Development and Validation of Cardiovascular Disease Risk Prediction Tool: A Machine Learning Approach

Richa Sethi

January 2021

Introduction

To stay relevant in today's age, health insurance companies need to accelerate their digitalization and data augmentation journeys. Especially now, with COVID-19 lockdowns and ongoing physical-distancing protocols it is imperative to rethink the current insurance underwriting ways as the paramedic home visits to conduct medical exams have become highly undesirable. In this environment, risk assessment must shift toward more remote, data-driven models, while distribution must shift from in-person interactions to more online interactions.

The goal of this project was to develop a cardiovascular disease risk assessment screening tool to predict risk of heart attack and stroke among adults using self-reported information. This would in turn could be used to get a probability of an individual developing cardiovascular disease based on a short questionnaire and evaluate risk score.

Data analysis, modeling, and model evaluation was performed in Python enabled by numpy, pandas, matplotlib, seaborn, scikit-learn, imblearn etc. All the notebooks related to this project can be found at https://github.com/richasethi3/CVD_Prediction.

Dataset Overview

The dataset was obtained from the National Health and Nutrition Examination Survey (NHANES) – a population-based program of studies designed to assess the health and nutritional status of adults and children in the United States. It is conducted by Center of Disease Control's (CDC) National Center for Health Statistics (NCHS) on a sample of the U.S. population of all ages to reflect the nation overall, rather than individual states or counties. 5-year (2005-2016) demographics, examination, laboratory and questionnaire data was collected from NHANES for 60, 936 individuals. The original variables from the dataset were renamed for simplicity. **Table 1** below shows the original variables, renamed variables and variable description of the dataset collected.

Table 1: Table of variables and their description

S.No.	Original Variable	Renamed Variable	Description
1	SEQN	seqn	Unique sequence number
2	RIAGENDR	gender	Gender
3	RIDAGEYR	Age	Age in years
4	RIDETH1	ethnicity	Ethnicity/race
5	INDFMIN2	income	Household annual income
6	BPXPULS	pulse_regular	Pulse regular or irregular?
7	BPXSY1	sysbp	Systolic BP (mm Hg)
8	BPXDI1	diabp	Diastolic BP (mm Hg)
9	BMXBMI	bmi	Body Mass Index (kg/m**2)

10	BMXWAIST	waistcircum	Waist circumference (inches)
11	LBDHDD	hdl	HDL cholesterol (mg/dl)
12	LBDLDL	ldl	LDL cholesterol (mg/dl)
13	LBXTR	trigly	Triglycerides
14	LBXTC	totchol	Total cholesterol
15	DIQ010	diabetes	Doctor diagnosed diabetes
16	KIQ022	kidney_fail	Doctor diagnosed failing kidneys
17	MCQ160B	congestive_fail	Doctor diagnosed congestive heart failure
18	MCQ160C	coronary_disease	Doctor diagnosed coronary disease
19	MCQ160D	angina	Doctor diagnosed angina
20	MCQ160E	heart_attack	Doctor diagnosed heart attack
21	MCQ300A	fam_history	Family history of heart disease
22	SMQ020	smoking	Smoking history

Data Processing and Cleaning

The data for individuals over 20 was filtered and used for the analysis. This eliminated close to 44% of the data and there was data remaining for 34,180 individuals. Furthermore, the ldl and trigly columns were missing over 55% of the data. As shown in **Figure 1**, the missing data did not show any clear trend, so the missing rows were dropped. The remaining missing columns, namely sysbp, diabp, waistcircum, pulse_regular, bmi and income were imputed by either median or mode, as appropriate.

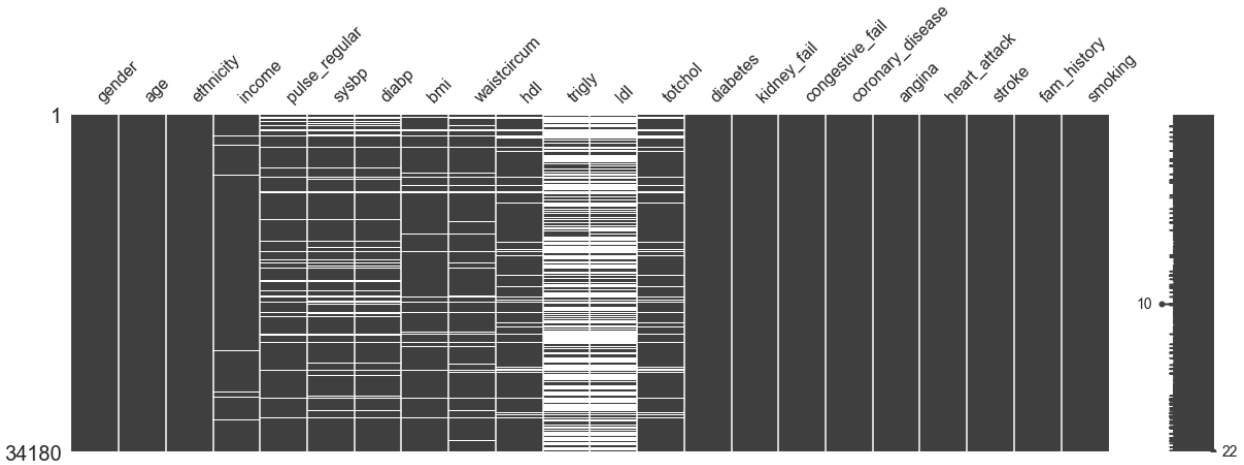

Figure 1: Missing number matrix for the dataset

The features in original data were transformed to 0 or 1, based on whether the condition was True or False. In addition, the variables used for gender were 0 for male and 1 for female. The target variable called CVD_risk was created which was 0 if congestive_fail, coronary_disease, angina and heart_attack were all False, and 1 if either of the aforementioned variables were True. The distribution of features after cleaning and processing are shown in **Figure 2.**

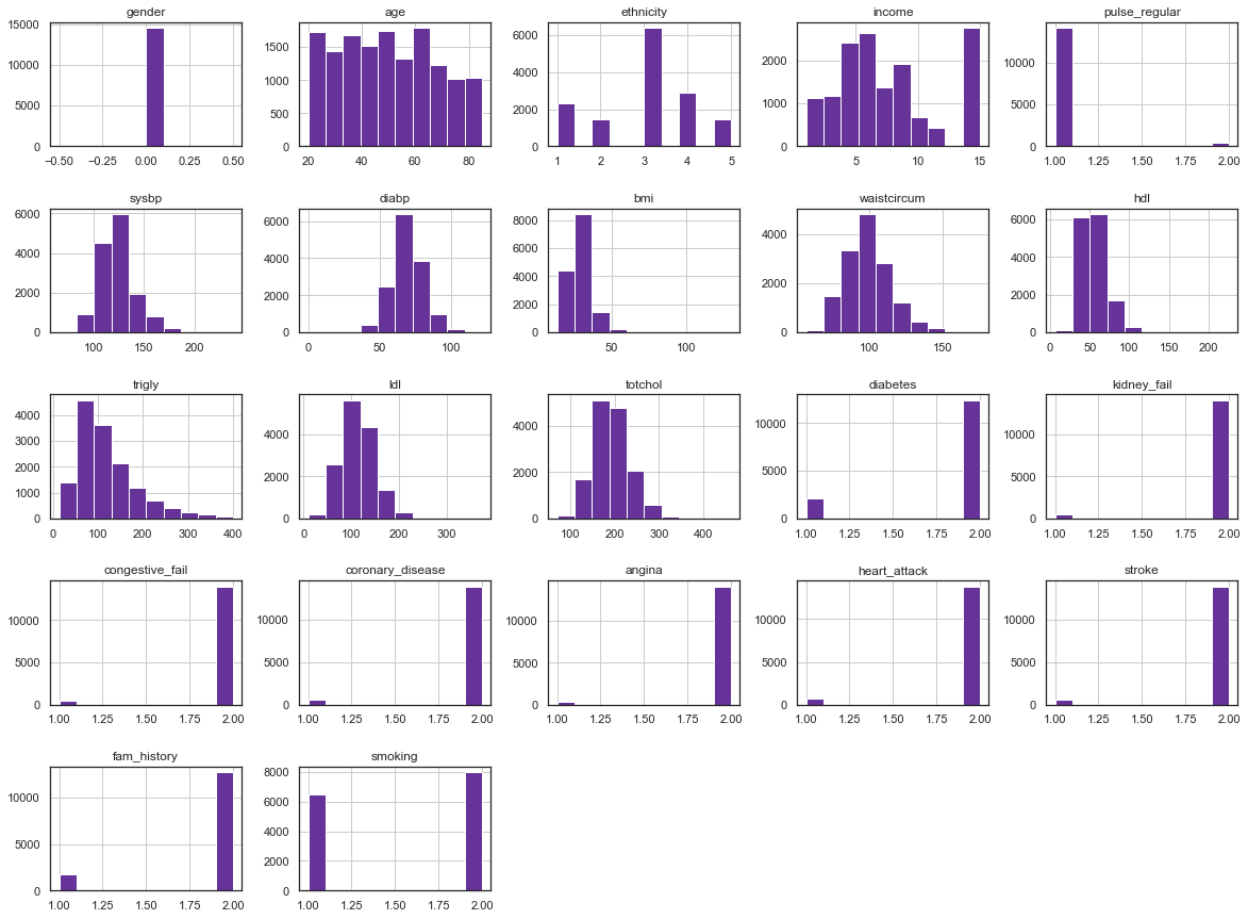

Figure 2: Distribution of features

Further, some of the continuous and categorical variables were transformed as shown in **Table 2**. It wasn't sure if this was the ideal mapping, the effect of the mapping was evaluated later.

Table 2: Original and transformed (binned columns)

Columns	**Transformed columns**
age	age_cat (20s, 30s, 40s, 50s, 60s, 70s & above)
income	income_cat (lowest, lower middle, middle & over)
sbp, dbp	hypertension_cat (normal:0, high:1)
bmi	bmi_cat(underweight, ideal, overweight)
waistcircum, gender	waistcircum_cat (normal:0, high:1)
trigly	trigly_cat (normal:0, high:1)
ldl	ldl_cat (normal:0, high:1)
hdl	hdl_cat (normal:0, high:1)
totchol	totchol_cat (normal:0, high:1)

Target Characteristics

The target feature CVD_risk showed unequal distribution of classes with 12,878 negative classes and 1,611 positive classes, corresponding to 1:8 ratio between positive and negative class. The plot of the class distribution is shown in **Figure 3**.

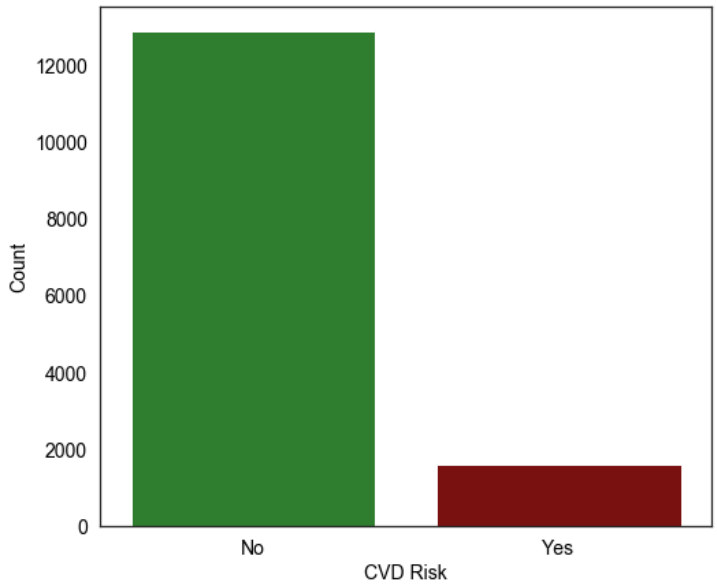

Figure 3: Target class distribution

Imbalanced datasets pose a challenge for predictive modeling as most of the machine learning algorithms used for classification are designed around the assumption of an equal number of examples for each class. This results in models that have poor predictive performance, specifically for the minority class and are essentially treated as noise. Therefore, specialized sampling techniques were employed to handle class imbalance.

Additionally, this dataset showed a high degree of feature overlap between classes. An analysis of kernel density plots revealed that the data is similarly distributed on all the features as shown in **Figure 4**. Most negative CVD_risk features had similar values as positive CVD_risk. This high degree of overlap may forecast difficulty for the models to predict CVD risk with high precision.

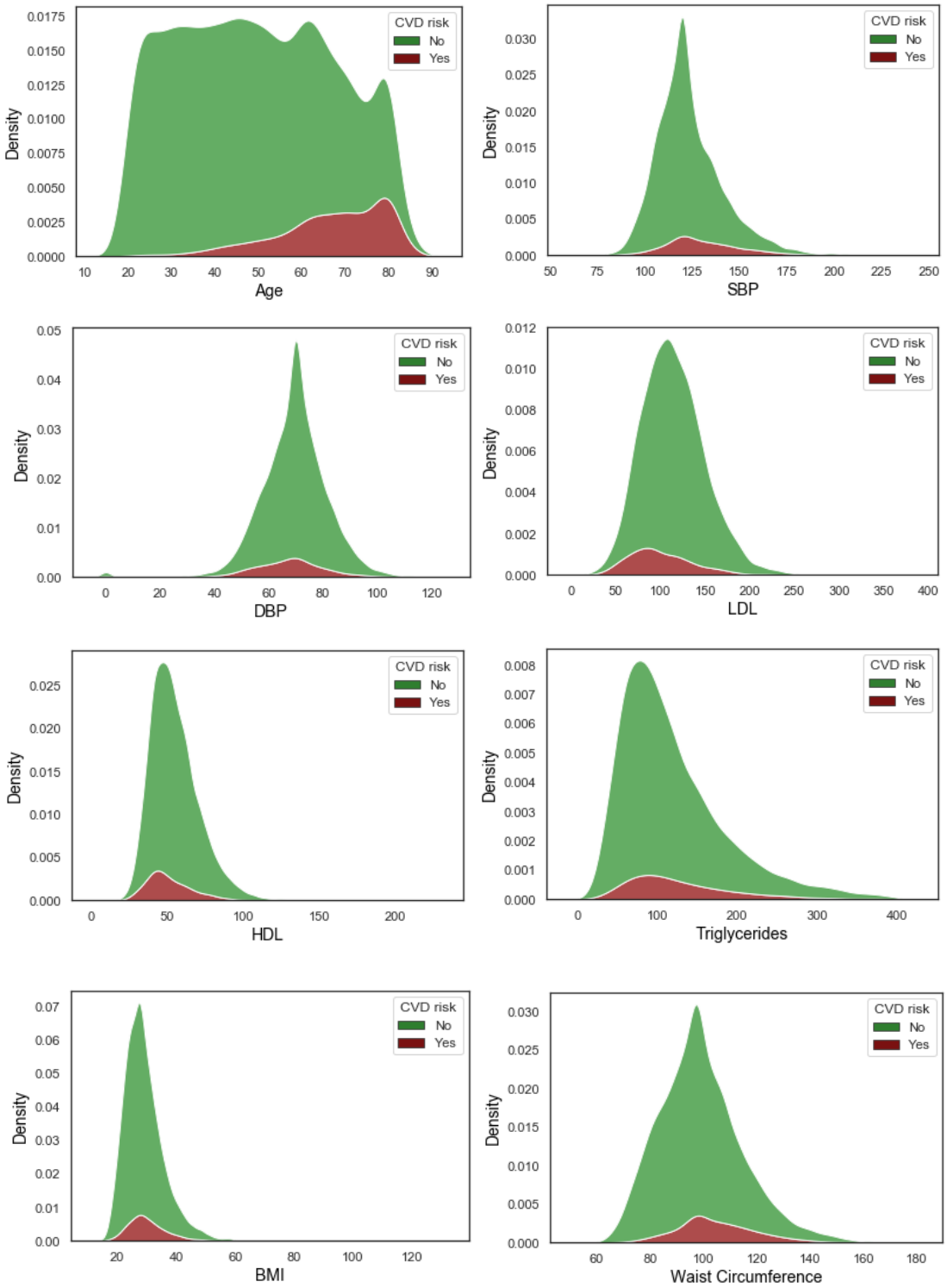

Figure 4: Overlap of target classes for different features

Data Analysis

The distribution of different features with CVD risk is shown in **Figure 5**. In general, the age of individuals with CVD risk is higher. In addition, the systolic blood pressure, BMI, waist circumference and triglycerides is higher as well for individuals at CVD risk, which intuitively makes sense. However, the diastolic blood pressure, ldl and total cholesterol is lower for individuals at CVD risk, contrary to the expectation.

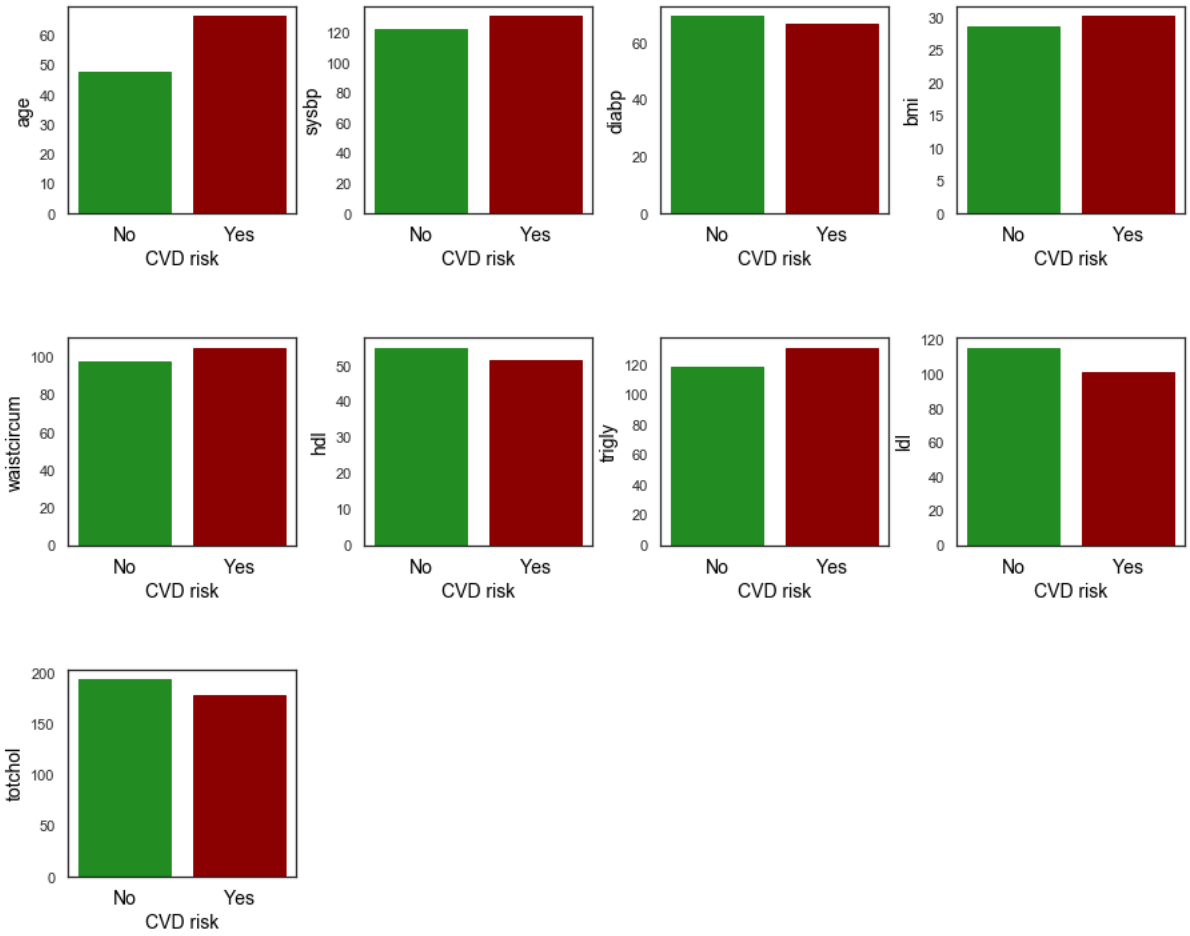

Figure 5: Count of features by CVD risk

Further visualization was done via boxplots as shown in **Figure 6**.

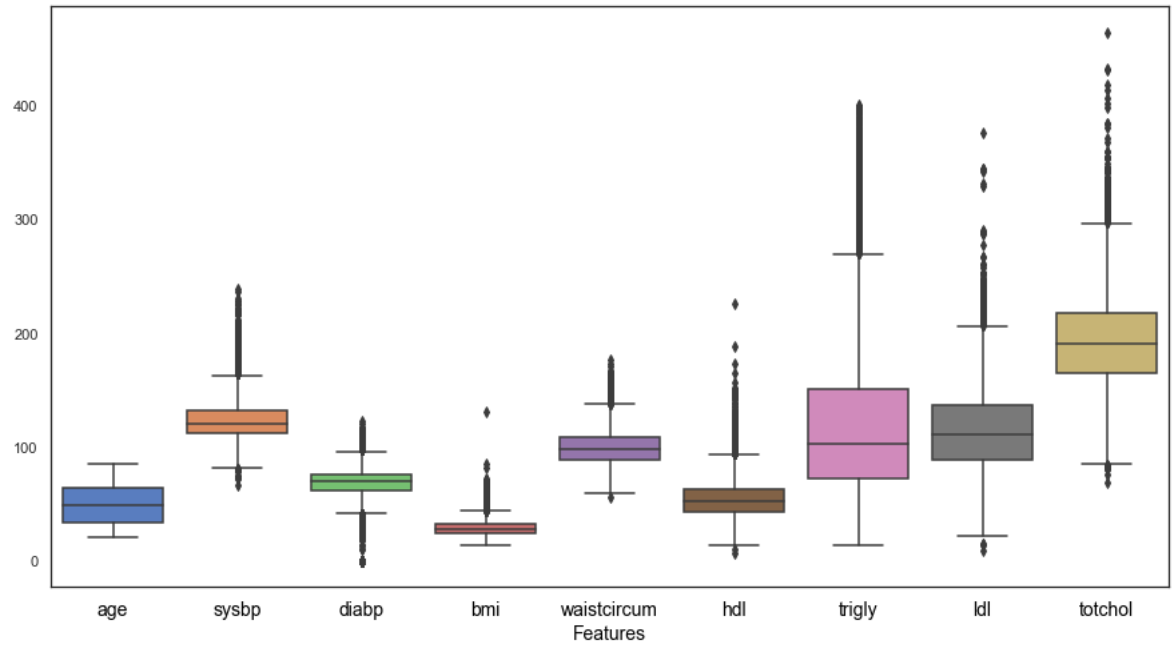

Figure 6: Boxplot of features

Figure 7 shows the odds ratio plot for different features, which shows the top 8 features with age, ethnicity, BMI, ethnicity and income as the most important features.

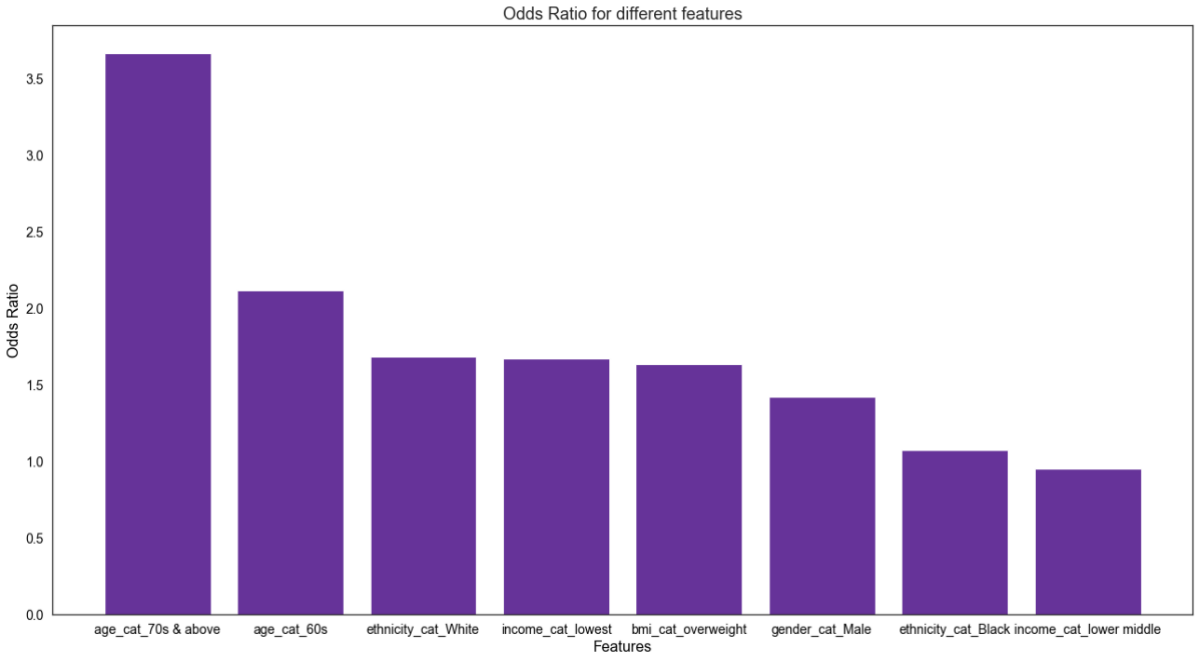

Figure 7: Odds ratio for top 8 features

Figure 8 shows the heatmap of different variables. Understandably, it's not very robust owing to some correlated variables, however, as can be observed strong correlation is seen among HDL and age, systolic blood pressure and age, CV risk and age, diabetes and age and diabetes and CVD risk.

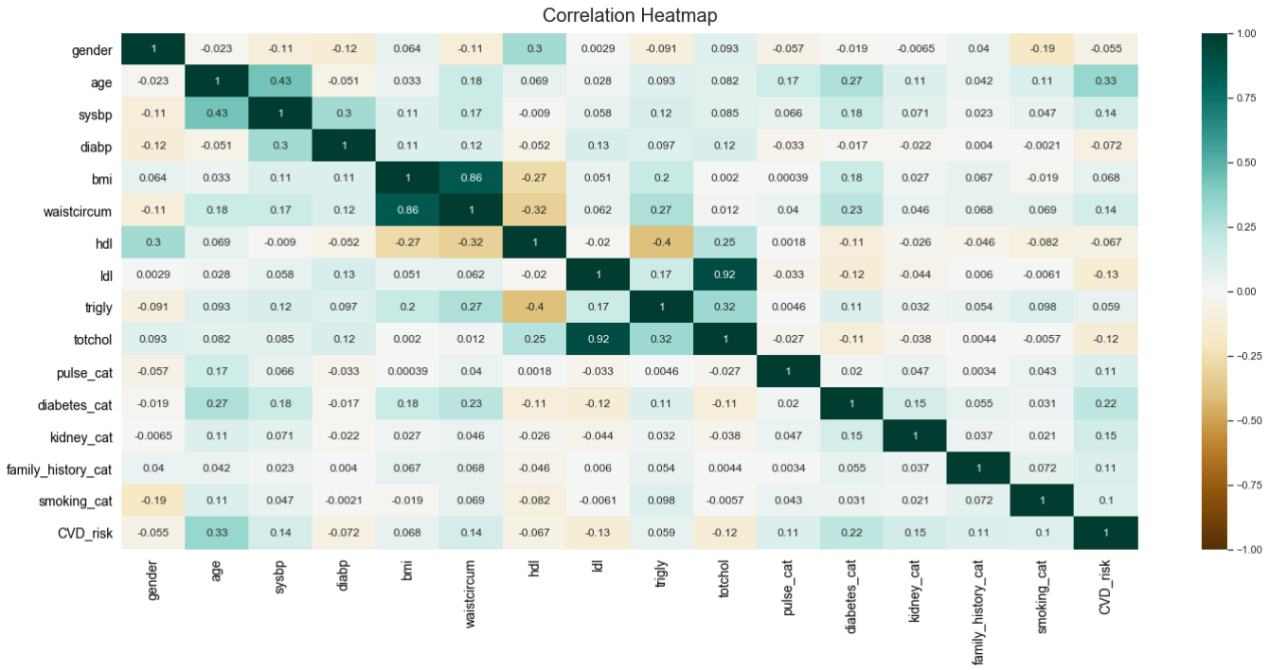

Figure 8: Correlation heatmap of the features

Data Preprocessing and Training

In our dataset, we kept the absolute features as well as the dummy features. It's expected to have features which are highly correlated i.e. somewhat linearly dependent with other features. These features contribute very less in predicting the output but increase the computational cost. Therefore, the correlation between the features was estimated and the features which have correlation coefficient greater than or equal to 0.6 were eliminated. The dropped features were waistcircum (correlated with bmi), totchol (correlated with ldl, hdl and trigly) and dummy features - hypertension_cat, waistcircum_cat, hdl_cat, trigly_cat, ldl_cat, totchol_cat, gender_cat_Female, gender_cat_Male, income_cat_lowest and bmi_cat_overweight.

The data was split and stratified on CVD_risk to ensure equal distribution of target under each set. The distribution of the data under each set is given in **Table 3**.

Table 3: Training, validation and testing set distribution

Dataset	Distribution %	Target %
Training	60	11
Validation	20	11
Testing	20	11

Naïve Baseline

A random guess model was run which showed poor metrics with ROC AUC as 0.52 and PRC AUC as 0.11 shown in **Figure 9**. With no machine learning, only 13% of the individuals with CVD risk were identified out of which only 14% of those predicted are actually true. A naive approach clearly did not provide much value.

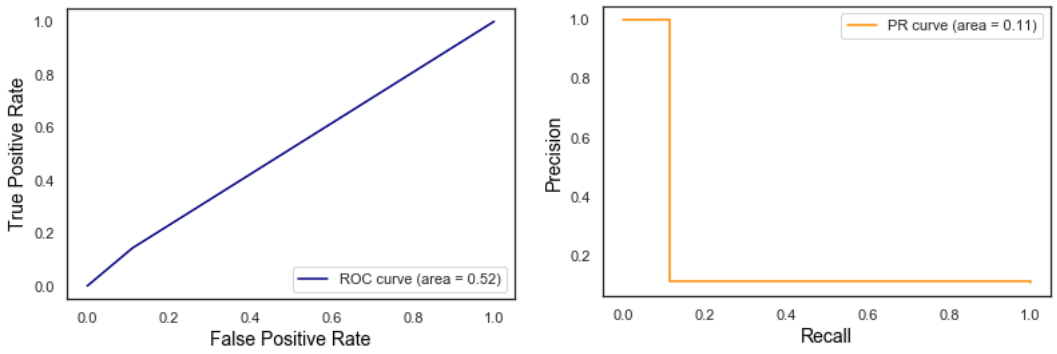

Figure 9: ROC curve and PR curve for naïve baseline model

Base Model Evaluation

To understand if machine learning is up to the task of improving on naïve predictions, classification models were assessed without hyperparameter tuning on both training and validation sets. The performance of four classification algorithms – Logistic Regression, Support Vector Machine Random Forest, and k-Nearest Neighbors were analyzed using the default scikit-learn APIs.

The performance evaluations were all conducted over the same train-test split to ensure consistency in the evaluations. For a classification problem, there are a number of metrics to consider like ROC AUC, F1-score, precision, recall and PR AUC, however for an imbalanced dataset, it could be misleading to focus on accuracy alone, since the dataset if dominated by individuals with no CVD risk. The aim of this work was to improve all the metrics with the focus on improving precision and recall.

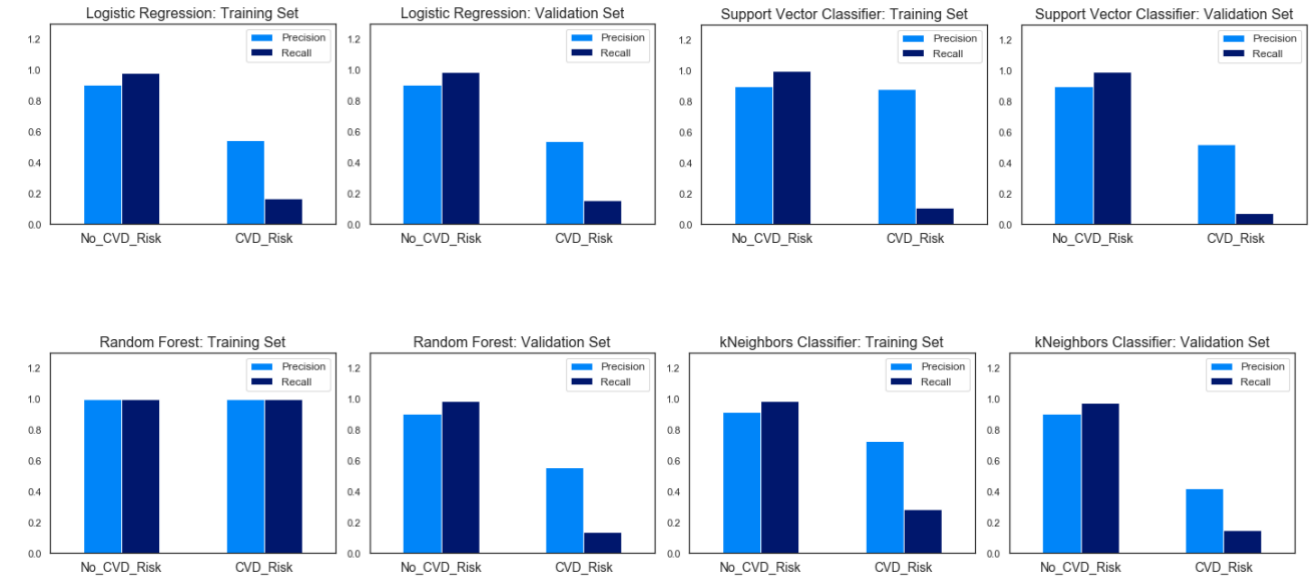

Figure 10: Baseline precision and recall scores

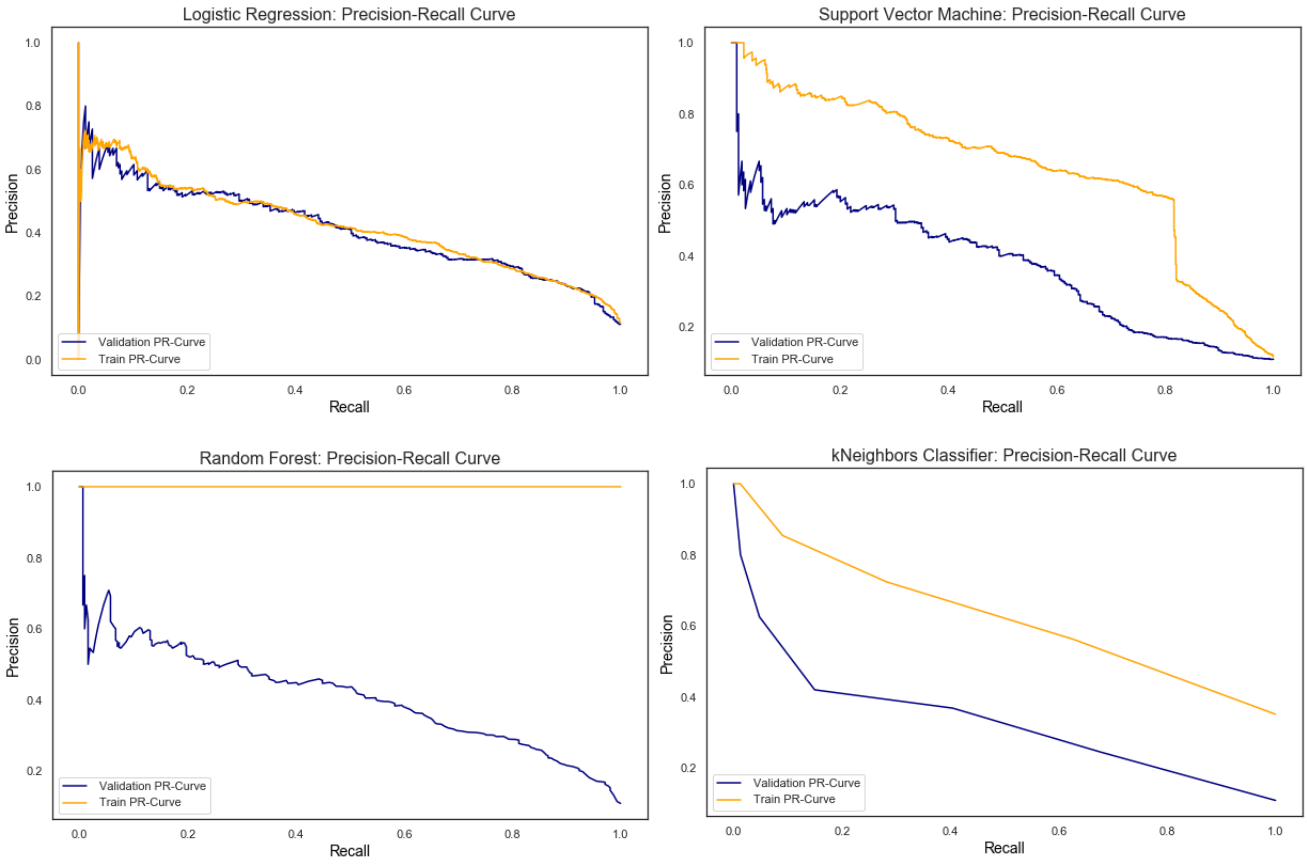

Figure 11: Baseline precision-recall curves

As can be observed from **Figure 10** and **11**, we can conclude that all the algorithms except Logistic Regression are overfitting the training data and showing poor performance with the test data. In fact, Random Forest showed a 100% recall and precision on training set but a mere 14% recall on the validation set. The logistic regression stands out – it shows a fairly low recall (about 15%), however, the precision is decent (about 55%) for both the training and test sets. The model seems to be good at correctly predicting cases with no CVD risk but is excessively likely to predict cases as negative. One positive thing about this model is that the training dataset results closely match the validation set results, implying that model is neither underfitting nor overfitting, also can be concluded from the monotonically reducing PR AUC curves.

To address the aforementioned issues and improve model performance, various resampling techniques were studied as mentioned in the next section.

Resampling Strategies

Since the positive CVD risk cases comprise 11% of the total cases, resampling techniques were employed to handle this highly unbalanced dataset. An important point to note here is that the resampling was performed only on the training to ensure an artificial structure wasn't created in the testing set.

Random Undersampling

Random undersampling involves randomly selecting examples from the majority class and deleting them from the training dataset. In the random under-sampling, the majority class instances are discarded at random until a more balanced distribution is reached.

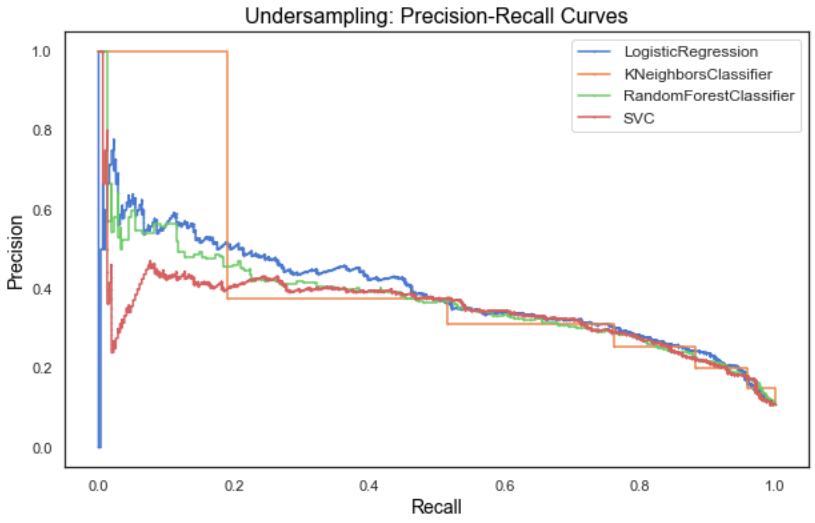

Figure 12: Precision-recall curves for models after undersampling

Table 4: Model metrics after undersampling

	MLA Name	Accuracy	Precision	Recall	F-1 score	ROC AUC	PRC AUC
0	Logistic Regression	0.7543	0.2794	0.8025	0.4145	0.7755	0.3890
1	KNeighbors Classifier	0.7319	0.2540	0.7611	0.3809	0.7447	0.3591
2	Random Forest Classifier	0.7433	0.2698	0.8025	0.4038	0.7693	0.3710
3	SVC	0.7336	0.2644	0.8185	0.3997	0.7709	0.3474

As can be observed from **Figure 12** and **Table 4**, there is a precision-recall tradeoff for the models. Undersampling has resulted in increased recall, however the precision has gone down substantially, which is common in imbalanced classes. All the models show similar recall and precision numbers, and will be considered for hyperparameter tuning.

Random Oversampling

Random oversampling involves randomly selecting examples from the minority class, with replacement, and adding them to the training dataset.

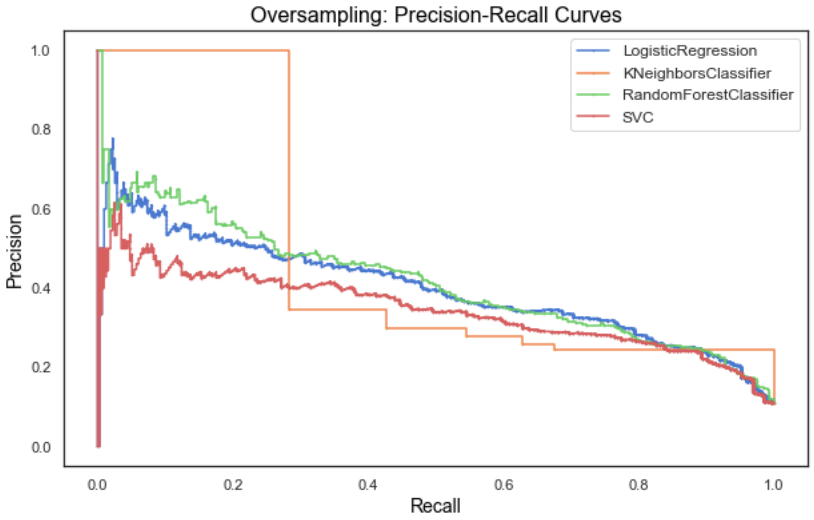

Figure 13: Precision-recall curves for models after oversampling

Table 5: Model metrics after oversampling

	MLA Name	Accuracy	Precision	Recall	F-1 score	ROC AUC	PRC AUC
0	Logistic Regression	0.7581	0.2823	0.7994	0.4173	0.7762	0.3984
1	KNeighbors Classifier	0.7968	0.2771	0.5446	0.3673	0.6860	0.3624
2	Random Forest Classifier	0.8937	0.5192	0.2580	0.3447	0.6145	0.4148
3	SVC	0.7729	0.2844	0.7229	0.4083	0.7510	0.3440

Similar, to undersampling, an increase in recall and a decrease in precision is observed for oversampling. As can be observed from **Figure 13** and **Table 5**, kNeighbors Classifier and Random

Forest Classifier show the worst precision and recall in this case. Logistic Regression with oversampling will be considered for hyperparameter tuning.

SMOTE

SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.

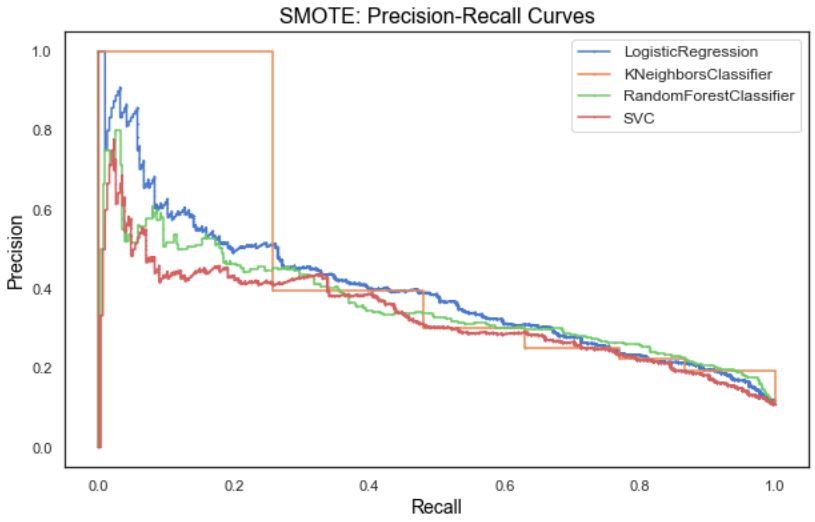

Figure 14: Precision-recall curves for models after SMOTE sampling

Table 6: Model metrics after SMOTE sampling

	MLA Name	Accuracy	Precision	Recall	F-1 score	ROC AUC	PRC AUC
0	Logistic Regression	0.8037	0.3071	0.6465	0.4164	0.7346	0.3932
1	KNeighbors Classifier	0.7547	0.2497	0.6306	0.3577	0.7002	0.3726
2	Random Forest Classifier	0.8464	0.3342	0.4204	0.3724	0.6593	0.3587
3	SVC	0.8081	0.2907	0.5350	0.3767	0.6882	0.3306

Based on **Figure 14** and **Table 6**, a very similar pattern of precision-recall tradeoff is observed for SMOTE, however, the absolute precision and recall numbers do not look as good as prior sampling techniques used, so SMOTE sampling won't be considered further.

SMOTEENN

SMOTEENN involves combination over- and under-sampling using SMOTE and Edited Nearest Neighbours.

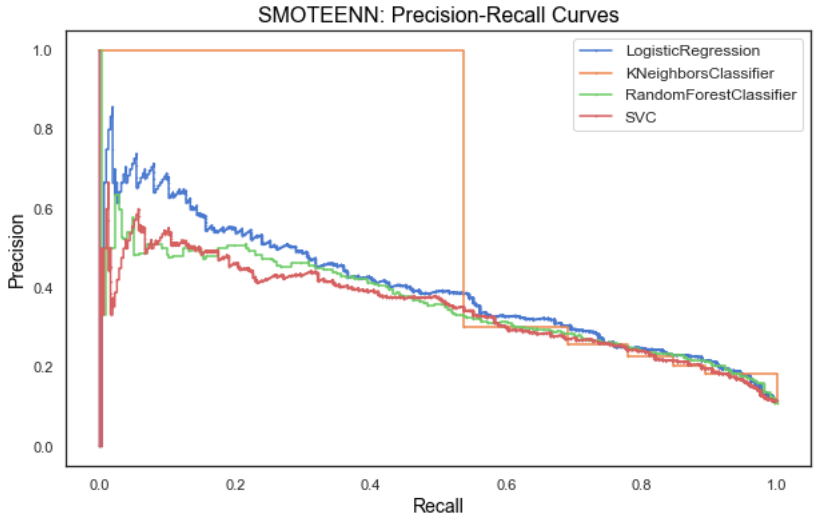

Figure 15: Precision-recall curves for models after SMOTEENN sampling

Table 7: Model metrics after SMOTEENN sampling

	MLA Name	Accuracy	Precision	Recall	F-1 score	ROC AUC	PRC AUC
0	Logistic Regression	0.7215	0.2492	0.7803	0.3778	0.7473	0.3995
1	KNeighbors Classifier	0.6888	0.2273	0.7803	0.3520	0.7289	0.4533
2	Random Forest Classifier	0.7654	0.2746	0.7102	0.3961	0.7411	0.3605
3	SVC	0.7422	0.2602	0.7484	0.3862	0.7449	0.3480

It can be noticed from **Figure 15** and **Table 7**, a similar pattern of precision-recall tradeoff is seen and Logistic Regression and kNeighbors will be considered for hyperparameter tuning.

Hyperparameter Tuning

Upon identifying the optimum models and the respective resampling techniques, hyperparameter tuning was performed via grid search 5-fold cross-validation. The tested hyperparameters for different models are as follows:

1. Logistic Regression - C: 0.001, 0.01, 0.1, 1, 10, 100, solver: newton-cg, lbfgs, liblinear, penalty: l2
2. Support Vector Classifier - C: 1, 0.1, 0.01, kernel: linear
3. Random Forest - n_estimators: 200, 400, 600
4. kNeighborsClassifier - n_neighbors: 5, 6, 7, 8, leaf_size: 1, 2

The performance of the tested models with hyperparameter tuning is shown in **Table 8**. All the models demonstrated an improvement in performance after hyperparameter tuning compared to pre-hyperparameter tuning.

Table 8: Model metrics after hyperparameter tuning

	Model	Sampling	Precision	Recall	PRC_AUC	ROC_AUC
2	Logistic Regression	SMOTEENN	0.2147	0.9045	0.3750	0.8366
1	Logistic Regression	Undersampled	0.2154	0.8535	0.3346	0.8175
6	KNeighbors Classifier	SMOTEENN	0.2254	0.8248	0.4595	0.7980
0	Logistic Regression	Oversampled	0.2443	0.8121	0.3599	0.8273
3	Support Vector Classifier	Undersampled	0.2656	0.8121	0.3858	0.8385
4	Random Forest	Undersampled	0.2748	0.8121	0.4148	0.8474
5	KNeighbors Classifier	Undersampled	0.2830	0.7994	0.3870	0.8299

As can be observed, Logistic Regression used in combination with SMOTEENN sampling returns the highest recall but a relatively low Precision. kNeighbors Classifier, on the other hand offers a high precision but a low Recall. There are various tradeoffs with different model architectures, and therefore the choice of the model will depend on the desired performance level across different metrics. For our current problem statement, our focus was to increase the number of individuals with CVD risk, however an increase in recall comes at the cost of reducing precision.

Figure 15 shows the Precision Recall curve and the ROC curve for the validation set with the most optimum model – Logistic Regression with SMOTEENN sampling.

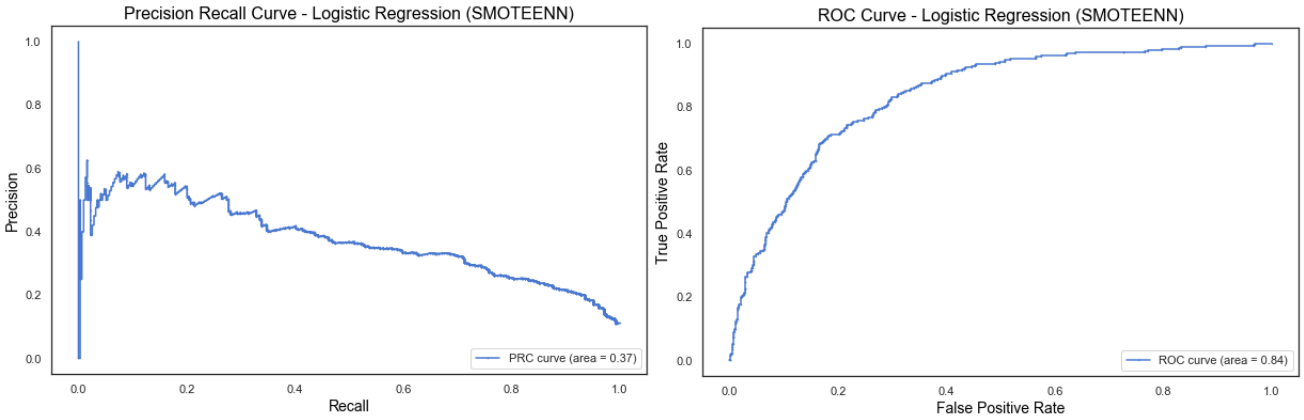

Figure 15: Precision-recall and ROC curves for Logistic Regression after SMOTEENN sampling for the validation set

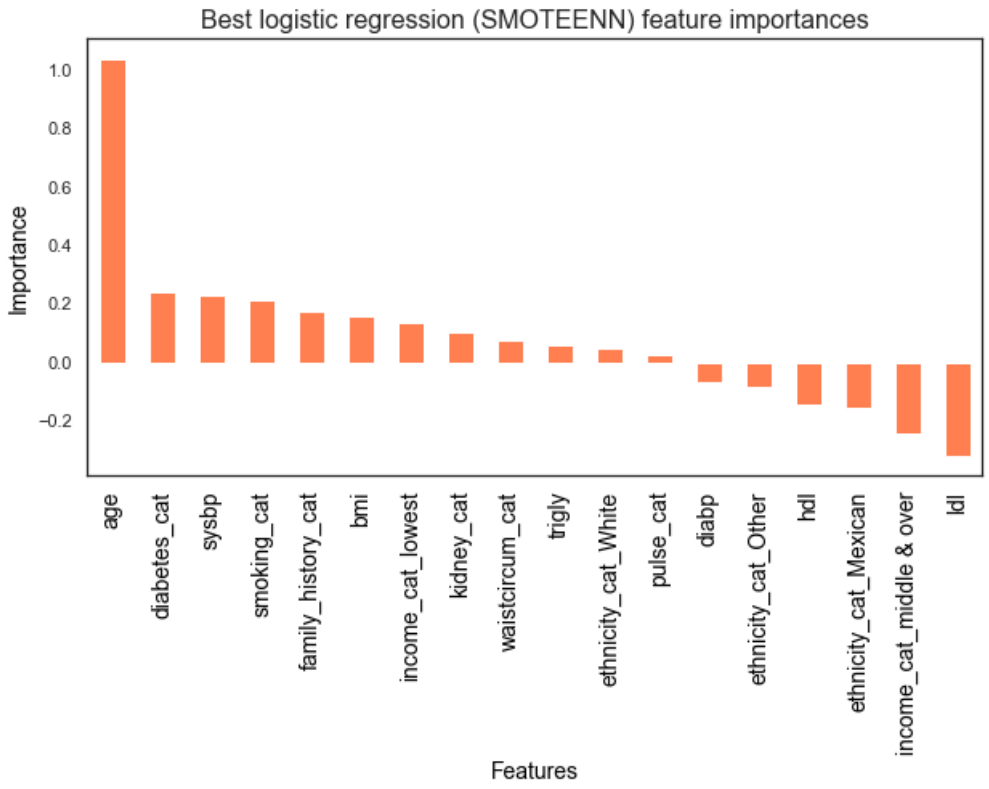

Figure 16: Best Logistic Regression features after SMOTEENN sampling for the validation set

Figure 16 identifies the best features identified by the model with the top 5 features being age, diabetes, systolic blood pressure, smoking and family history, which intuitively make sense.

The final model when tested on the holdout/test set showed area under PRC as 0.36 and area under ROC curve as 0.82 (**Figure 17**), which was similar to the results for validation test.

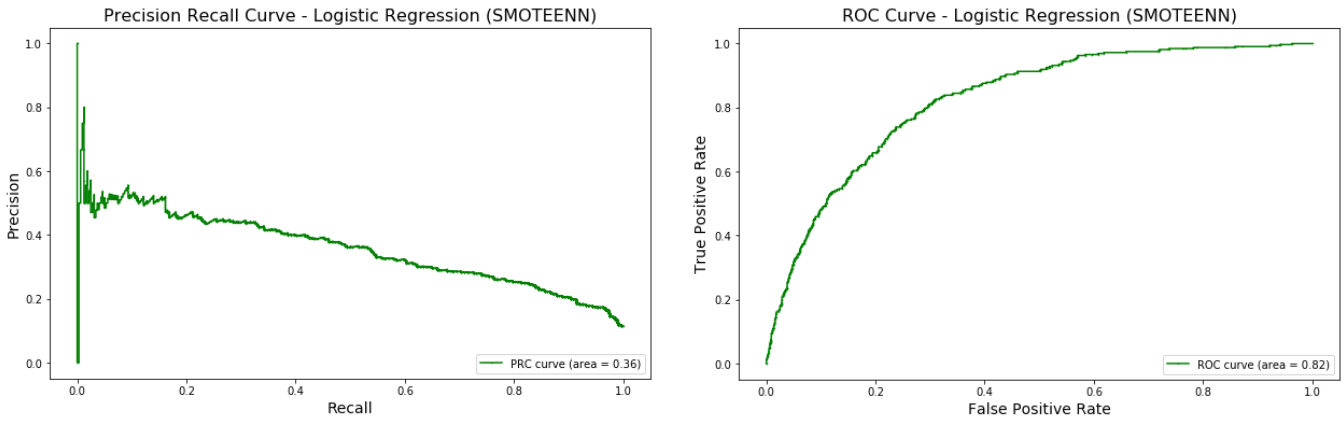

Figure 17: Precision-recall and ROC curves for Logistic Regression after SMOTEENN sampling for the test set

To ascertain that the estimated class probabilities were reflective of the true underlying sample probabilities, a reliability diagram or calibration curve was plotted for the final model as shown in **Figure 18**. As can be observed, our chosen model follows the perfectly until at high true probabilities where our model tends to over-forecast the results.

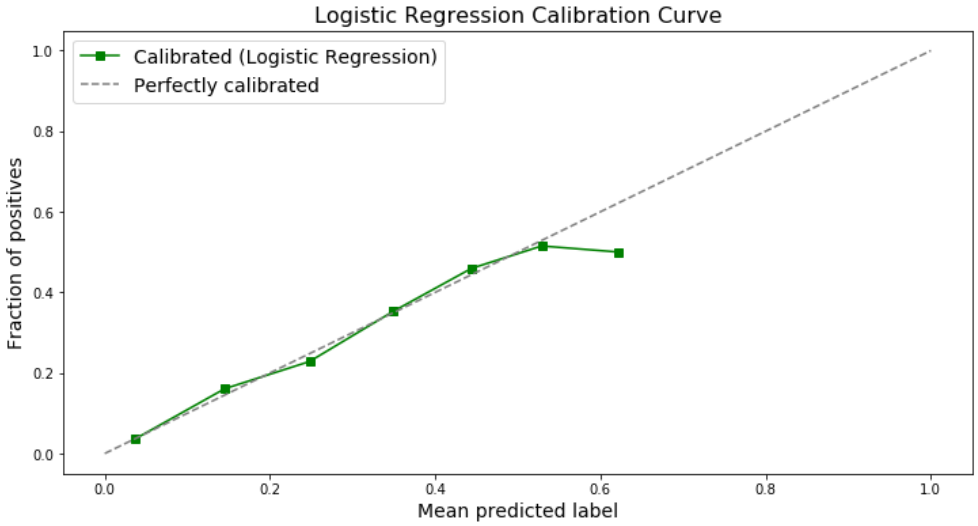

Figure 18: Calibration curve for the final model

Based on this model, a tool was built that would take the user's inputs on age, diabetes, hypertension, smoking and family history and would give the cardiovascular disease risk for an individual as shown in **Figure 19**. The tool can be accessed from the notebook https://github.com/richasethi3/CVD_Prediction/blob/master/04_Modeling/CVD_Prediction_Modeling.ipynb.

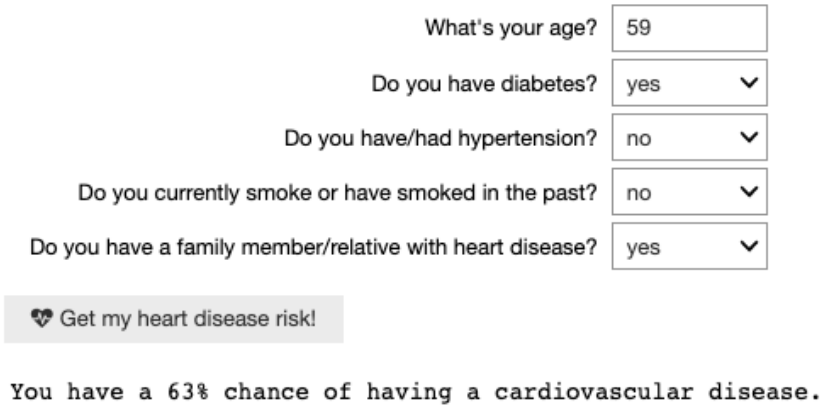

Figure 19: Snapshot of the cardiovascular disease risk assessment tool

This approach can help transform the health insurance industry in two ways: they can provide new sources of insight to simplify and streamline current underwriting, and they can enhance the understanding of risk to enable more refined, granular categorizations of risk.

This kind of algorithmic underwriting can increasingly become a prerequisite maintaining current position in the market. Moreover, with COVID-19 making today's in-person purchasing experience more difficult, many companies will increasingly recognize that automatic underwriting transformation is all the more urgent. Streamlined underwriting can set the stage for future innovation in the industry. Insurance product sales will shift from low-engagement, one-time transactions to an ongoing relationship between the customer and the insurance agency; this engagement will be defined by continuous underwriting and a greater focus on health and wellness. Increasingly, market segmentation will reach the level of individuals, with a richer understanding of each person in the risk pool.

Conclusions

Listed below are the conclusions from this study:

a) The baseline models showed an inequality in model performance between the negative and the positive class. Most models seemed especially weak with respect to recalls on CVD risk.

b) The target variable class CVD risk was highly imbalanced with 1:8 ratio between positive and negative class. To combat the problem of model bias toward the majority class, several resampling techniques were used either to increase the frequency of the minority class or decrease the frequency of the majority class.

c) Post resampling, the models with the best recall numbers and the corresponding resampling technique were chosen to perform hyperparameter tuning, which showed Logistic Regression with SMOTEENN sampling as the most optimum model with 90% recall and 20% precision.

d) The top 5 features for the most optimum model were age, diabetes, systolic blood pressure, smoking and family history, which were employed in the development of a cardiovascular disease prediction tool, which takes user input and predicts the disease risk based.

Future Work

This analysis and model are by no means an exhaustive study for this problem. This work was performed based on my personal interest on this topic, however the use of domain knowledge, experience and an alternative approach could result in a model that is a different from the current one. In the presence of current computational resources, time and knowledge, this was the best model that could be developed.

Few approaches that could result in better and more robust model architecture are:

a) Use of different learning algorithms including ensemble approach could result in a model with greater depth and breadth with increased predictive power.

b) Use of other resampling techniques and with different majority to minority ratios.

c) Include more features as input to the model could increase model performance.

d) Use of other advanced feature selection algorithms.