Data Science
CSE558
Dr. Supratim Shit

KEY INDICATORS OF GENERAL HEALTH

AN OVERVIEW OF KEY HEALTH INDICATORS RESPONSIBLE FOR GENERAL HEALTH

INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI

Karan Gupta	2021258
Kuber Budhija	2021260
Shantanu Prakash	2021285
Shivesh Gulati	2021286
Rahul Oberoi	2021555

PROBLEM STATEMENT AND ITS IMPORTANCE

Problem Statement: To analyze and assess an individual's general health by examining a range of parameters, including BMI, ethnicity, sleep time, age, gender, and lifestyle factors. The aim is to identify patterns that correlate with various levels of health risk, allowing for a comprehensive understanding of how different factors impact an individual's overall health.

Importance

- This assessment can guide individuals and healthcare providers in making informed decisions to improve health and reduce potential risks.
- Enhance our understanding of how lifestyle choices and medical conditions contribute to an individual's health.
- Assist in developing targeted interventions by identifying high-risk groups.

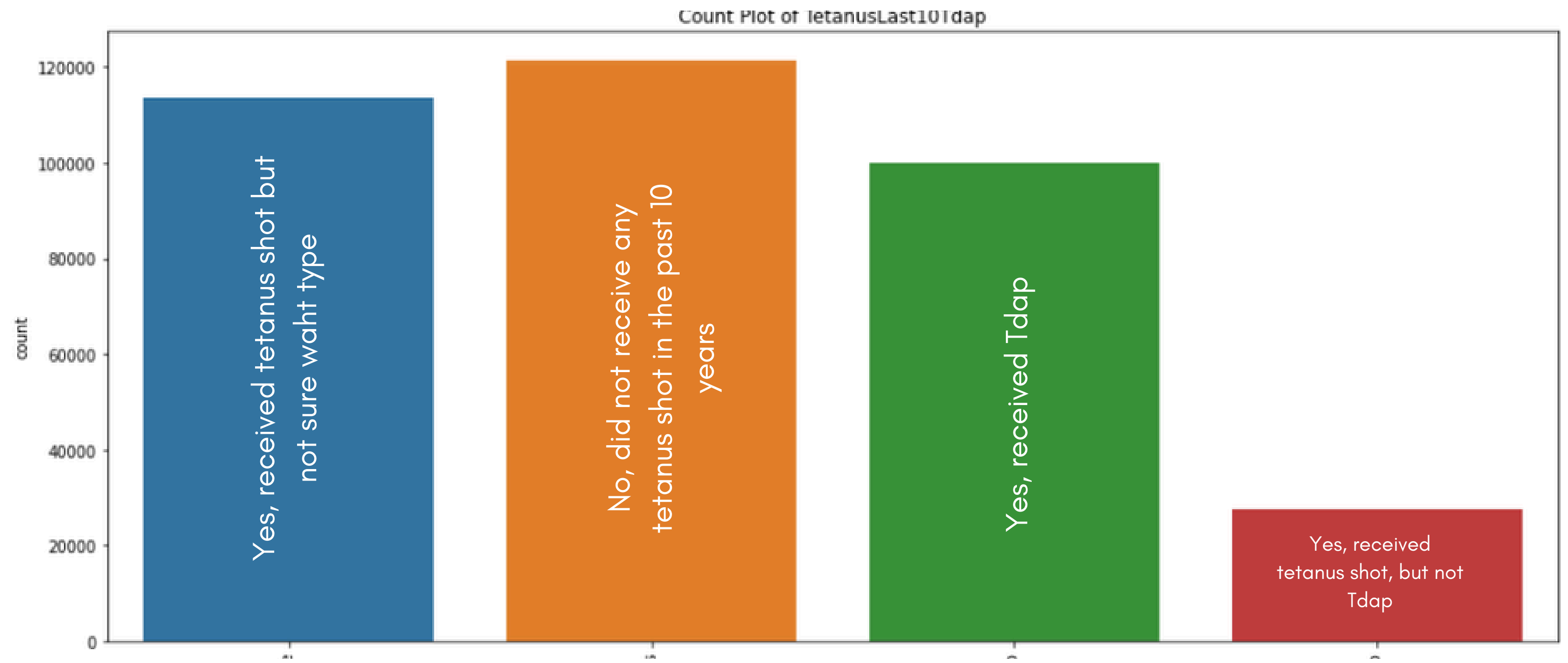
DATASET DESCRIPTION

Number of Entries	445132
Number of Duplicates	157
Number of features	39
Number of categorical features	34
Number of numerical features	5
Target Column	General Health
Total NaN values	902665
Total Data Points	17805280

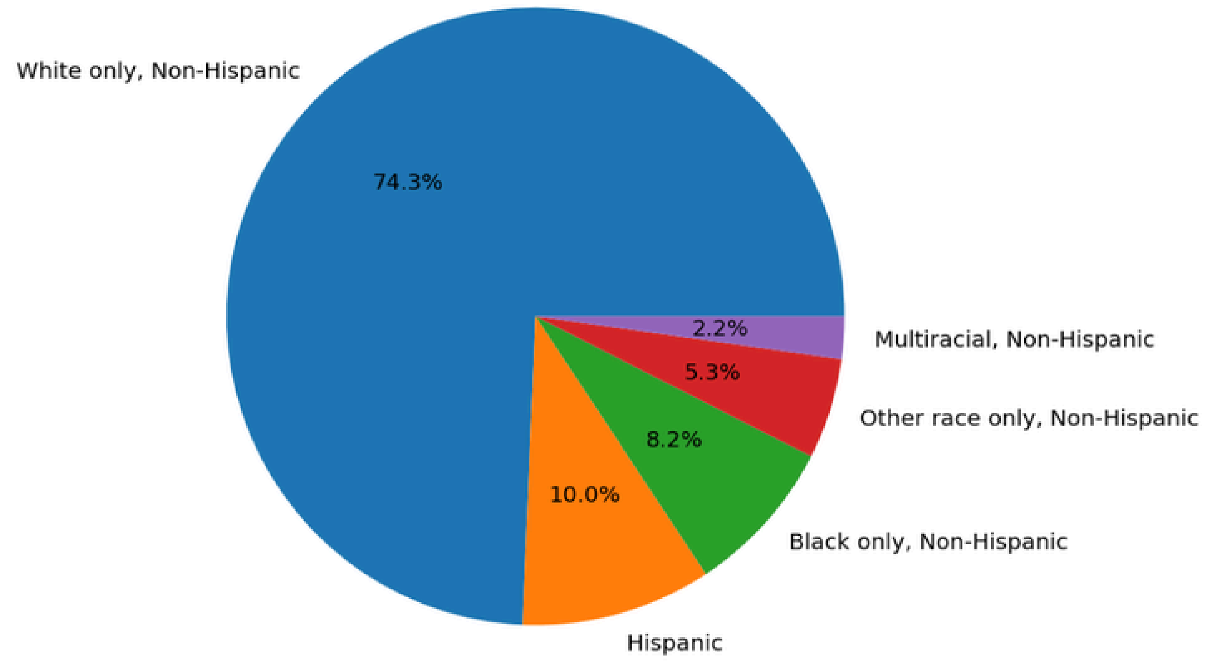
Categorical				**Numerical**
State	Had Asthma	Blind or Vision Difficulty	Race Ethnicity Category	Physical Health Days
Sex	Had Skin Cancer	Difficulty Concentrating	Age Category	Mental Health Days
Last Checkup Time	Had COPD	Difficulty Walking	Alcohol Drinkers	Sleep Hours
Physical Activities	Had Depressive Disorder	Difficulty Dressing Bathing	HIV Testing	Height In Meters
Removed Teeth	Had Kidney Disease	Difficulty Errands	FluVaxLast12	Weight In Kilograms
Had Heart Attack	Had Arthritis	Smoker Status	PneumoVaxEver	BMI
Had Angina	Had Diabetes	E Cigarette Usage	TetanusLast10Tdap	
Had Stroke	Deaf or Hard of Hearing	Chest Scan	HighRiskLastYear	
CovidPos				

Dataset link: https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease

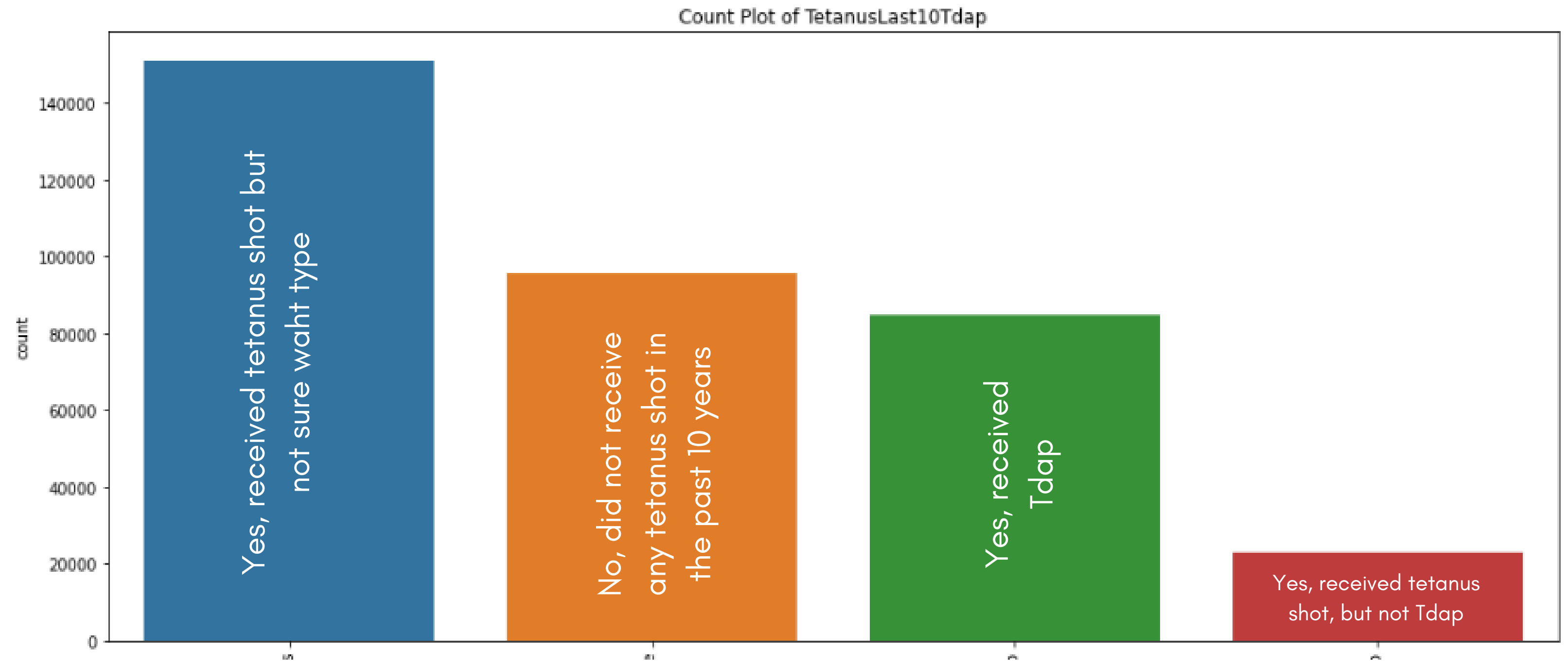
CATEGORICAL EDA: COUNT PLOT AND PIE CHART



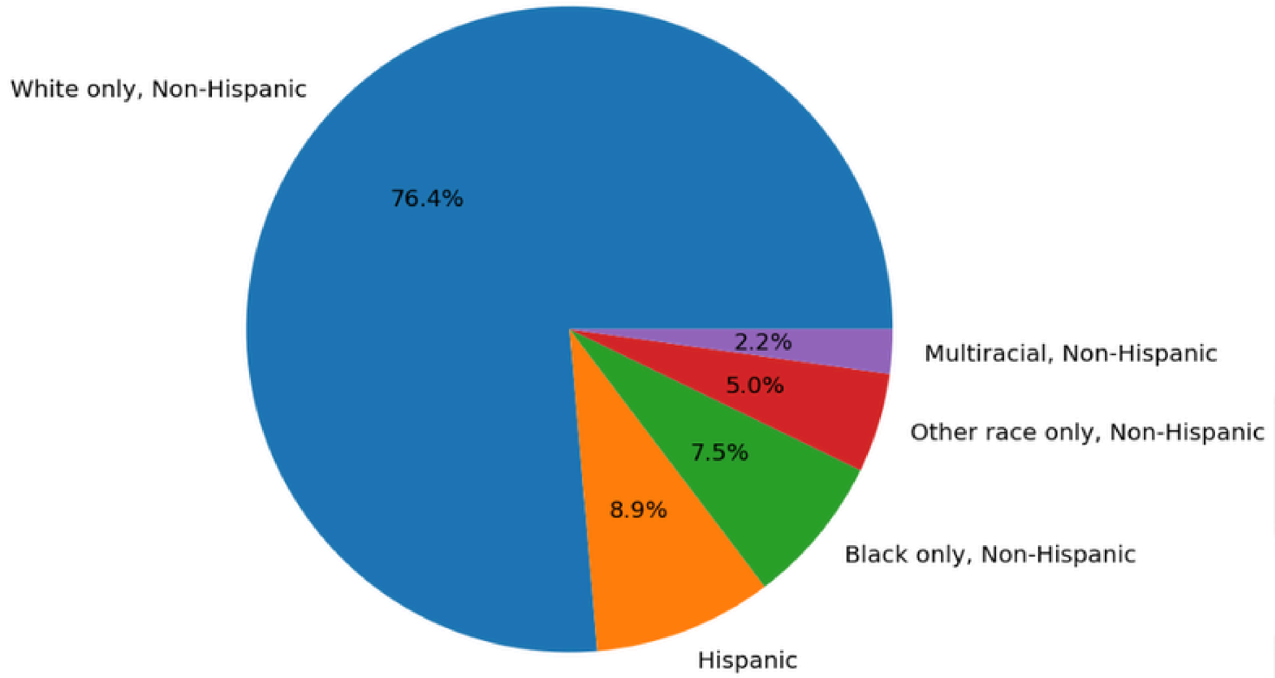
Before Preprocessing



Before Preprocessing



After Preprocessing



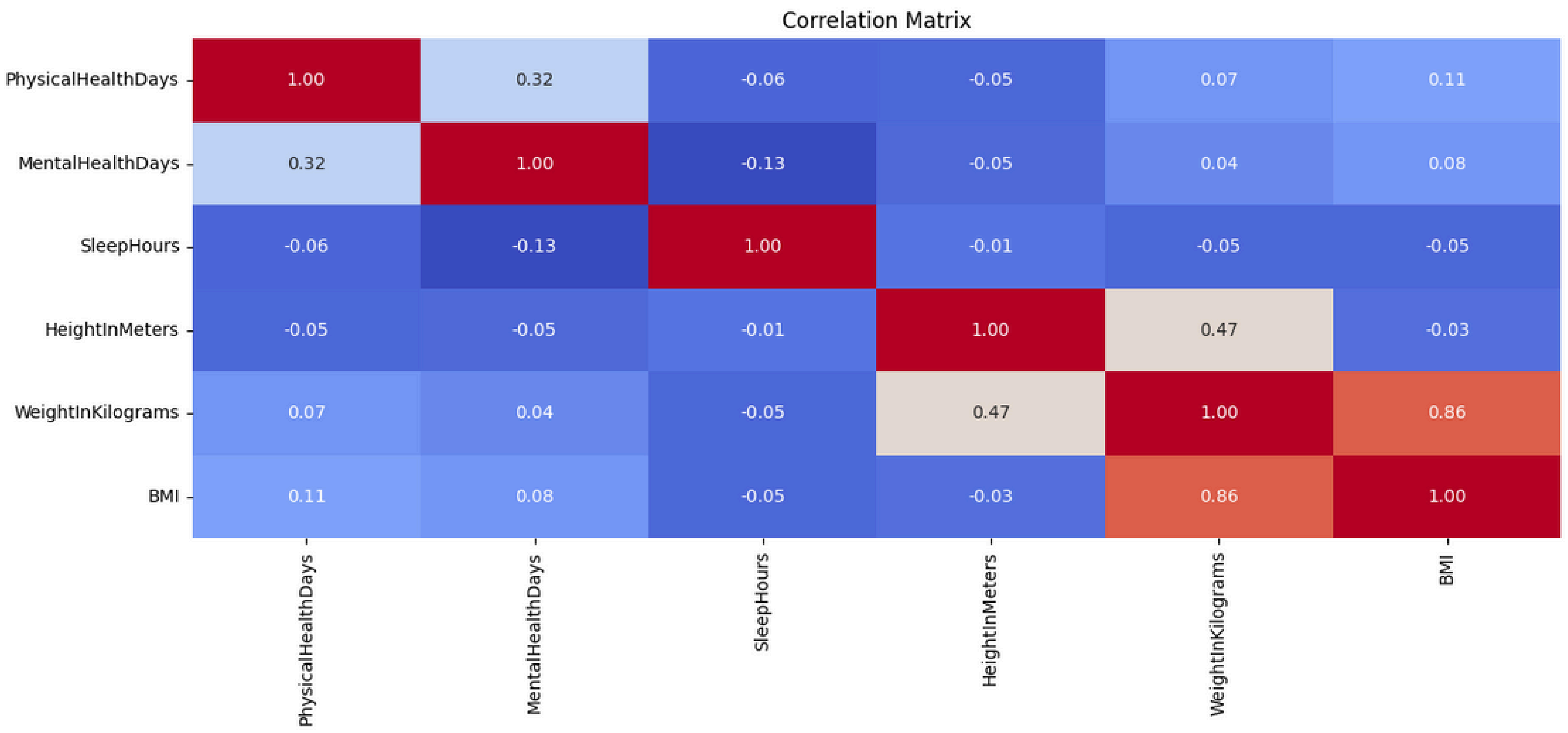
After Preprocessing

NUMERICAL EDA

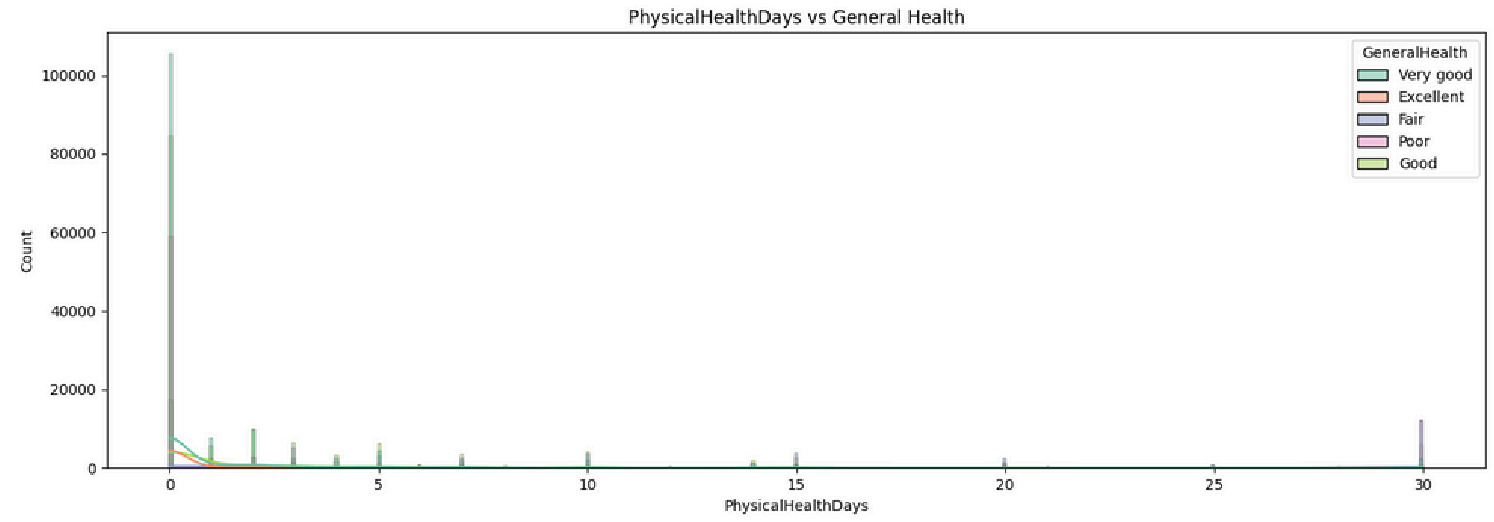
- In the **box plots**,
  - for the columns ‘SleepHours’ , ‘WeightInKilograms’ and ‘HeightInMeters’ vs ‘GeneralHealth’ we observed high amount of outliers which we removed during preprocessing.

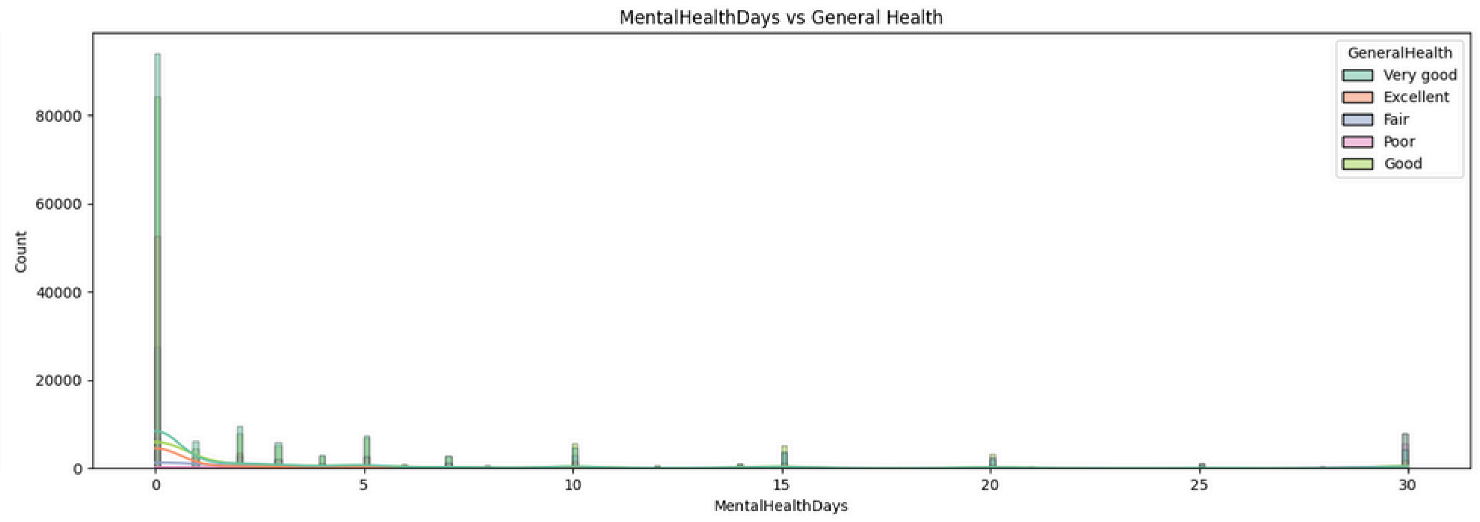
- In **histograms**,
  - *‘MentalHealthDays vs GeneralHealth’* and *‘PhysicalHealthDays vs GeneralHealth’* indicate distribution of the two feature columns is very dispersed. (Most of the values are zero).

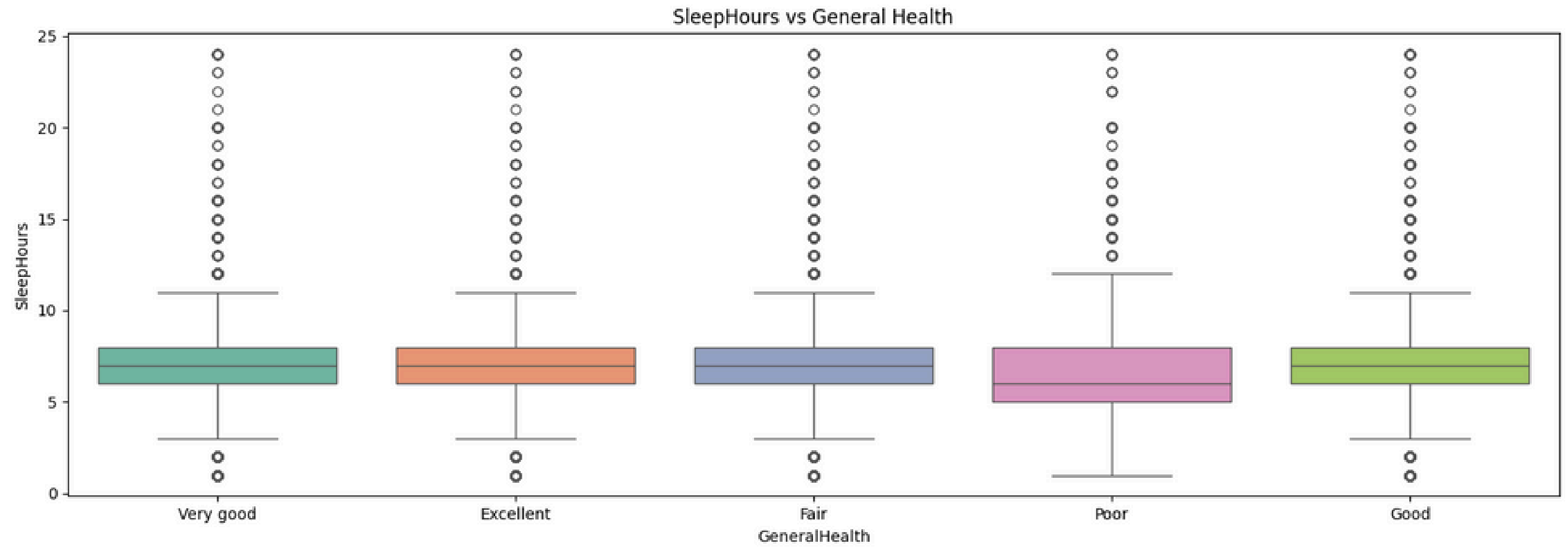
- In **correlation heatmap**, it is visible there is no strong correlation between the feature columns indicating majority of our columns are independent.



NUMERICAL EDA







Before Preprocessing

After Preprocessing

PREPROCESSING

1. Box Plot Outlier Detection : On the basis of Box plots, we detected values outside 1.5 IQR range in columns ‘SleepHours’ , ‘WeightInKilograms’ and ‘HeightInMeters’ and performed outlier removal after taking some marging from that.
2. Data Imputation (Replacing Nan values) [Which was necessary to be performed before LOF outlier detection]
   a. Categorical: Mode
   b. Numerical: Median
3. LOF Outlier Detection
   - Contamination rate= 0.1
   - Number of neighbours 20

For a given Data set

$$D_n = \{ (x_i, y_i) | x_i \in R^2, y_i \in \{X, Y, Z\} \}$$

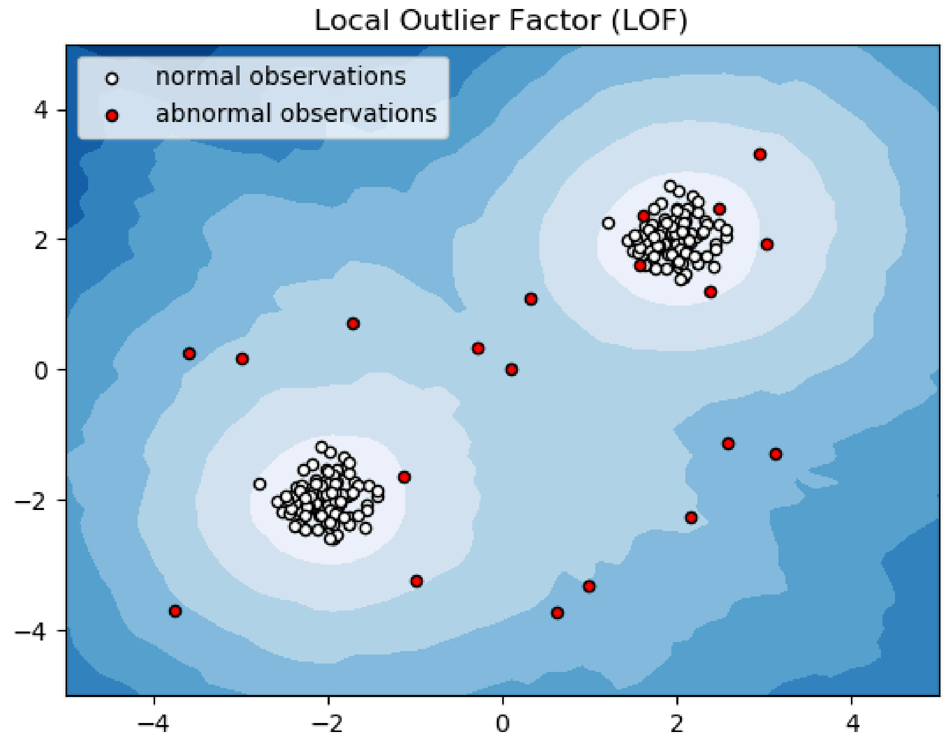
Local Outlier Factor for each data point is given by

$$\boxed{LOF(x_i) = \frac{\sum_{x_j \in N(x_i)} lrd(x_j)}{|N(x_i)|} \times \frac{1}{lrd(x_i)}}$$

$|N(x_i)|$: Number of elements in the neighborhood of x_i

$lrd(x_i)$: Local Reachability Density of x_i

www.mlp



Number of Entries	354862
Number of Duplicates	0
Number of features	39
Total NaN values	0
Total Data Points	14194480

HYPOTHESIS TESTING

HYPOTHESIS 1

Null Hypothesis (H0): *Age Category* does not affect General Health

Alternate Hypothesis (H1): *Age Category* affects general health

Note: We have tested multiple categorical columns, Age Category, here, is used as an example

To see results of all Tests click here

HYPOTHESIS 2

Null Hypothesis (H0): More than 40% of the population are overweight

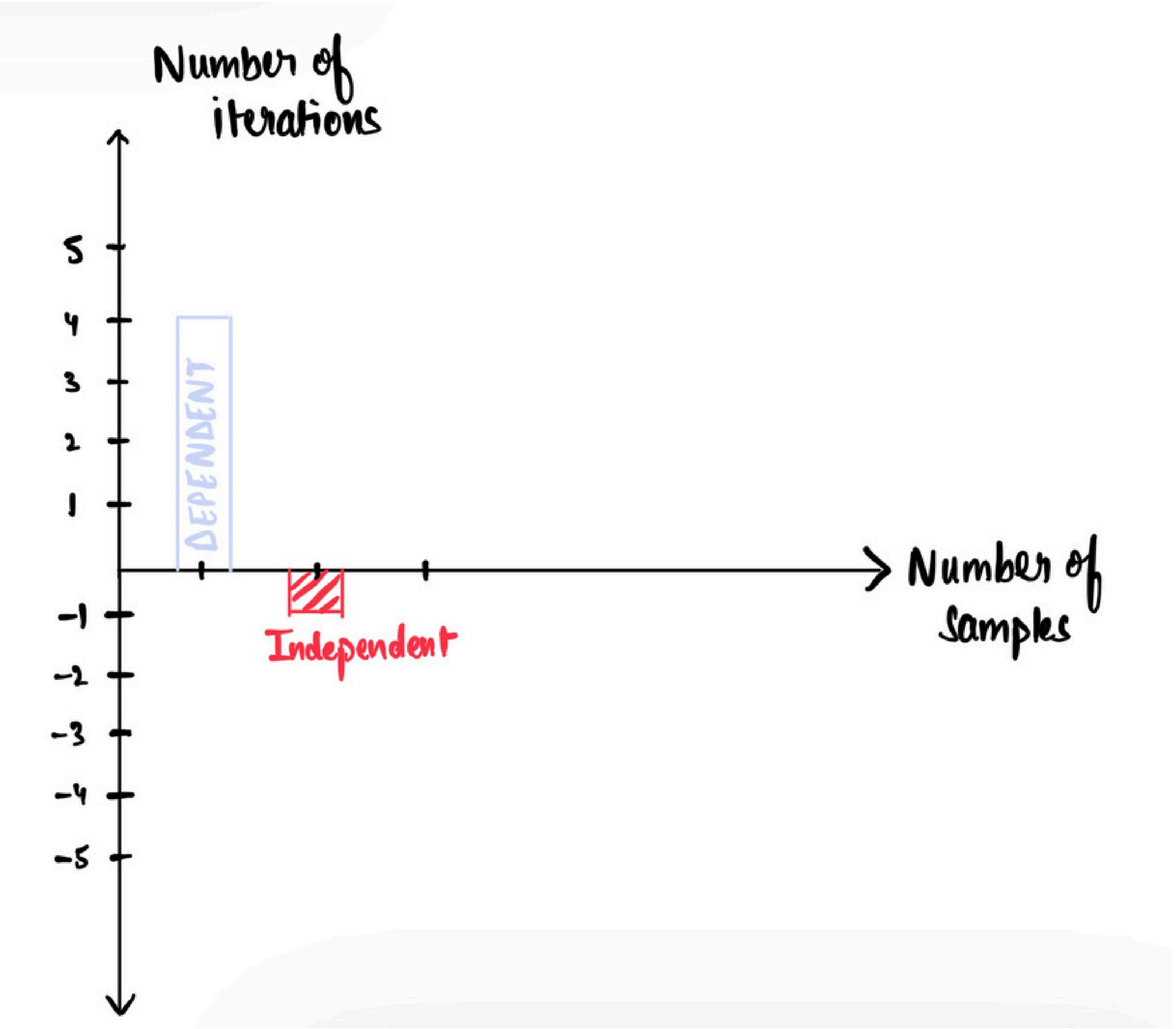
Alternate Hypothesis (H1): Atmost 40% people of the population are overweight

HYPOTHESIS 3

Null Hypothesis (H0): Mean BMI for all the classes in the “GeneralHealth” column is the same.

Alternate Hypothesis (H1): Mean BMI for all the classes in the “GeneralHealth” column is not the same.

χ^2 INDEPENCE TEST : EXPERIMENTS



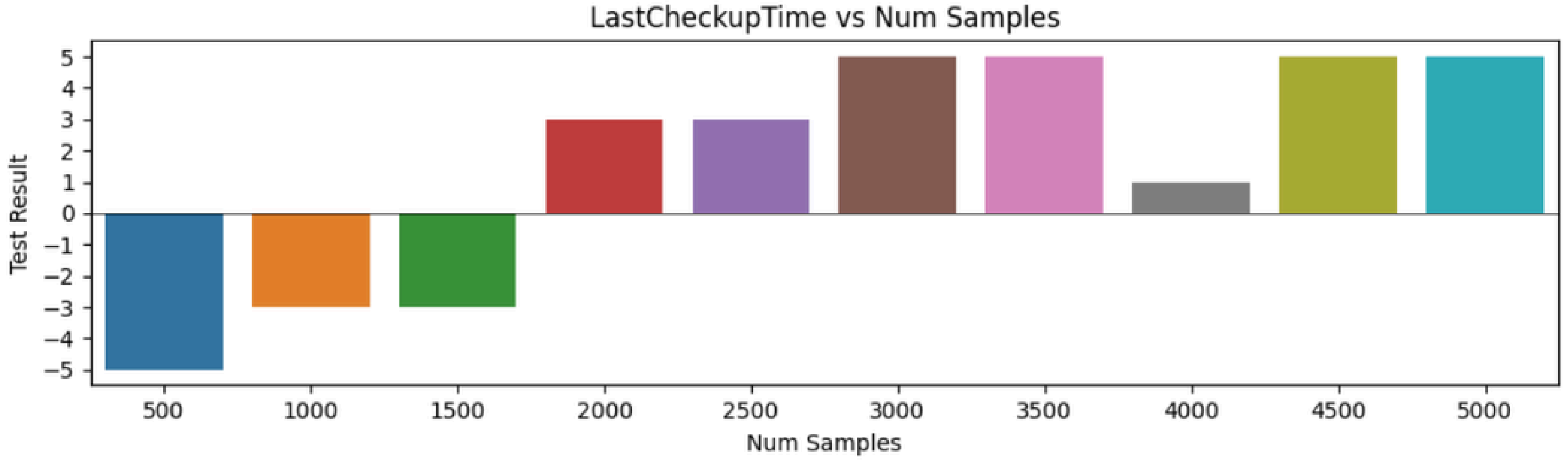
- We have used stratified sampling to create balanced subsamples where all of the target classes are equally represented. This will allows us to test for dependency without the skewed influence of the majority classes.

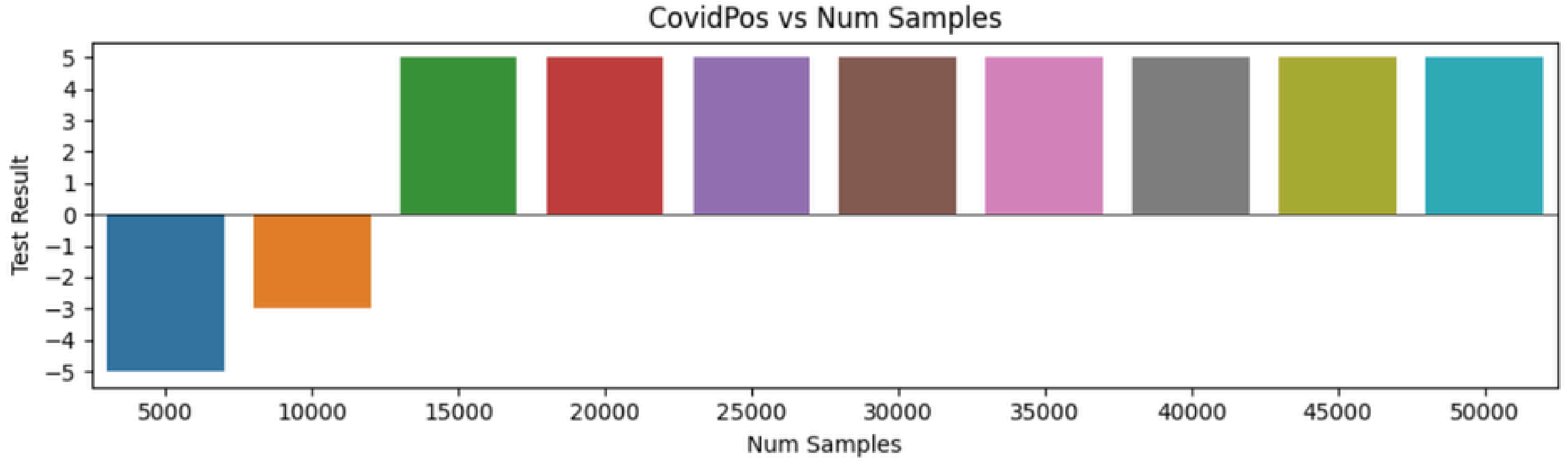
- We have taken multiple stratified samples of different sizes ranging from 100-1000 and 1000-10,000 for 5 iterations each, ensuring each sample has a similar proportion of all the classes.

- We then applied the chi-squared test for independence on each of these stratified samples, where +1 represented dependence (rejection of null hypothesis) and -1 represented independence (fail to reject null hypothesis). This allowed us to verify if the hypothesis test is being satisfied consistently across the samples on an average.

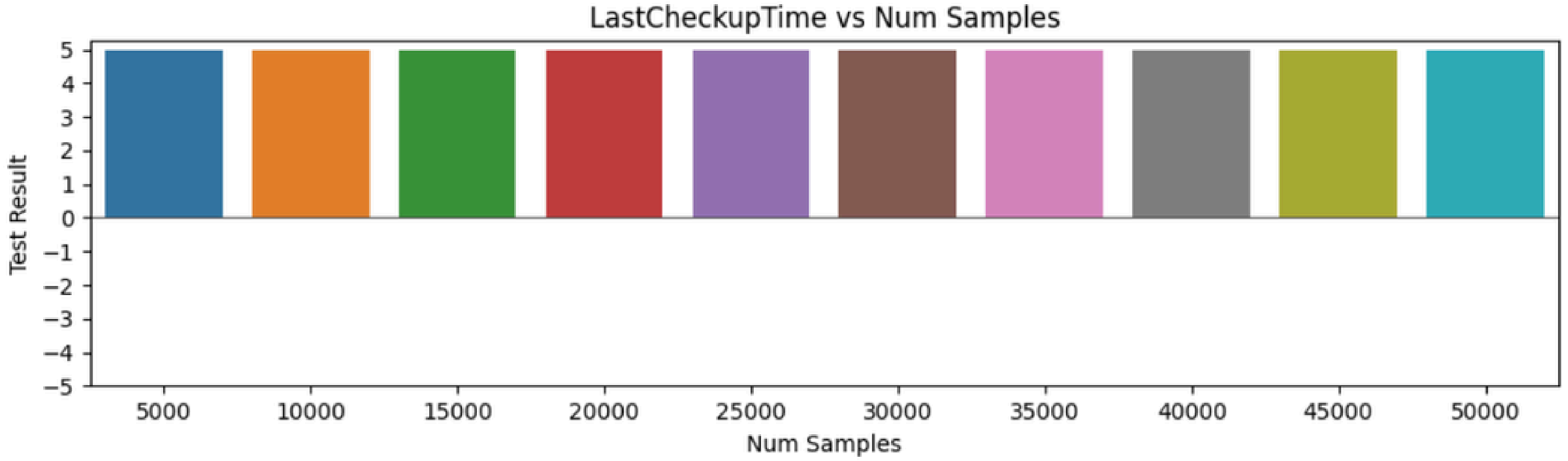
χ^2 RESULTS: RANDOM SAMPLING

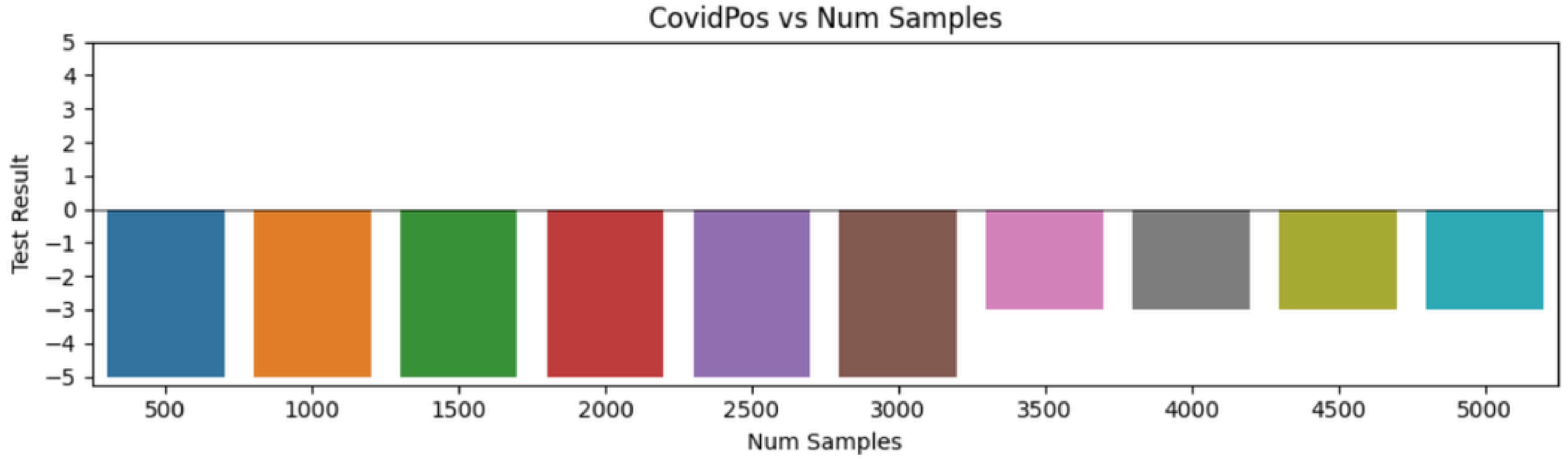
Samples: 100 - 1000





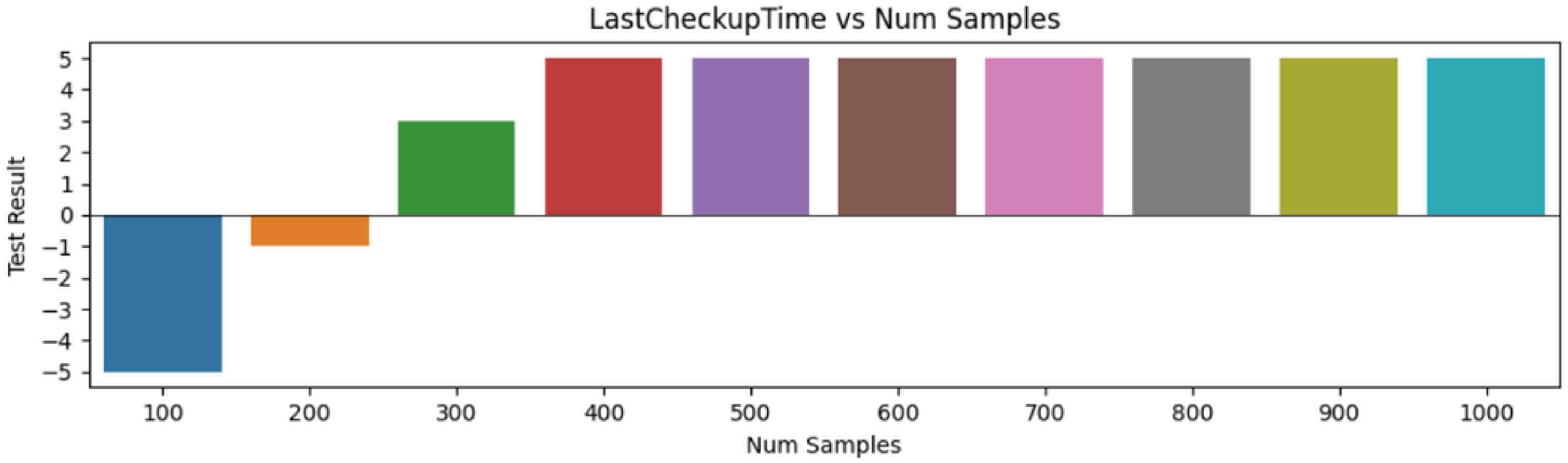
Samples: 1000 - 10000

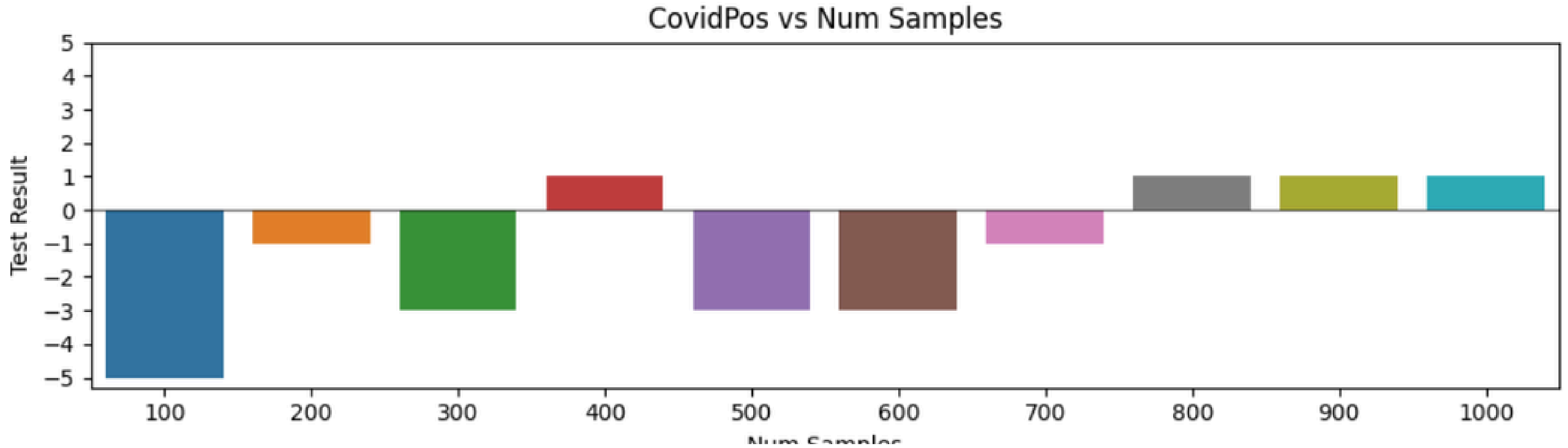




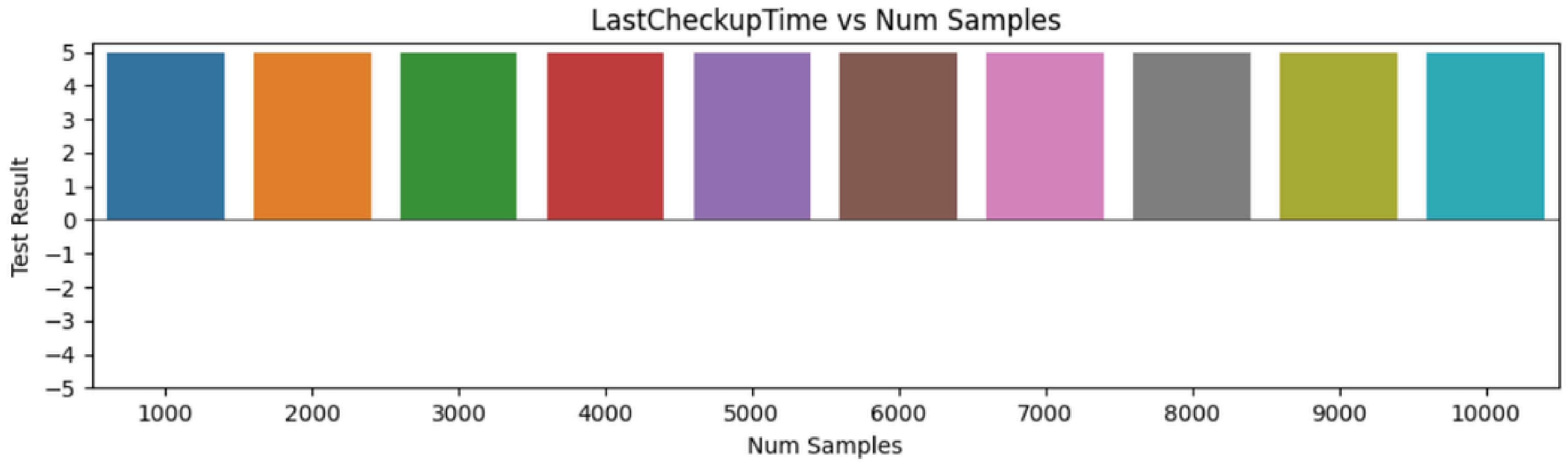
χ^2 RESULTS: STRATIFIED SAMPLING

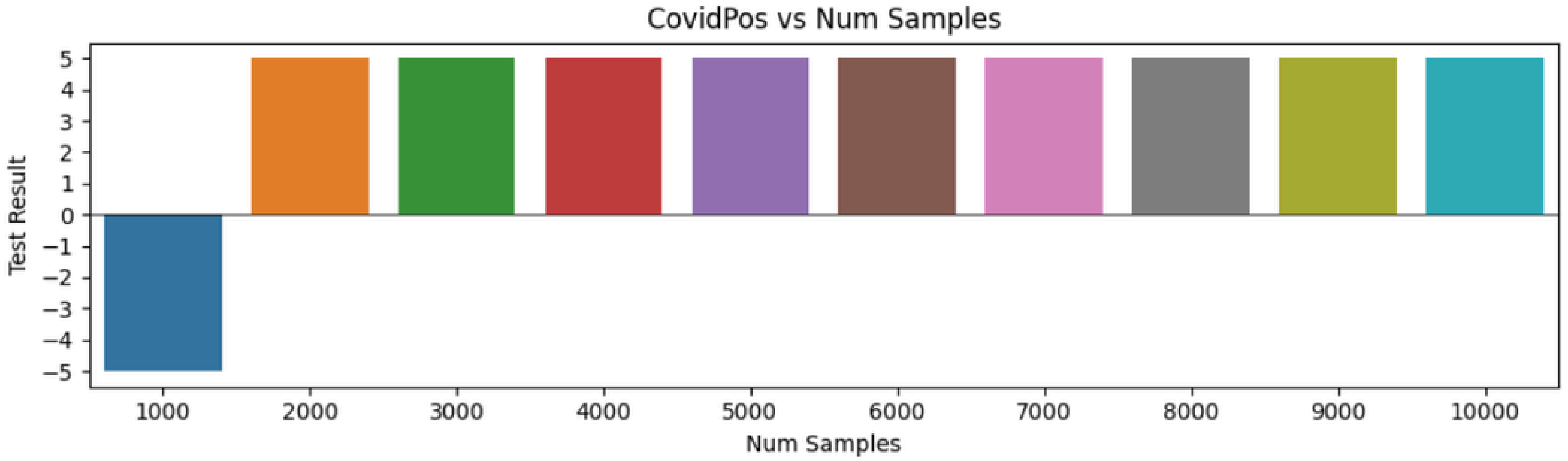
Samples: 100 - 1000





Samples: 1000 - 10000





Note: The x-axis ticks represent n number of samples per class. (For example, 100 means 100 samples per class)

χ^2 TEST FOR INDEPENDENCE : VALIDATION

Column-1	Column-2	Test Statistic	Critical Value	Conclusion
Sex	GeneralHealth	263.4067	13.2767	Dependent (Reject Ho)
PhysicalActivities	GeneralHealth	30766.4642	13.2767	Dependent (Reject Ho)
RemovedTeeth	GeneralHealth	30049.0960	26.2169	Dependent (Reject Ho)
HadSkinCancer	GeneralHealth	426.6530	13.2767	Dependent (Reject Ho)
DifficultyWalking	GeneralHealth	72629.8900	13.2767	Dependent (Reject H0)
SmokerStatus	GeneralHealth	11043.3808	26.2169	Dependent (Reject Ho)
AgeCategory	GeneralHealth	8250.446	73.6826	Dependent (Reject Ho)
AlcoholDrinkers	GeneralHealth	11231.749	13.2767	Dependent (Reject Ho)
HighRiskLastYear	GeneralHealth	11.3720	13.2767	Independent (Accept Ho)
CovidPos	GeneralHealth	562.5706	20.0902	Dependent (Reject Ho)

χ^2 TEST FOR INDEPENDENCE: INFERENCE

- We tested the independence of the target GeneralHealth column against all the categorical columns available in the dataset; salient tests performed are listed in the previous slide.

- We observed that common factors that are generally known to be related to health problems like age, alcohol consumption, COVID-19, and smoking status are not independent of the GeneralHealth column and hence have an effect on it.

- We also observed that factors like gender, and skin cancer also affected health of a person.

- Among all the categorical columns only the **HighRiskLastYear (**The person has injected any drug other than those prescribed for him/her in the past year or the person has contracted any STI in the previous year**)** column is the only column which showed independence for all of the rows.

Z-TEST FOR PROPORTION: TESTS AND EXPERIMENTS

- According to the World Health Organization, 43% of people are classified as overweight (Source). Based on this statistic, we have used **40%** as a reasonable estimate for the proportion of overweight individuals in our hypothesis.
- We have tested it for 10% (nearly 40,000 samples) of our data using stratified and random sampling. The results are written below.

Random Sampling

Z-Stat Value: 117.585
Critical Value: 1.644
Result: Fail to Reject Ho

Stratified Sampling

Z-Stat Value: 118.810
Critical Value: 1.644
Result: Fail to Reject Ho

Z-TEST FOR PROPORTION: VALIDATION

- To validate our experiments, we conducted Z-Test for proportion on our entire dataset and we observed that more than 40% of the people are overweight. The median weight and height values that we received on our dataset are **81.19 kg** and **1.7 m (170 cm)**. Calculating the BMI on these values we get **28** (Range of overweight is 25.0 - 29.9) which is in accordance with the result of the z-test.

Complete Dataset

Z-Stat Value: 350.398

Critical Value: 1.644

Result: Fail to Reject Ho

ONE-WAY ANOVA TEST : TESTS AND EXPERIMENTS

- The motivation behind conducting this test was to analyze if there is variation in the mean BMI of the people that lie in various categories of health, for this we conducted the ANOVA test

- We tested that the mean BMI for all the classes (Poor, Fair, Good, Very Good, Excellent) in the "GeneralHealth" is the same for all of the classes i.e. mean of "poor" class = mean of "fair" class = mean of "good" class = mean of "very good" class = mean of "excellent" class.

- We have tested it for 5% (nearly 20,000) and 10% (nearly 40,000) data using stratified and random sampling. The results are written below.

Random Sampling

	20,000	**40,000**
F-Stat	328.578	662.152
Critical Value	2.372	2.372
Result	Reject H0	Reject H0

Stratified Sampling

	20,000	**40,000**
F-Stat	360.863	680.170
Critical Value	2.372	2.372
Result	Reject H0	Reject H0

ONE-WAY ANOVA TEST : VALIDATION

- To validate our experiments, we performed the ANOVA test upon our entire dataset and we observed that the mean BMI of all the classes is **NOT** the same.

- This result is in accordance to the tests we conducted, indicating that the mean BMI of different categories of health ranging from poor to excellent cannot be the same

Complete Dataset

F-Stat Value: 5849.724

Critical Value: 2.371

Result: Reject H0

MODELS USED AND RESULTS

Model	Accuracy	Macro F1
Naive Bayes	0.3257	0.3438
Logistic Regression	0.4534	0.3717
Decision Trees	0.3564	0.3369
Random Forest	0.4533	0.4169
Support Vector Classifier	0.4502	0.3304
AdaBoost	0.4571	0.3821
XGBoost	0.4725	0.4360

Reasons for choosing these models:

Naive Bayes

It is a simple probabilistic model that assumes feature independence. Its speed and efficiency made it a strong baseline for our task.

Decision Trees

It is a highly interpretable model that splits data based on feature thresholds. It works well as a baseline due to its simplicity.

AdaBoost

It iteratively focuses on misclassified examples, creating a robust ensemble of weak learners. It handles noise well and improves accuracy on challenging datasets.

Logistic Regression

It is a linear model ideal for classification tasks. It serves as a benchmark for comparing with more complex models and performs well when the data is approximately linearly separable.

Random Forests

It improves upon decision trees by combining multiple trees, increasing accuracy and reducing overfitting. It also provides valuable insights into feature importance.

XGBoost

It is an efficient gradient boosting algorithm with built-in regularization to prevent overfitting. Its ability to handle missing data make it reliable for complex tasks.

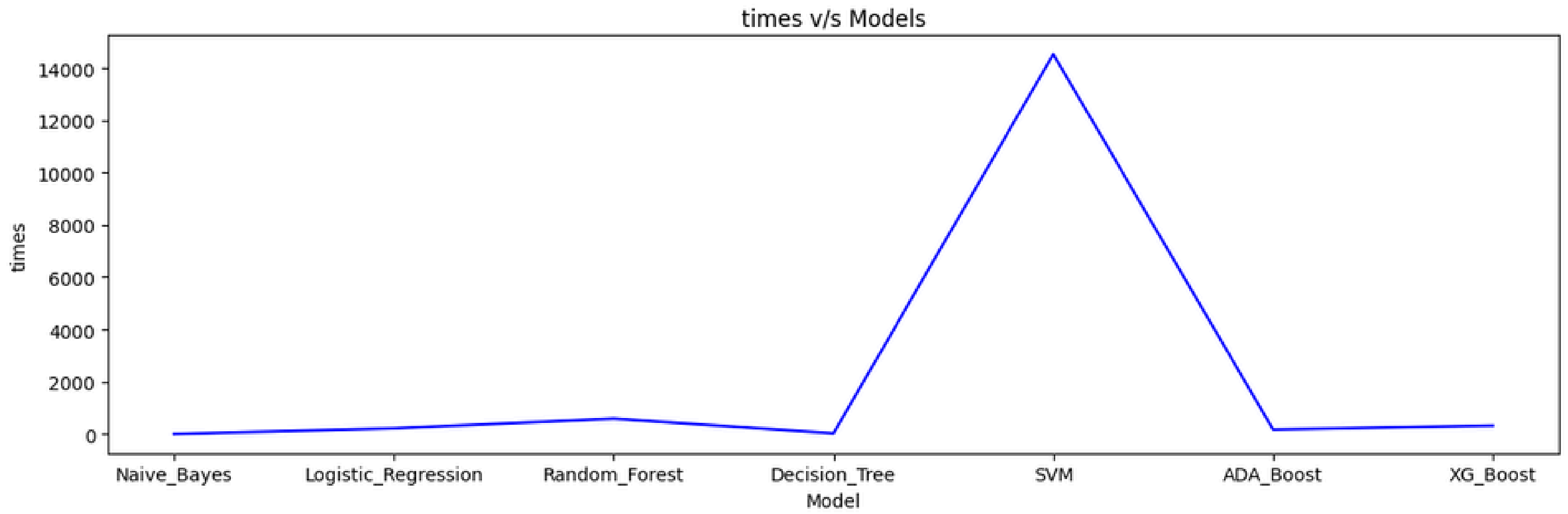
Support Vector Classifier

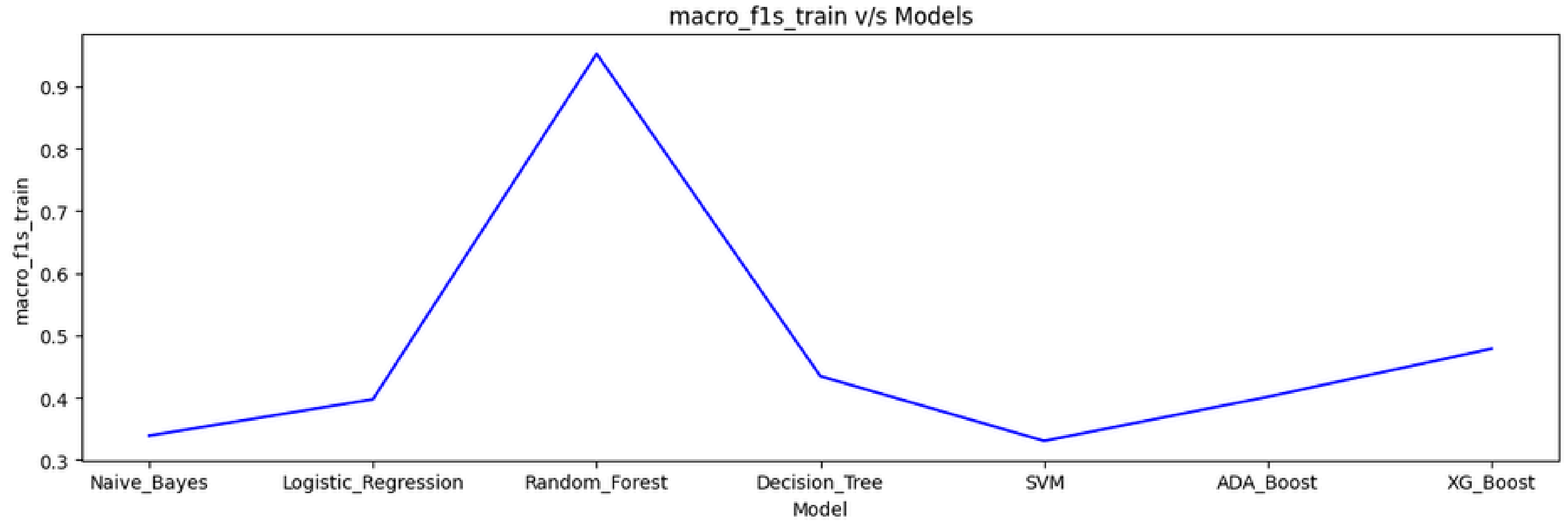
It is effective for high-dimensional datasets and finds the decision boundary that maximizes the margin between classes. Kernel functions enable it to handle non-linear patterns.

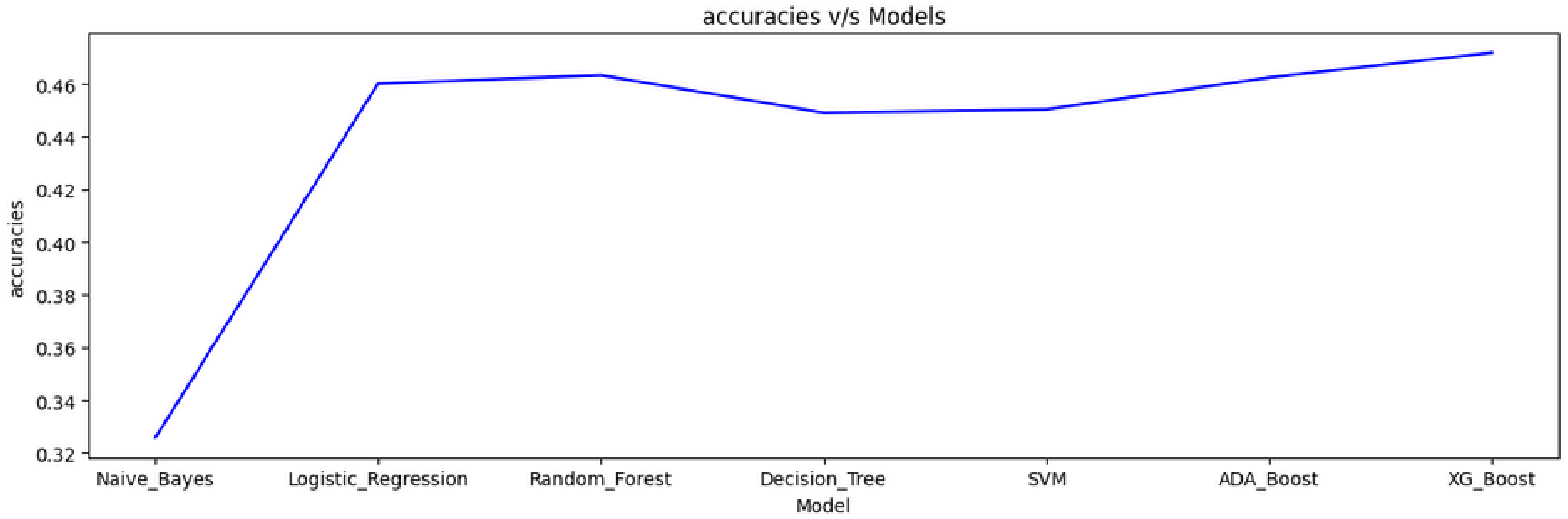
GRID SEARCH RESULTS

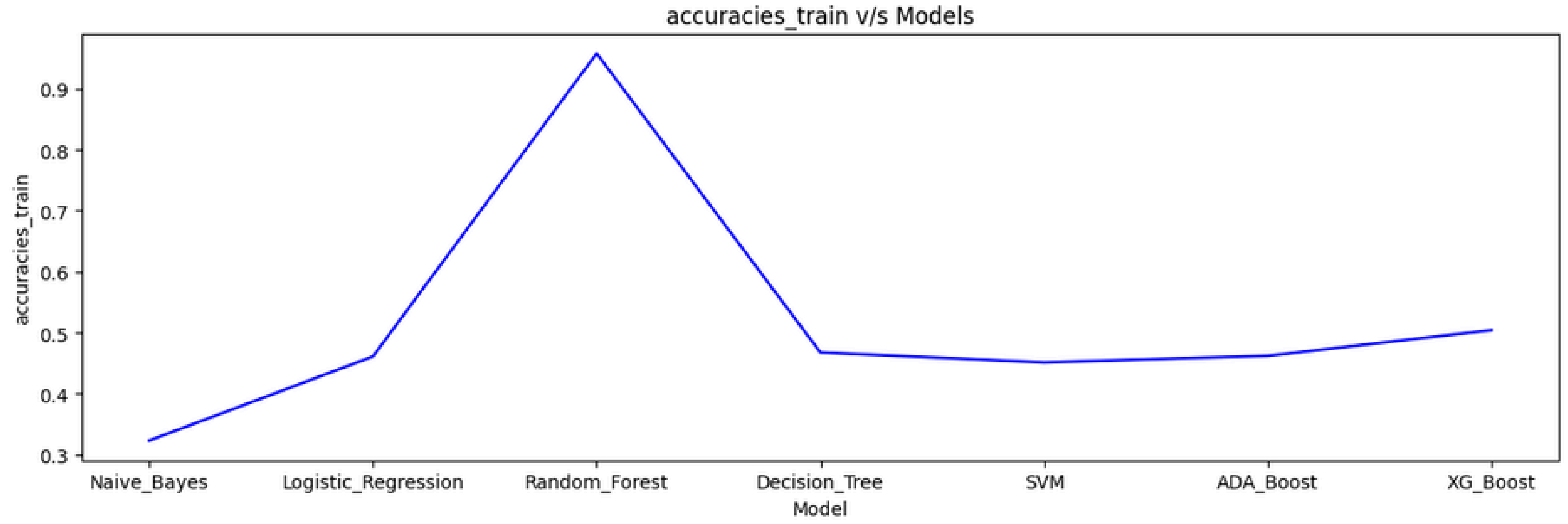
Model	**Parameters**	**Accuracy (Test | Train)**		**Macro F1 (Test | Train)**		**Time (in seconds)**
Naive Bayes	-	0.3257	0.3231	0.4718	0.3390	0.2980
Logistic Regression	{'C': **0.1**, 'penalty': '**l2**', 'solver': '**saga**'}	0.4600	0.4609	0.3973	0.3973	216.5667
Decision Trees	{'criterion': '**gini**', 'max_depth': **10**, 'min_samples_leaf': **1**, 'min_samples_split': **2**}	0.4490	0.4676	0.4207	0.4344	25.924320
Random Forest	{'max_depth': **None**, 'min_samples_leaf': **2**, 'min_samples_split': **5**, 'n_estimators': **100**}	0.4632	0.9577	0.4068	0.9523	588.541620
Support Vector Classifier	-	0.4502	0.4511	0.3304	0.3306	14529.0550
AdaBoost	{'learning_rate': **1**, 'n_estimators': **100**}	0.4623	0.4620	0.4038	0.4013	170.2865
XGBoost	{'learning_rate': **0.1**, 'max_depth': **7**, 'n_estimators': **100**, 'subsample': **0.8**}	0.4718	0.5042	0.4307	0.4786	321.5975

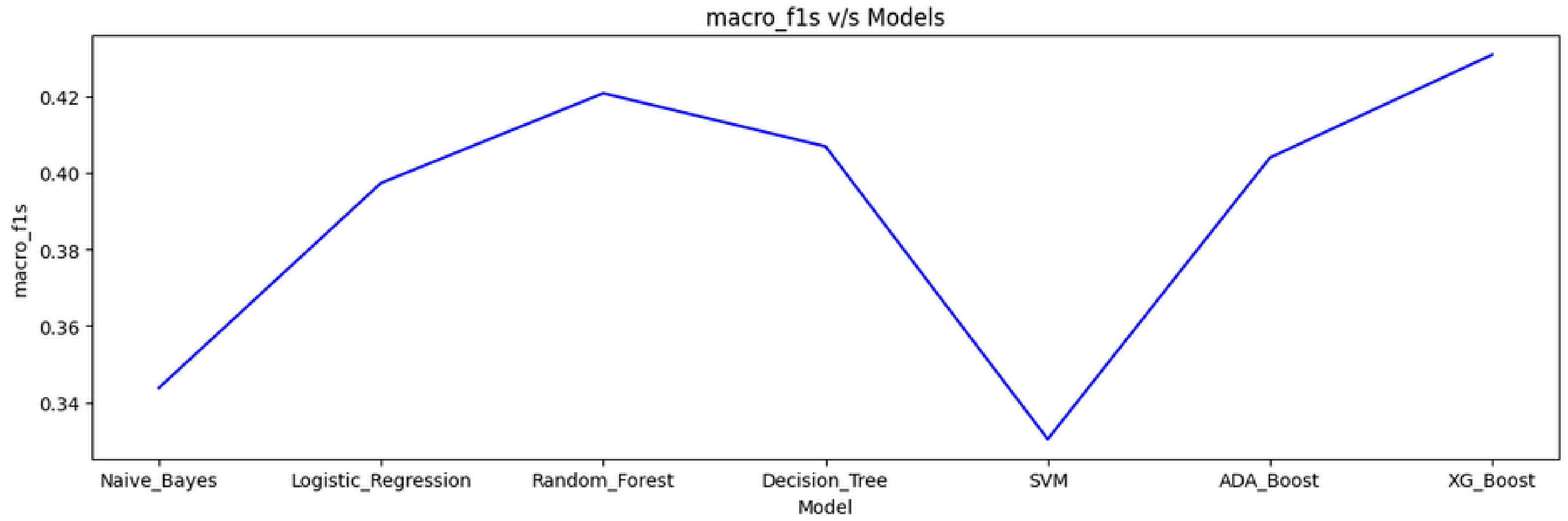
GRAPH


times v/s Models
times
14000
12000
10000
8000
6000
4000
2000
0
Naive_Bayes
Logistic_Regression
Random_Forest
Decision_Tree
SVM
ADA_Boost
XG_Boost
Model

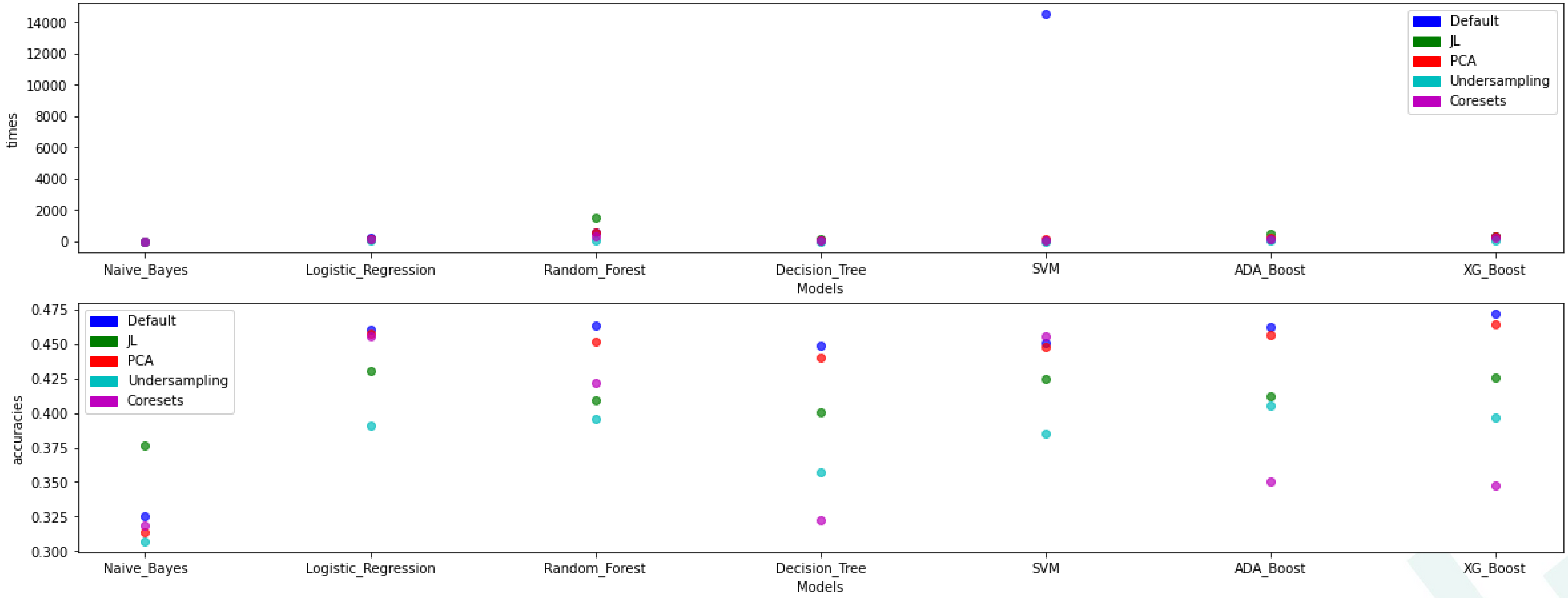

macro_f1s_train v/s Models
macro_f1s_train
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Naive_Bayes
Logistic_Regression
Random_Forest
Decision_Tree
SVM
ADA_Boost
XG_Boost
Model


accuracies v/s Models
accuracies
0.46
0.44
0.42
0.40
0.38
0.36
0.34
0.32
Naive_Bayes
Logistic_Regression
Random_Forest
Decision_Tree
SVM
ADA_Boost
XG_Boost
Model


accuracies_train v/s Models
accuracies_train
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Naive_Bayes
Logistic_Regression
Random_Forest
Decision_Tree
SVM
ADA_Boost
XG_Boost
Model

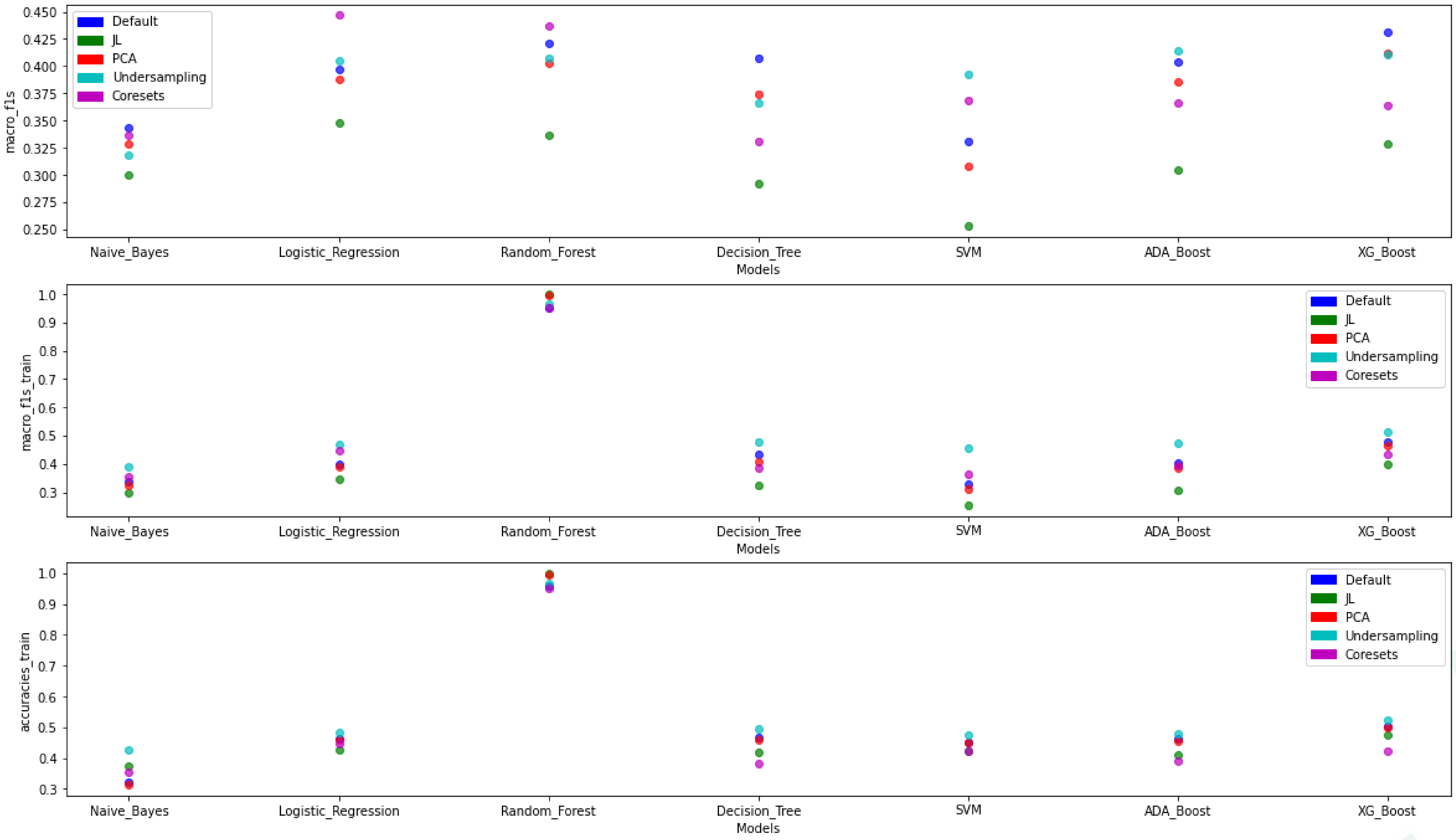

macro_f1s v/s Models
macro_f1s
0.42
0.40
0.38
0.36
0.34
Naive_Bayes
Logistic_Regression
Random_Forest
Decision_Tree
SVM
ADA_Boost
XG_Boost
Model

COMPARISON



14000
12000
10000
8000
6000
4000
2000
0
times
Default
JL
PCA
Undersampling
Coresets
Naive_Bayes
Logistic_Regression
Random_Forest
Decision_Tree
SVM
ADA_Boost
XG_Boost
Models
0.475
0.450
0.425
0.400
0.375
0.350
0.325
0.300
accuracies
Default
JL
PCA
Undersampling
Coresets
Naive_Bayes
Logistic_Regression
Random_Forest
Decision_Tree
SVM
ADA_Boost
XG_Boost
Models

COMPARISON



macro_f1s
0.450
0.425
0.400
0.375
0.350
0.325
0.300
0.275
0.250
Default
JL
PCA
Undersampling
Coresets
Naive_Bayes
Logistic_Regression
Random_Forest
Decision_Tree
SVM
ADA_Boost
XG_Boost
Models
macro_f1s_train
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Default
JL
PCA
Undersampling
Coresets
Naive_Bayes
Logistic_Regression
Random_Forest
Decision_Tree
SVM
ADA_Boost
XG_Boost
Models
accuracies_train
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Default
JL
PCA
Undersampling
Coresets
Naive_Bayes
Logistic_Regression
Random_Forest
Decision_Tree
SVM
ADA_Boost
XG_Boost
Models

RANDOMIZED SCALING TECHNIQUES (WITH JL)

Model	Parameters	Accuracy (Test | Train)		Macro F1 (Test | Train)		Time (in seconds)
Naive Bayes	-	0.376270	0.375143	0.299769	0.298138	0.115166
Logistic Regression	{'C': **0.1**, 'penalty': '**l2**', 'solver': '**saga**'}	0.430953	0.428326	0.347704	0.345526	115.568049
Decision Trees	{'criterion': '**gini**', 'max_depth': **10**, 'min_samples_leaf': **1**, 'min_samples_split': **2**}	0.400758	0.417396	0.292501	0.323228	116.007687
Random Forest	{'max_depth': **None**, 'min_samples_leaf': **2**, 'min_samples_split': **5**, 'n_estimators': **100**}	0.409282	0.999975	0.336829	0.999981	1534.356214
Support Vector Classifier	-	0.424993	0.423852	0.252701	0.252252	82.439969
AdaBoost	{'learning_rate': **1**, 'n_estimators': **100**}	0.412044	0.410925	0.304197	0.305299	476.258464
XGBoost	{'learning_rate': **0.1**, 'max_depth': **7**, 'n_estimators': **100**, 'subsample': **0.8**}	0.425923	0.474122	0.328420	0.399236	280.525275

RANDOMIZED SCALING TECHNIQUES (WITH PCA)

Model	Parameters	Accuracy (Test | Train)		Macro F1 (Test | Train)		Time (in seconds)
Naive Bayes	-	0.313950	0.311791	0.328641	0.324252	0.208781
Logistic Regression	{'C': **0.1**, 'penalty': '**l2**', 'solver': '**saga**'}	0.457357	0.458651	0.387441	0.390150	140.989292
Decision Trees	{'criterion': '**gini**', 'max_depth': **10**, 'min_samples_leaf': **1**, 'min_samples_split': **2**}	0.440055	0.460335	0.373969	0.408578	31.206321
Random Forest	{'max_depth': **None**, 'min_samples_leaf': **2**, 'min_samples_split': **5**, 'n_estimators': **100**}	0.451637	0.996157	0.402871	0.995456	552.510053
Support Vector Classifier	-	0.447649	0.449637	0.307573	0.309674	98.168821
AdaBoost	{'learning_rate': **1**, 'n_estimators': **100**}	0.456469	0.455889	0.385864	0.385600	194.582767
XGBoost	{'learning_rate': **0.1**, 'max_depth': **7**, 'n_estimators': **100**, 'subsample': **0.8**}	0.464416	0.500706	0.411666	0.465401	272.502932

RANDOMIZED SCALING TECHNIQUES (WITH UNDERSAMPLING)

Model	Parameters	Accuracy (Test | Train)		Macro F1 (Test | Train)		Time (in seconds)
Naive Bayes	-	0.307427	0.427651	0.318358	0.388804	0.041223
Logistic Regression	{'C': **0.1**, 'penalty': '**l2**', 'solver': '**saga**'}	0.391402	0.481142	0.404983	0.498221	34.830495
Decision Trees	{'criterion': '**gini**', 'max_depth': **10**, 'min_samples_leaf': **1**, 'min_samples_split': **2**}	0.395305	0.965844	0.406824	0.986571	77.163552
Random Forest	{'max_depth': **None**, 'min_samples_leaf': **2**, 'min_samples_split': **5**, 'n_estimators': **100**}	0.357530	0.496554	0.386558	0.480042	5.967576
Support Vector Classifier	-	0.384780	0.473034	0.392128	0.457008	8.308846
AdaBoost	{'learning_rate': **1**, 'n_estimators': **100**}	0.405830	0.472929	0.413735	0.473821	27.482340
XGBoost	{'learning_rate': **0.1**, 'max_depth': **7**, 'n_estimators': **100**, 'subsample': **0.8**}	0.398807	0.522447	0.411202	0.512734	68.082329

RANDOMIZED SCALING TECHNIQUES (WITH CORESETS)

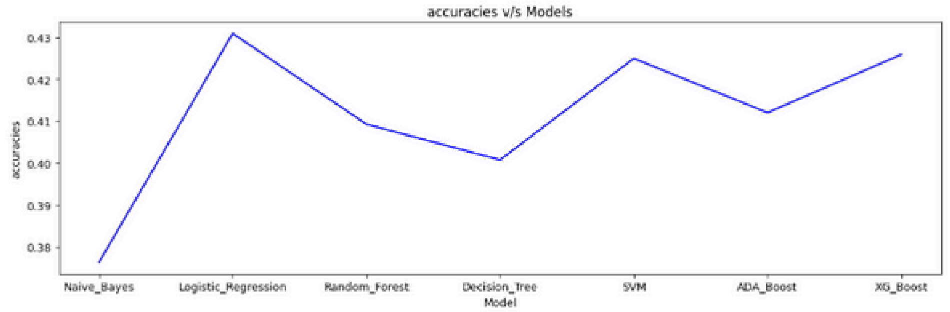
Model	Parameters	Accuracy (Test | Train)		Macro F1 (Test | Train)		Time (in seconds)
Naive Bayes	-	0.318842	0.353845	0.338145	0.357200	0.124115
Logistic Regression	{'C': **0.1**, 'penalty': '**l2**', 'solver': '**saga**'}	0.455568	0.445460	0.447112	0.449831	120.575568
Decision Trees	{'criterion': '**gini**', 'max_depth': **10**, 'min_samples_leaf': **1**, 'min_samples_split': **2**}	0.322556	0.381810	0.331083	0.383647	15.636208
Random Forest	{'max_depth': **None**, 'min_samples_leaf': **2**, 'min_samples_split': **5**, 'n_estimators': **100**}	0.422132	0.951705	0.439019	0.951329	293.374386
Support Vector Classifier	-	0.457537	0.422045	0.388889	0.362009	49.390893
AdaBoost	{'learning_rate': **1**, 'n_estimators': **100**}	0.350508	0.389000	0.386211	0.395703	111.681761
XGBoost	{'learning_rate': **0.1**, 'max_depth': **7**, 'n_estimators': **100**, 'subsample': **0.8**}	0.347752	0.423410	0.364303	0.434087	198.639795

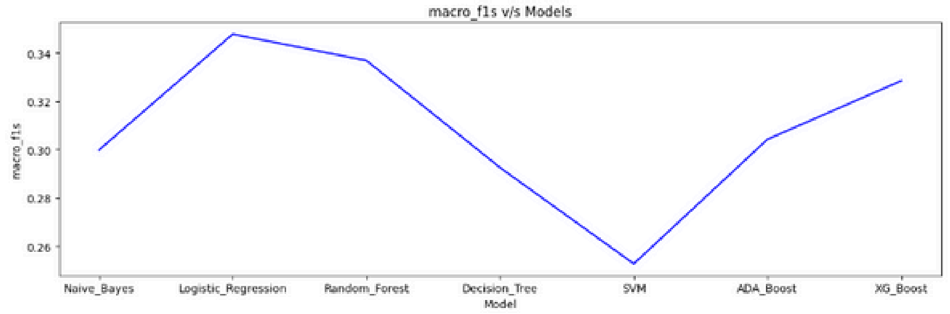
JL (JOHNSON-LINDENSTRAUSS LEMMA)

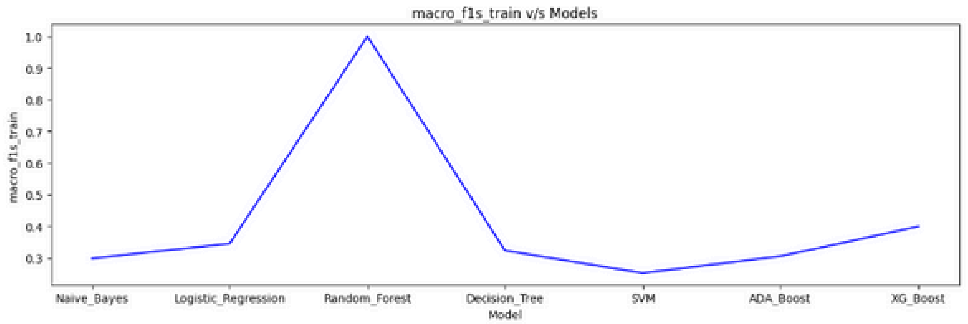
Overview: The JL Lemma is a dimensionality reduction technique that guarantees the preservation of pairwise distances between points in a high-dimensional space when mapped to a lower-dimensional space. This is particularly useful for data with high dimensionality.

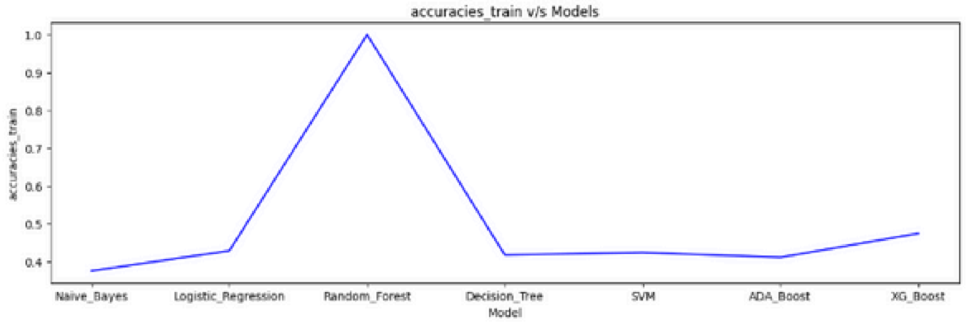
Dimensionality Reduction: Reduces the number of features while maintaining the geometric structure of the data.
Error Bound: The JL Lemma ensures that the distance between any two points is preserved up to a small error with high probability.

For our implementation we have taken JL dimentions as dX20
d=39 (number of features in our dataset)
Initial Train Dataset shape: 283889 x 39
Final Train Dataset Shape: 283889 x 20





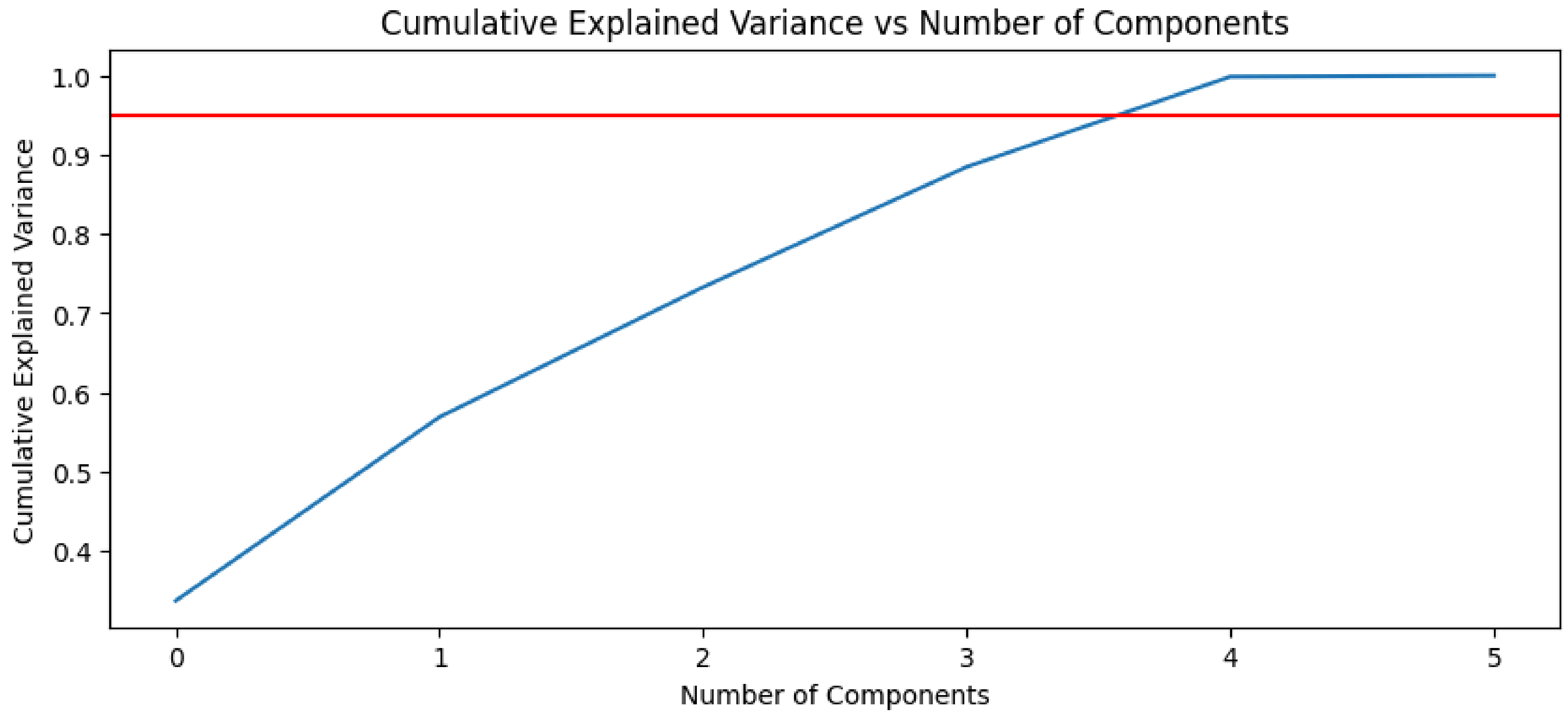


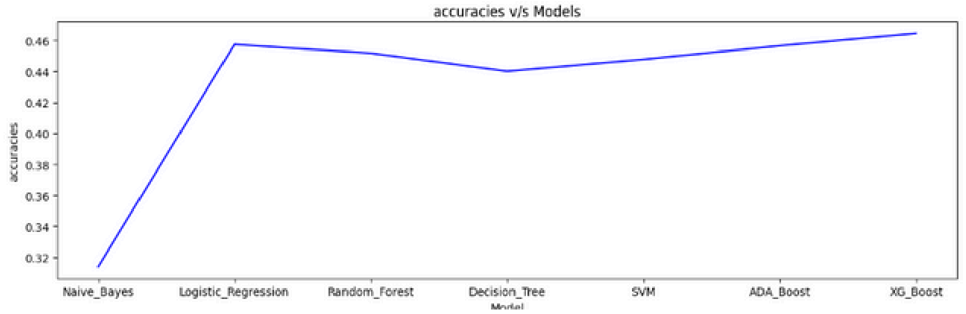


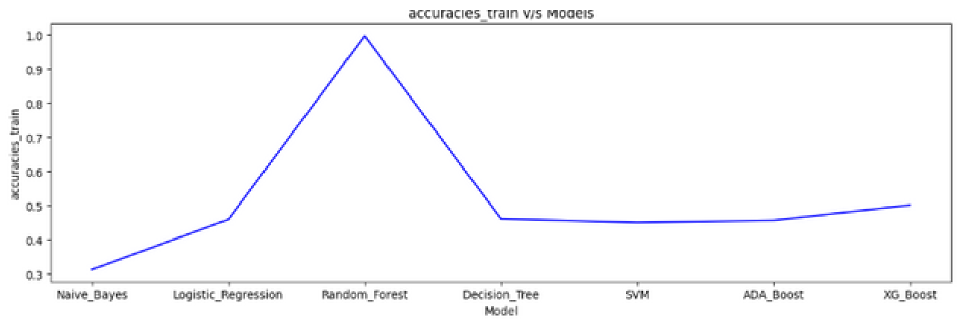
PCA (PRINCIPAL COMPONENT ANALYSIS)

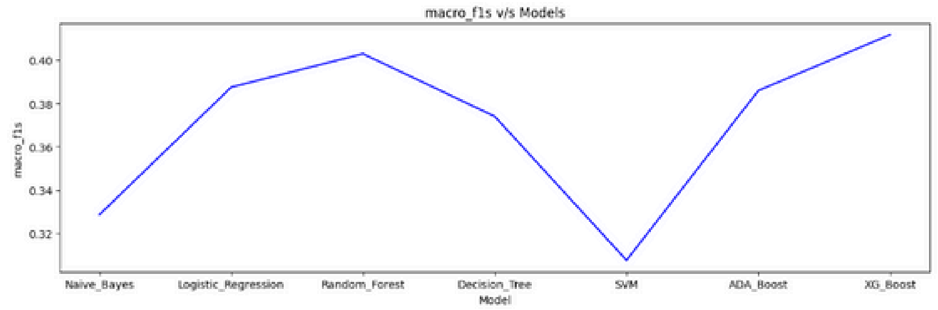
Dimensionality Reduction: By selecting only the top principal components, PCA reduces the dataset's dimensions without losing significant information.

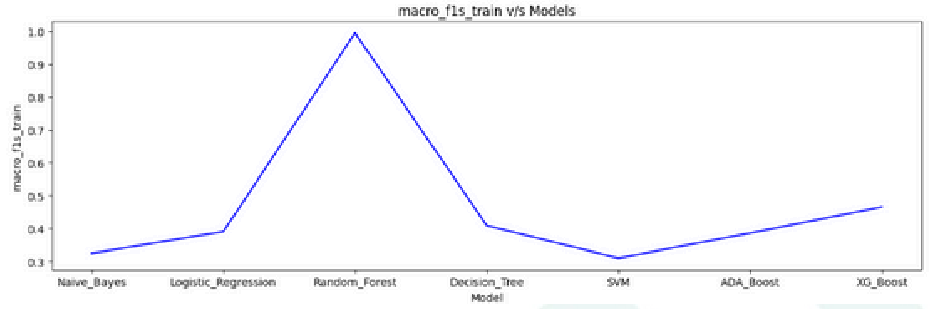
Variance Maximization: PCA identifies the directions (principal components) where the data varies the most.











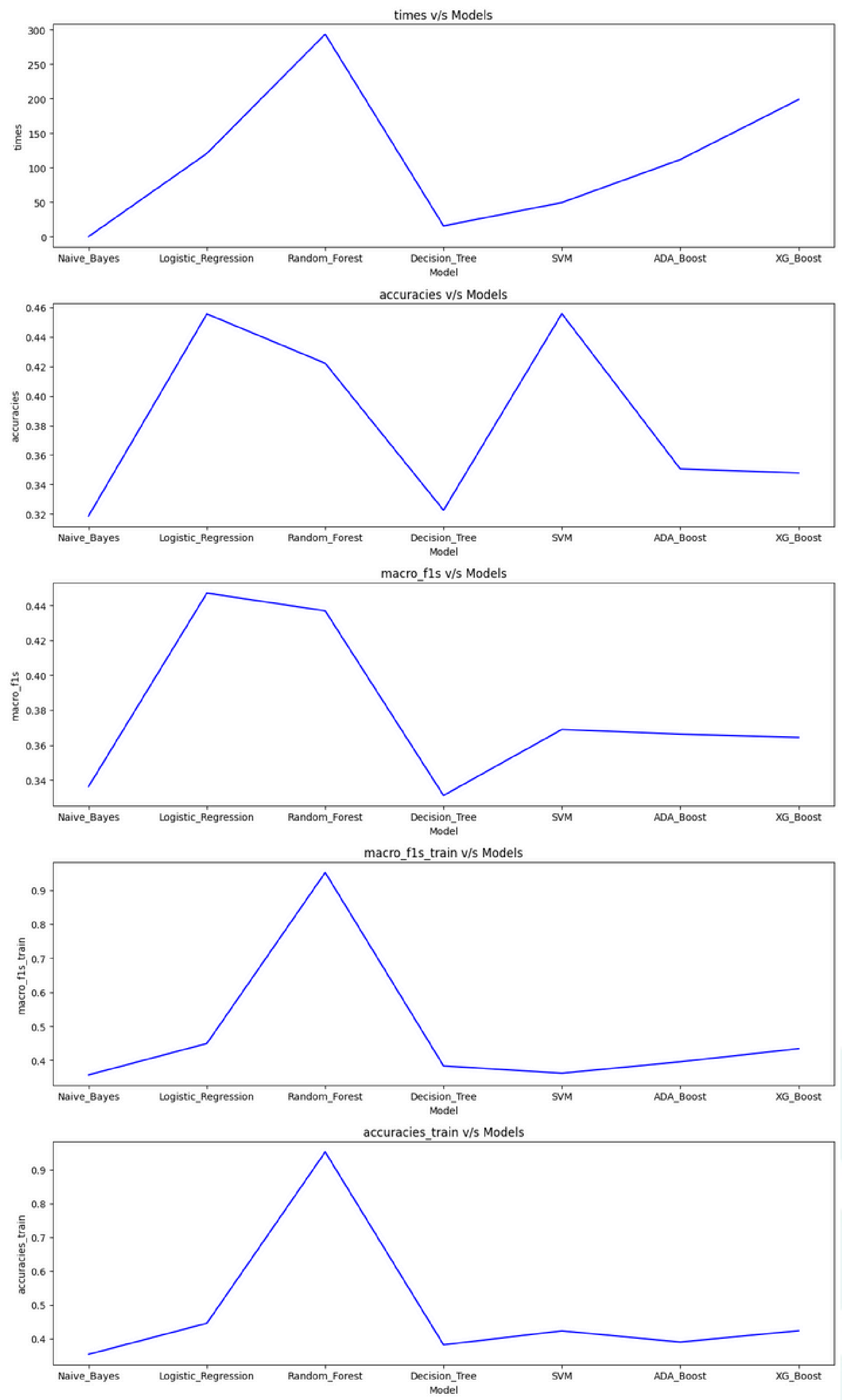
For our implementation we have taken pca components as 4(for numerical features only)
Initial Train Dataset shape: 283889 x 39
Final Train Dataset Shape: 283889 x 37

CORESETS

Key Features:

- **Coreset Selection:** Selects a representative subset of data points that approximates the original dataset in terms of class distribution and data structure.
- **Importance Weights:** Ensures underrepresented classes are sampled more frequently, maintaining class balance.
- **Probabilistic Sampling:** Uses computed weights to create a probability distribution for selecting samples, reducing bias.

For our implementation
sample_weights= class imbalance
Initial Train Dataset shape: 283889 x 39
Final Train Dataset Shape: 200000 x 39



CONCLUSION AND FUTURE WORK

We applied **seven machine learning models**, including Naive Bayes, Logistic Regression, Decision Trees, Random Forest, SVC, AdaBoost, and XGBoost, alongside scaling techniques such as PCA, JL Lemma, coresets, and randomized undersampling. **XGBoost** emerged as the best-performing model, achieving the highest accuracy while maintaining reasonable computational efficiency. The scaling methods effectively reduced training time with minimal impact on model performance, demonstrating their value for large-scale data

Future work will focus on further optimizing the models through advanced hyperparameter tuning and experimenting with additional scaling techniques. We also plan to explore DL models and domain-specific feature engineering to enhance overall performance.