Binary Prediction of Smoker Status using Bio-Signals and Machine Learning

Utilizing Machine Learning to Predict Smoker Status from Bio-Signals

Group S-1

Abstract and Introduction

Problem

“Smoking increases risk of developing more than 50 serious health condition” - NHS

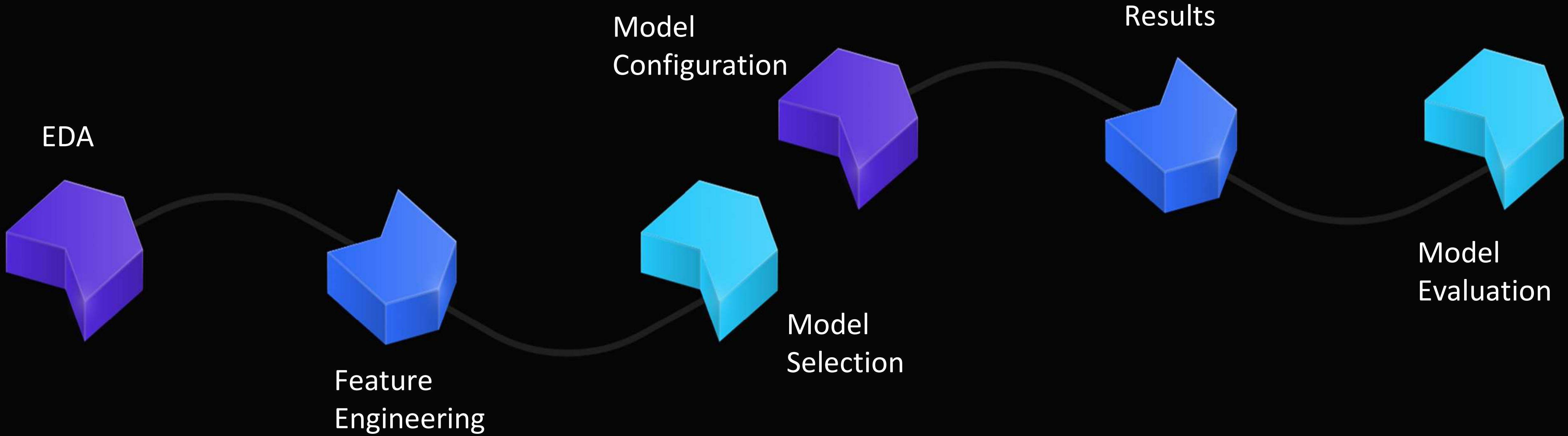
Objective

Explores the use of machine learning to predict smoking status from a variety of medical features

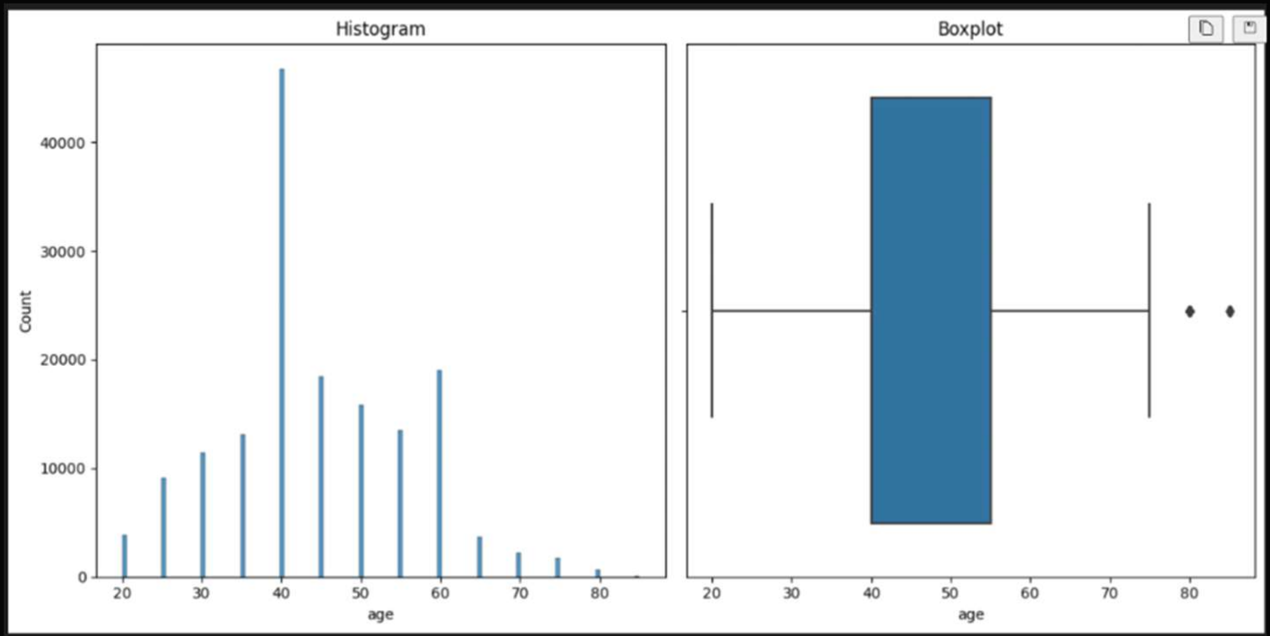
Scope

Our journey includes data preprocessing, model selection, evaluation, and deployment

Steps towards predicting smoking status



Exploratory Data Analysis



1. Quality and Structure

- ➔ Shape
- ➔ Unique values per feature
- ➔ Computed statistical measures
- ➔ Graphical representations

Exploratory Data Analysis

StandardScaler

Kolmogorov-Smirnov test

Select populations

t-test or Mann-Whitney U

p_value (significance)

Conclusion

Medical Description

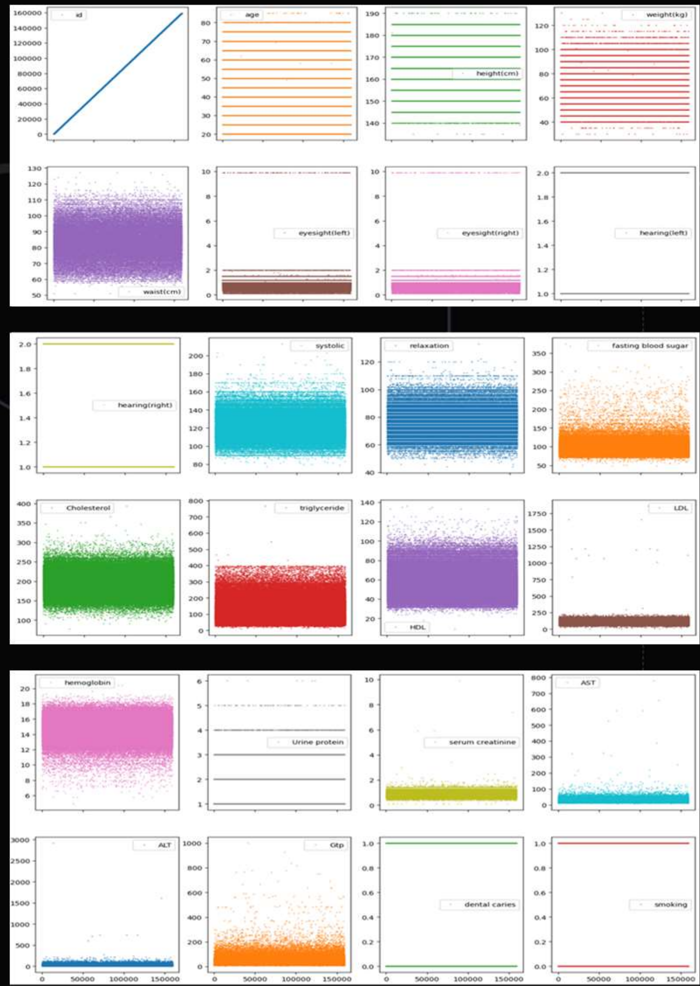
Reasonable Values

Data Type

Values Collected

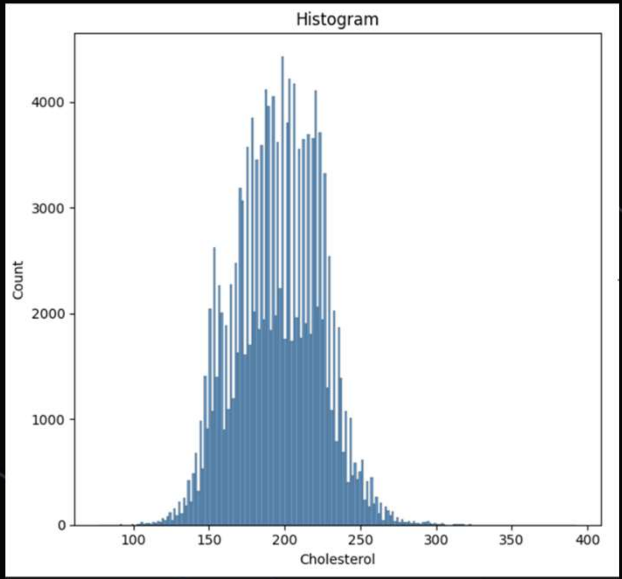
Relationship with target

Observations for f.e.

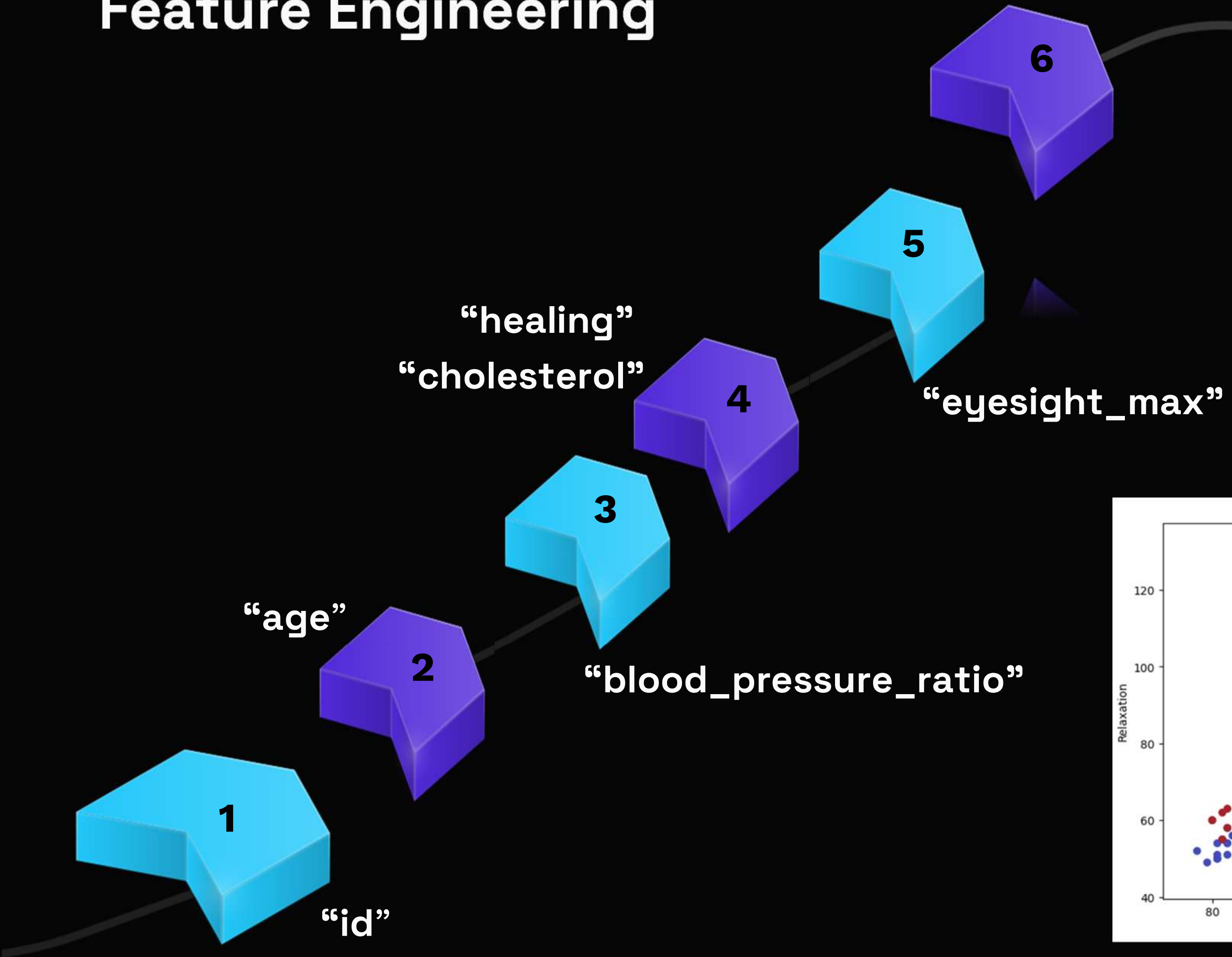


Exploratory Data Analysis

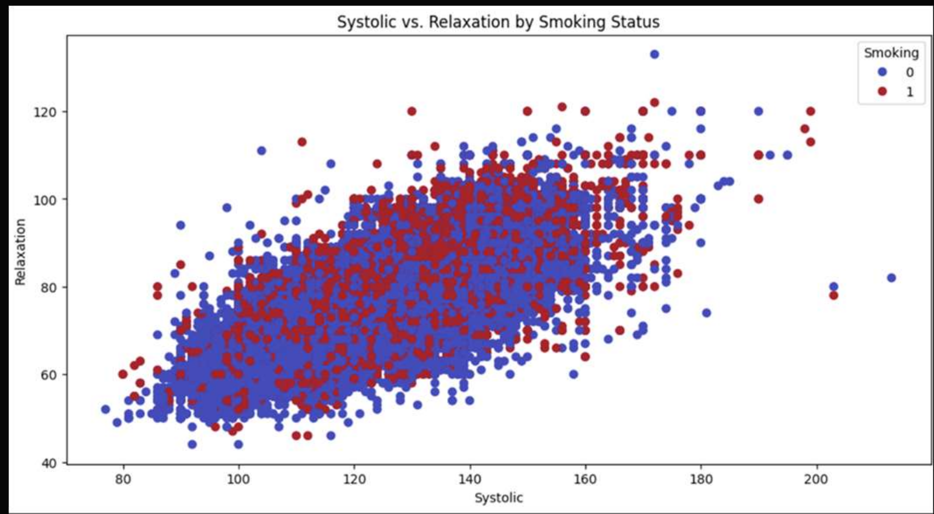
Feature	Insights
Age	Majority class = aged 40
Eyesight	Visual acuity of smokers < non-smokers. // Extreme outliers
Waist / Height	Smokers > non-smokers in the sample, contrary to studies.
Hearing	Few cases // Worse in non-smokers
Systolic / Relaxation	Too high values, overly specific cases
Fasting blood sugar	Extreme boundary values
Cholesterol / Triglyceride / HDL / LDL	Extreme values // Cholesterol = the sum of other variables
Hemoglobin	In smokers > non-smokers
Urine protein / Serum creatinine	Values as limit range data points (diseases)
AST / ALT / Gtp	Values exceed the boundaries (diseases)
Dental caries	Smokers > non-smokers



Feature Engineering



Feature	Max value
Blood_pressure_ratio	150
Fasting blood sugar	200
HDL	110
LDL	200
Serum creatinine	4
Urine protein	4
AST	100
ALT	100
Gtp	300



Model Configuration

Data Scaling

We rare using MinMaxScaler() due to its non-Gaussian distribution and the situation with the outliers

Sampling

Tried training the models using a sample of the dataset instead of the entire one. The resulting model did not make much sense.

Hyperparameter Tuning

Each model has its own set of characteristics, complexities and underlying assumptions.

- We started with GridSearch() and then shifted to RandomizedSearchCV()
- K-Fold cross validation with K=5
- ROC AUC as scoring metric for comparing the models

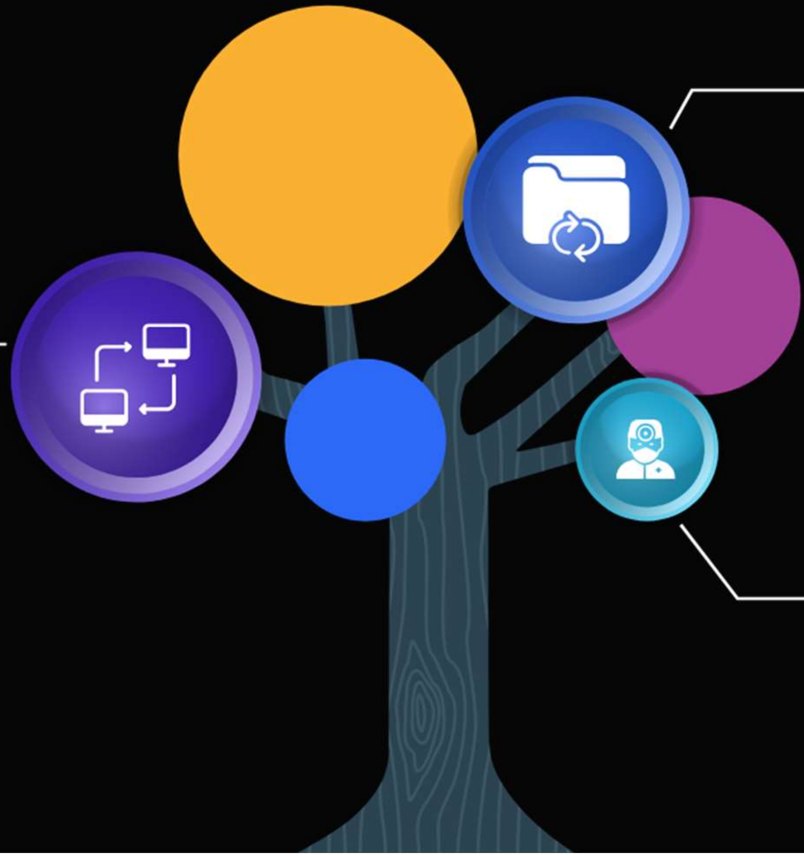
Model Selection

Non-probabilistic models

Decision Trees are straightforward and easy to understand but prone to overfitting and high variance.

Random Forest are ensemble methods that combine multiple Decision Trees to improve performance.

XGBoost is a gradient boosting algorithm that incorporate regularization techniques to prevent overfitting and also minimizes loss.



Logistic Regression

Understandable model where feature engineering is needed as it is prone to noise

SVMs

Robust against over-fitting but computationally intensive

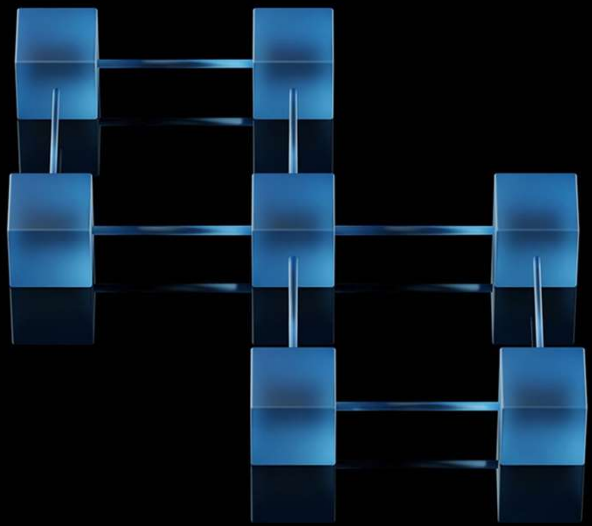
Coding the models

First Impressions

Logistic and XGBoost seem good (ROC AUC around 0.84)

Decision Trees and Random Forest give us a ROC AUC of 1

SVM is to heavy to compute, so taking into account it is not efficient with large datasets we discard it



Once training with hyper-parameter tuning

Once we run the code to perform the RandomizedSearch() we will perform the following analysis:

1. Original Dataframe
2. Transformed Dataframe
3. Transformed Dataframe with ages under-sampled

Evaluation of Models

	precision	recall	f-1 score
non-smoker	0.83	0.70	0.76
smoker	0.68	0.82	0.75
accuracy			0.75

Table 4: Scoring metrics for Logistic Regression model trained on the original dataframe

	precision	recall	f-1 score
non-smoker	0.82	0.71	0.76
smoker	0.68	0.82	0.74
accuracy			0.75

Table 5: Scoring metrics for Decision Tree model trained on the original dataframe

	precision	recall	f-1 score
non-smoker	0.91	0.95	0.93
smoker	0.93	0.88	0.91
accuracy			0.92

Table 6: Scoring metrics for Random Forest model trained on the original dataframe

	precision	recall	f-1 score
non-smoker	0.86	0.78	0.81
smoker	0.74	0.83	0.79
accuracy			0.80

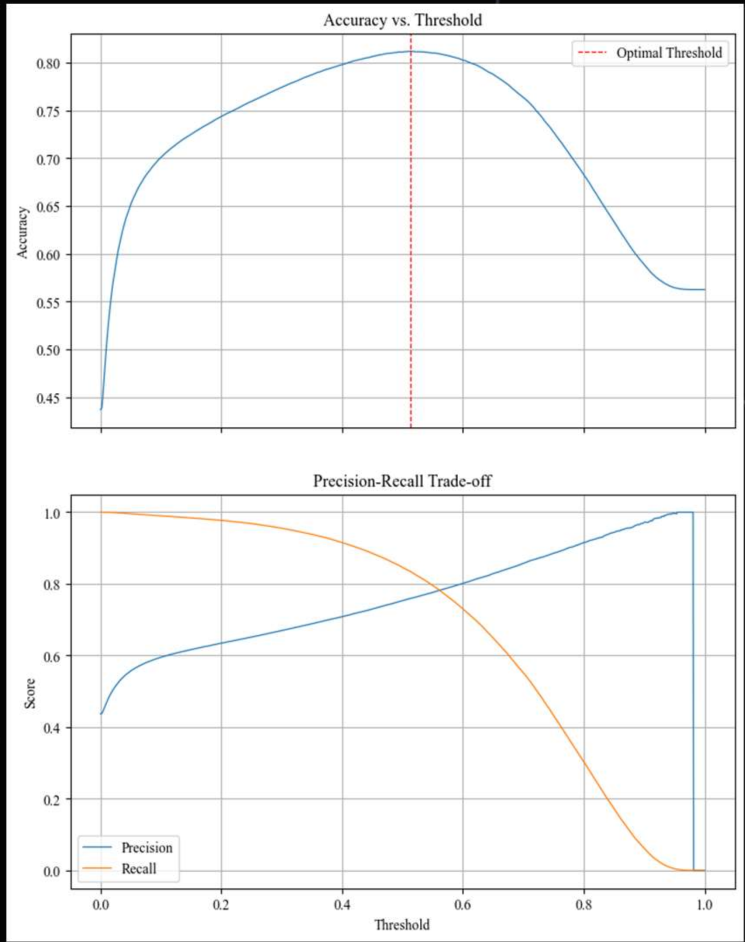
Table 7: Scoring metrics for XGBoost model trained on the original dataframe.

	precision	recall	f1-score
non-smoker	0.86	0.81	0.84
smoker	0.78	0.83	0.80
accuracy			0.82

Table 8: Scoring metrics for XGBoost model trained on the transformed dataframe.

	precision	recall	f1-score
non-smoker	0.85	0.80	0.82
smoker	0.76	0.81	0.78
accuracy			0.80

Table 9: Scoring metrics for XGBoost model trained on the transformed and age undersampled dataframe.

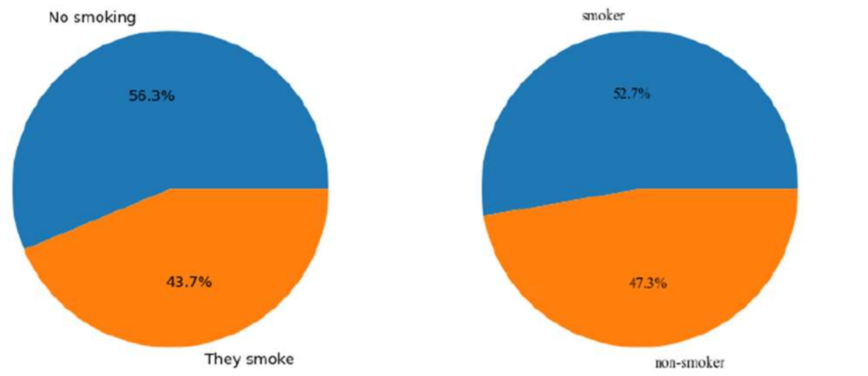


- Logistic Regression on the original dataset showed balanced performance across classes, but with room for improvement in accuracy.
- Decision Tree metrics indicated a similar trend, with a slight variation in precision and recall.
- Random Forest outperformed other models on the original data, showing high precision and recall for both classes.
- XGBoost on the original data had lower precision for non-smokers compared to the Random Forest model.
- Upon transformation, XGBoost's performance slightly decreased, which was unexpected given the feature engineering efforts.
- The age-undersampled dataset further decreased the XGBoost performance, hinting at the synthetic data's inability to capture the original data's essence.
- Scaling the original training dataset and applying XGBoost revealed a distinct peak in the Accuracy vs. Threshold graph, indicating an optimal point for classification threshold.
- The Precision-Recall Trade-off curve underscored the inverse relationship between precision and recall, emphasizing the need for a strategic balance in threshold setting to cater to specific application requirements.

Results and Kaggle Scores

Comparison of Predictive Model Scores

	Log. Regression	Decision Tree	Random Forest	XGBoost
Public Score	0.7673	0.75894	0.78071	0.78673
Private Score	0.76207	0.75714	0.77894	0.78687



(a) Smokers distribution in the original training dataset.

(b) Smokers distribution for the predictions in the test dataset.

- Scores ranged narrowly, with XGBoost achieving the highest accuracy on the original dataset.
- Decision Tree and Logistic Regression underperformed compared to Random Forest and XGBoost.
- Random Forest's running time was significantly longer, heavily dependent on CPU capabilities.
- Kaggle scores showed the original dataset without feature engineering yielded the best results.
- Final model predictions indicated a higher proportion of smokers compared to the original dataset's distribution.

Conclusion

Challenges

- Sensitive to over-fitting
- Effectiveness of Sampling
- Feature Engineering's Double-Edged Sword

Future Solutions

- Exploration of Alternative Models and Techniques: LightGBM, CatBoost
- Hyper-parameter tuning techniques such as Optuna or Ray Tune