such as confusion matrix, precision, Recall, and F1 Score. The Macro-averaged Precision, Recall are computed and Precision, Recall and F1 Score are computed for each class. The Code enhanced the classification using Hyperparameter tuning of the classifier. It uses GridSearchCV to execute a cross-validated grid search with stratified k-fold(n-splits = 5),By preserving the class distribution in each fold, stratified k-fold helps prevent biases in model evaluation and ensures a more accurate assessment of a model's performance, particularly in scenarios with uneven class frequencies. The hyperparameters are optimized to ensure that Macro-averaged F1 score is Maximised After the Grid Search the best Performing classifier and its associated hyperparameters are obtained.

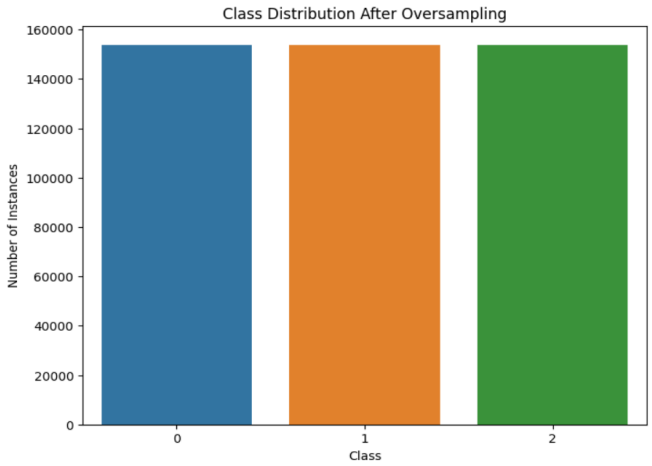

Figure 2: Overview of Data Set after oversampling

Table 1: Performance Of Different Classifiers Using RandomOverSampling Features

Classifier	Precision	Recall	F-measure
Adaptive Boosting	0.47	0.39	0.40
Decision Tree	0.40	0.39	0.40
Logistic Regression	**0.47**	**0.46**	**0.45**
Random Forest	0.40	0.39	0.39
Multinomial Naives Bayes	0.39	0.39	0.39

3 Experimental Setup

The Results for OverSampling for all the classifier is duly shown in Table 1 as the RandomOverSampler used there was an another technique for oversampling called as SMOTE as on Comparing the RandomOverSampler is giving more better results in comparision to SMOTE as SMOTE is Preferable for Smaller datasets and RandomOverSampler is preferred for Larger datasets though RandomOverSampler did not performed upto the mark as the Data being Imbalanced, Further The evaluation criterion Precision, Recall and F1 Score is obtained for each classifier, GridSearch was implemented on Logistic Regression classifier,then the grid search was trained on oversampled data and also cross validated on the original data using the grid, then the Best Classifier along with the best parameter of grid search ('C': 1.0, 'class-weight': 'balanced', 'dual': False, 'fit-intercept': True, 'intercept-scaling': 1, 'l1-ratio': None, 'max-iter': 1000, 'multi-class': 'auto', 'n-jobs': None, 'penalty': 'l2', 'random-state': None, 'solver': 'liblinear', 'tol': 0.0001, 'verbose': 0, 'warm-start': False) is obtained out of the code and Displays the Logistic Regression classifier and its parameters as the best classifier as it outworks among all the listed classifiers.

4 Results and Discussion

Actual Class	Predicted Class		
	0	1	2
0	34527	527	3413
1	650	22	162
2	4465	163	1734

Decision Tree

Actual Class	Predicted Class		
	0	1	2
0	34164	36	4267
1	571	5	258
2	3168	9	3185

Logistic Regression

Figure 3: Confusion Matrices for Decision Tree and Logistic Regression

Actual Class	Predicted Class		
	0	1	2
0	33883	1336	3248
1	637	33	164
2	4460	230	1672

Random Forest

Actual Class	Predicted Class		
	0	1	2
0	32318	793	5356
1	605	17	212
2	4091	117	2154

Multinomial Naives bayes

Figure 4: Confusion Matrices for Random forest and Multinomial Naives Bayes

Actual Class	Predicted Class		
	0	1	2
0	37464	0	1033
1	753	0	81
2	4998	0	1364

Adaptive Boosting

Figure 5: Confusion Matrix for Adaptive boosting

Since From the part of visualization and statistics it is clearly stating that the dataset is imbalanced heavily. To tackle this issue of imbalanced classes RandomOverSampler was introduced which creates the copies of the minority class, and balances the imbalanced class distribution and prevents the model from being biased towards the class that is in majorirty. The logistic regression model is trained to find the values of b0, b1,.....bk that maximize the likelihood of the observed data. During

the training process, an optimization algorithm is used to adjust the coefficients, and the logistic function is applied to transform the linear combination of predictor variables into a probability.Once the model is trained, it can be used to make predictions by evaluating the logistic function for new instances. If the predicted probability is greater than a certain threshold (typically 0.5), the instance is classified as belonging to the positive class; otherwise, it is classified as belonging to the negative class.Hyperparameters for regularization such as l2 allows us to knob the complexity of the model, n-splits prevents overfitting,especially when we are encountering the data that is oversampled, After the oversampling each class is at same number of instances i.e 153866 instances.Because of the massive number of class 1 and class 2 samples that have been created, the training data is now overfitting to the model, making the goal of increasing precision, recall, and f1 score of the cross validation data counterproductive. Figure 3,4,5 shows the experimental results of all the appropriate models used in the form of the confusion matrix.Since the Logistic Regression model has the highest F1, precision and Recall it outworks all the model that are considered for this framework.

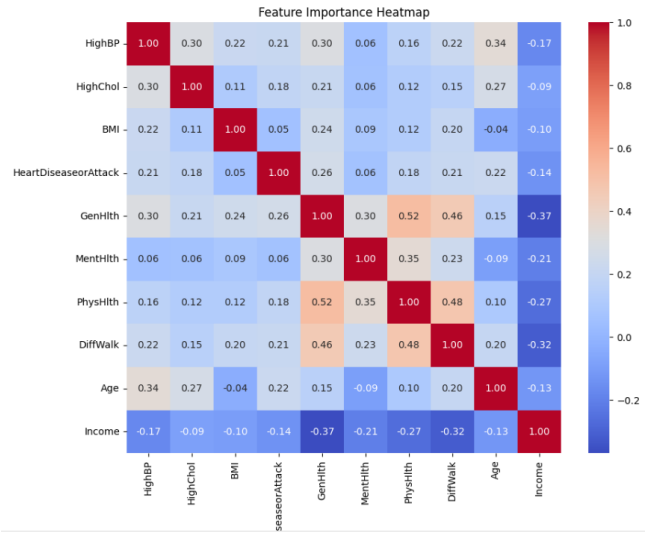

Figure 6: Heat Map addressing correlation between the features

5 Conclusion

In the future, this framework's research may concentrate on creating oversampling methods that dynamically adjust to features and changes in the dataset. It may also examine methods for automating and modifying sampling strategies and class weights in light of the dataset's characteristics. Additionally, as RandomOverSampler performs poorly in oversampling, new techniques to generate synthetic data or feature vectors that are physiologically appropriate to fit a specific class label can be investigated.

References

[1] Pradeep Kandhasamy, S. Balamurali, Performance Analysis of Classifier Models to Predict Diabetes Mellitus[2015] ScienceDirect, https://doi.org/10.1016/j.procs.2015.03.182

[2] Iyer, A.; Jeyalatha, S.; Sumbaly, R. Diagnosis of Diabetes Using Classification Mining Techniques. Int. J. Data Min. Knowl. Manag. Process. 2015, 5, 1–14, https://www.aircconline.com/ijdkp/V5N1/5115ijdkp01.pdf

You can find the code on GitHub at: `https://github.com/Adityakishore09/ML_Project_Diabetes_Prediction_over_telephonic_health_survey`