July 2022

# HEART DISEASE RISK PREDICTOR

A dive into understanding features driving the risk of heart disease by developing machine learning models

**Group 9**
The University of Texas at Austin

# Meet the team!

Aishwarya Rajeev

Harshit Jain

Krish Engineer

Shreyansh Agrawal

Soumith Reddy

Tyler Cushing

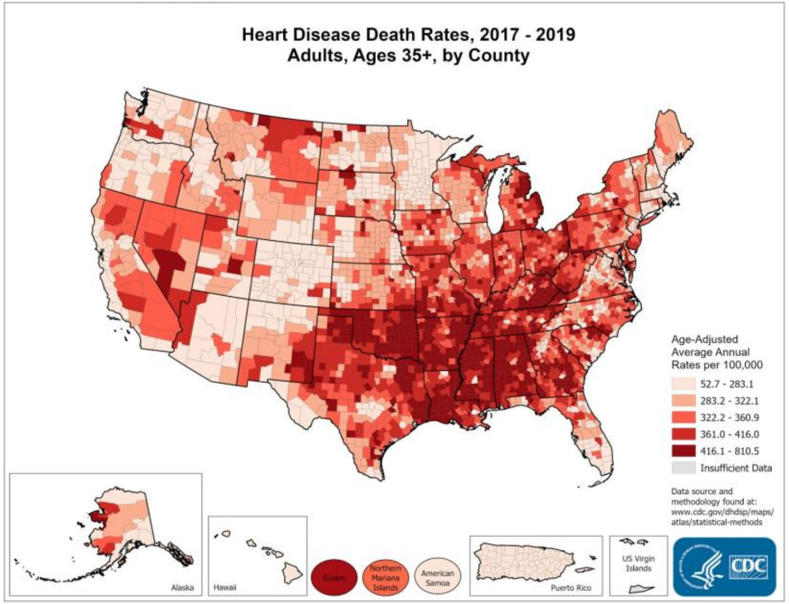
# Today we'll discuss…

- Problem Statement
- Dataset Description
- Exploratory Data Analysis
- Selected predictive model
- Comparison with other models
- Looking forward

# What are we trying to solve for…

- ~50% of Americans are in risk of future heart diseases[1]
- **Approach**: using variables to create an understanding of what factors lead to heart disease
- **Impact**: Make advancements in healthcare by being able to predict a patient's condition in relation to heart disease

Current State of Heart Disease[1]




Source: CDC Website

# Dataset Description

- **~300k adults surveyed by the CDC**
  - Largest health survey conducted in the world
- **18 variables**
  - Respondent's health status
  - Demographic info
  - Age
- **Cleaning of data:**
  - Categorical-> Numerical
  - Simplified age categories
  - Class imbalance
- **Purpose:** detect patterns in respondents' health conditions that could lead to heart disease

# Features and what they mean…

## Variables

- **HeartDisease** : Have you ever had a coronary heart disease or heart attack? (Yes / No)
- **BMI** : Body Mass Index
- **Smoking** : Have you ever smoked? (Yes / No)
- **AlcoholDrinking** : Have you ever drank alcohol? (Yes / No)
- **Stroke** : (Ever told) (you had) a stroke? (Yes / No)
- **PhysicalHealth** : How many days during the past 30 days was your physical health not good? (0-30 days)
- **MentalHealth** : How many days during the past 30 days was mental health not good? (0-30 days)
- **DiffWalking**: Difficulty walking or climbing stairs (Yes / No)
- **Sex** : Male or Female
- **Race** : Ethnicity
- **Diabetic** : (Ever told) (you had) diabetes? (Yes / No)
- **PhysicalActivity** : Doing physical activity or exercise during the past 30 days other than their regular job (Yes / No)
- **GenHealth** : General health
- **SleepTime**: Hours of sleeping in 24-hour period
- **Asthma** : (Ever told) (you had) asthma? (Yes / No)
- **KidneyDisease** : (Ever told) (you had) kidney disease? (Yes / No)
- **SkinCancer** : (Ever told) (you had) skin cancer? (Yes / No)
- **AgeCategory** : 30-44, 45-59, 60 and up

# Exploratory Data Analysis Summary

The classes are imbalanced in this dataset; less than 10% of the entire dataset have the dependent variable ‘Heart Disease’ labeled as ”yes”

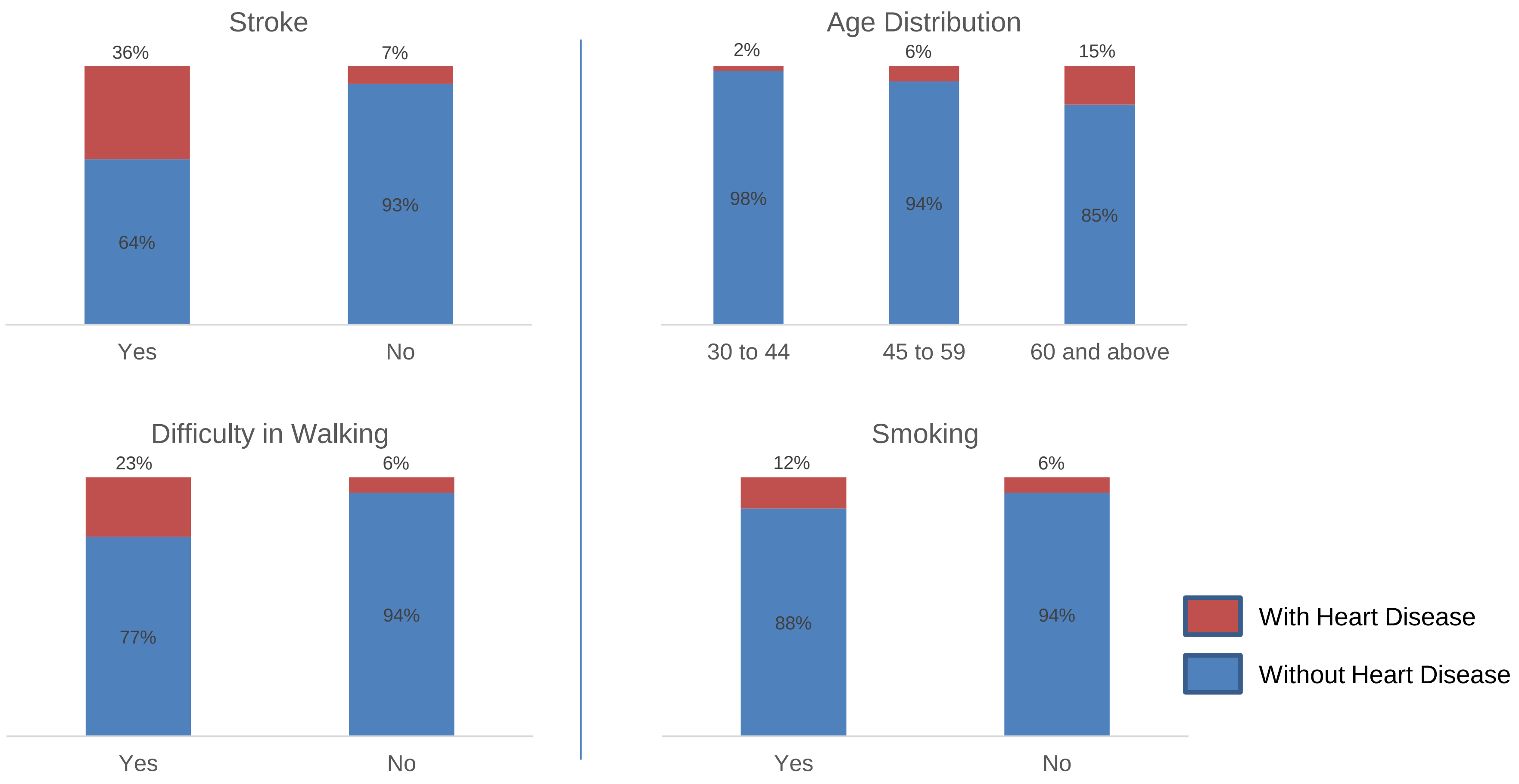
There is no missing data; all the rows and columns have a valid entry

The distributions of patients across features with and without heart diseases are similar

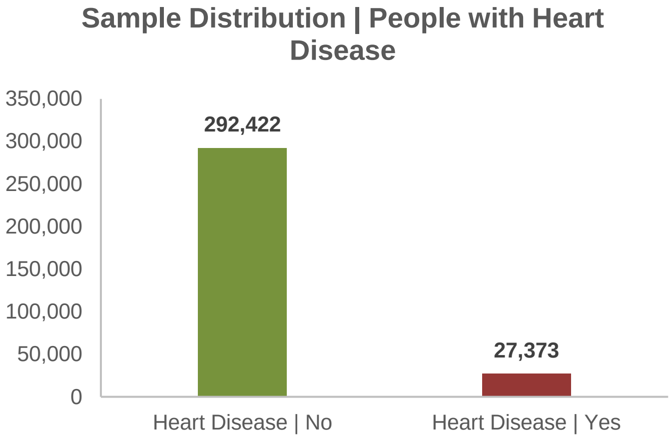
There is no significant correlation between the continuous variables

We have identified a few features such as Stroke, Smoking, Difficulty Walking, etc. have a greater impact on the possibility of having a Heart Disease, the similar trends are observed in our model output

The above information is supplemented with the relevant plots in the Appendix section


Stroke
36%
7%
64%
93%
Yes
No
Age Distribution
2%
6%
15%
98%
94%
85%
30 to 44
45 to 59
60 and above
Difficulty in Walking
23%
6%
77%
94%
Yes
No
Smoking
12%
6%
88%
94%
Yes
No
With Heart Disease
Without Heart Disease

# Class Imbalance and it's problems




- <10% of sample contains the target class
- This presents problems as we do not have enough data to explain the variance/ drivers for heart disease
- If we don't account for class imbalance, we get sub standard models* with poor predictive power

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.93 | 0.96 | 0.95 | 87635 |
| 1 | 0.42 | 0.28 | 0.34 | 8304 |
| accuracy | | | 0.90 | 95939 |
| macro avg | 0.68 | 0.62 | 0.64 | 95939 |
| weighted avg | 0.89 | 0.90 | 0.90 | 95939 |

***Ability to predict people with heart disease is <30%. That is 70 people out of 100 who have heart disease would be misclassified !!!***

*models that we have considered include Logistic Regression, Decision Trees, XGBoost, and AdaBoost. Poor predictive power was realized across these models

# What can we do to account for class imbalance?

## Cost Sensitive Learning

Penalizes the cost function by class weight, i.e., misclassification of minority class costs more

**PROS –**

- Gives more importance to classifying minority class
- True positives are predicted with more accuracy
- Avoid processing time associated with sampling methods

**CONS –**

- Determining class weight can be tricky; governed by domain knowledge

## Sampling Methods

Synthetically adjusting the class imbalance by increasing the minority class (oversampling) and reducing the majority class (under sampling)

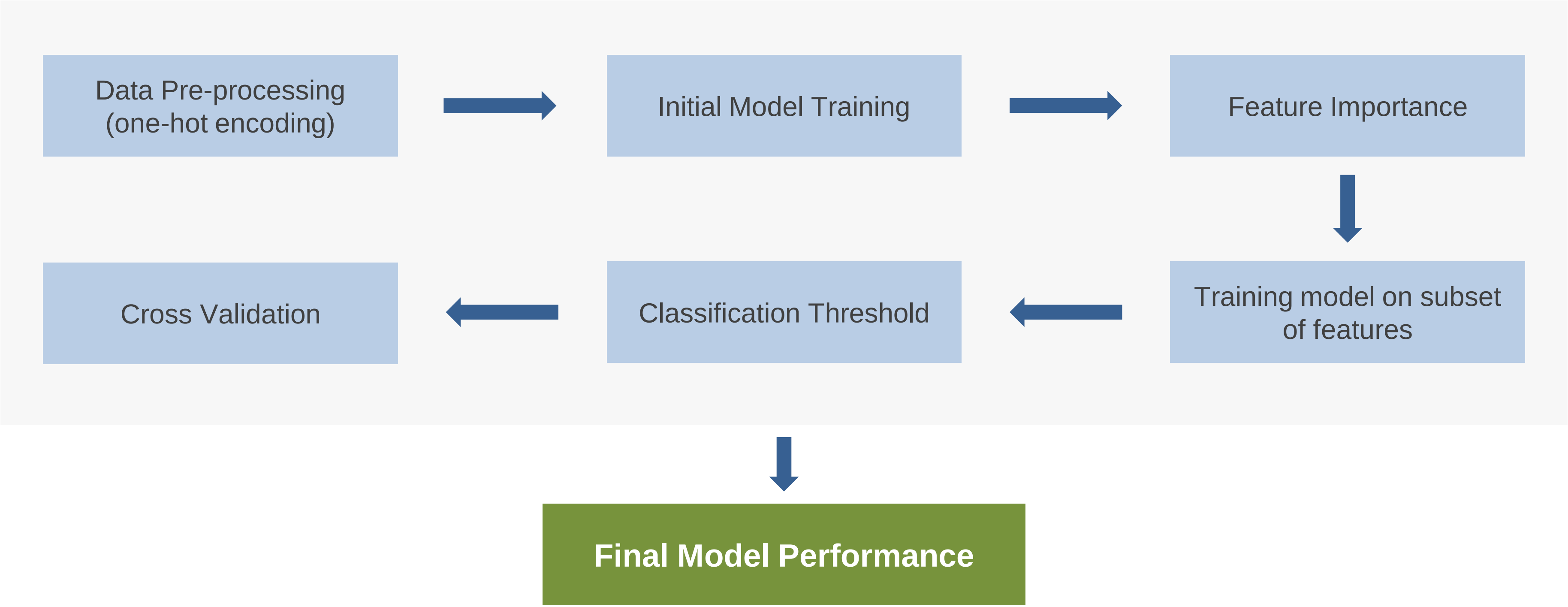
**PROS –**

- Improves class imbalance
- Provides enough data to understand the variance in each class

**CONS –**

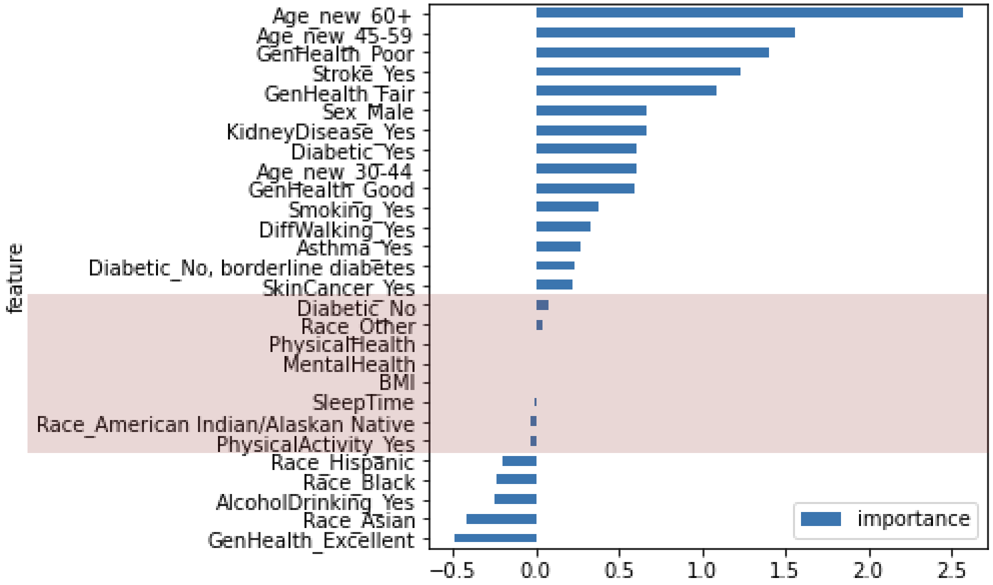
- Potential overfitting in case of oversampling
- Loss of important data in case of under sampling
- Increased processing time post oversampling

# Cost Sensitive Logistic Regression workflow



# Initial Model Results & Feature Importance

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.97 | 0.74 | 0.84 | 87635 |
| 1 | 0.22 | $\boxed{0.78}$ | 0.34 | 8304 |
| accuracy | | | 0.74 | 95939 |
| macro avg | 0.60 | 0.76 | 0.59 | 95939 |
| weighted avg | 0.91 | 0.74 | 0.80 | 95939 |



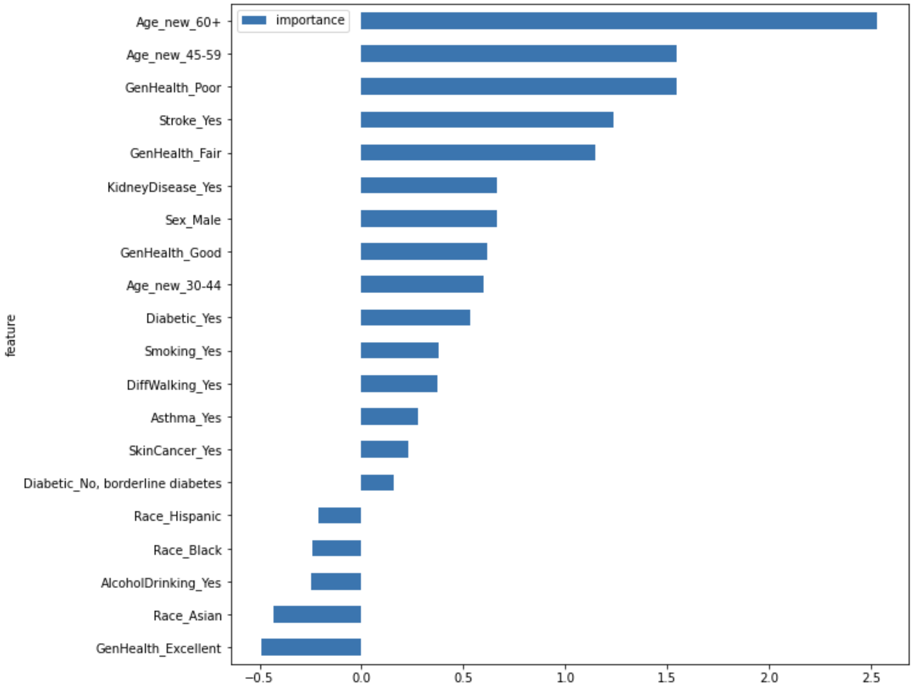

- Cost sensitive model has significantly improved the accuracy and recall from <30% to >75%
- Features marked in red had coefficients close to 0, hence lower importance and were removed from the subsequent models to reduce variance

Test train split = 30%; Classification Threshold = 0.5; Cross validation has not been done on this model.

# Tuning model to remove unnecessary features

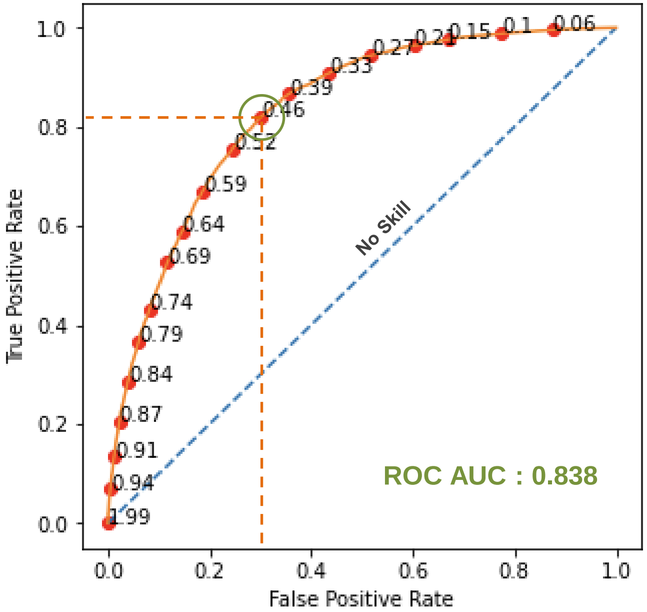
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.97 | 0.74 | 0.84 | 87635 |
| 1 | 0.22 | $\boxed{0.78}$ | 0.34 | 8304 |
| accuracy | | | 0.74 | 95939 |
| macro avg | 0.60 | 0.76 | 0.59 | 95939 |
| weighted avg | 0.91 | 0.74 | 0.80 | 95939 |

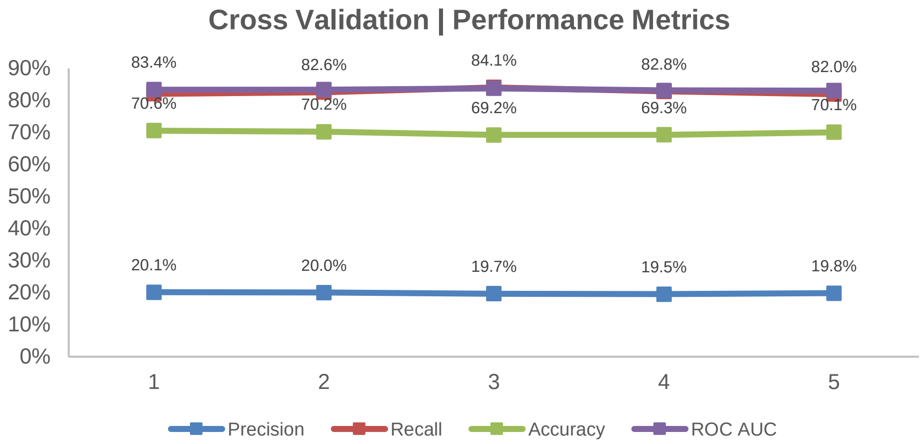
- Accuracy and recall have not decreased from the previous model suggesting the features we removed did not have any predictive power
- Additionally, all the features now have coefficients away from zero and have predictive power




Test train split = 30%; Classification Threshold = 0.5

# ROC Curve & Cross Validation



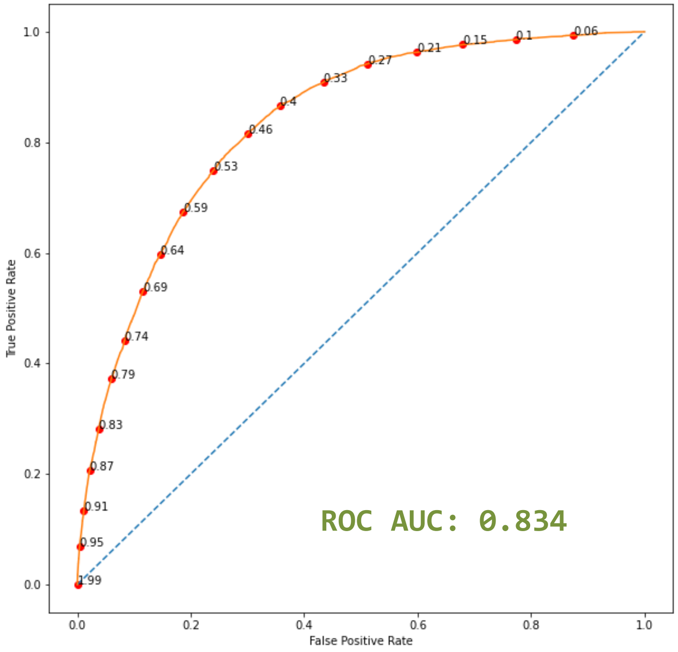




- To balance TPR and FPR, we have selected **0.45** as the classification threshold
- Cross validation shows very consistent results across validation sets making our model **generalized and reliable**

# Final Model Performance

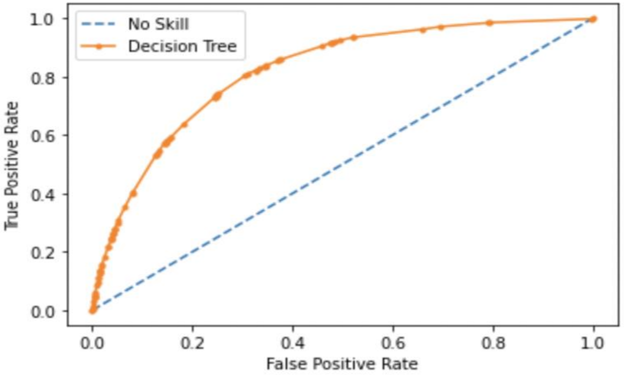
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 0.69 | 0.81 | 58484 |
| 1 | 0.20 | 0.82 | 0.32 | 5475 |
| accuracy | | | 0.70 | 63959 |
| macro avg | 0.59 | 0.76 | 0.57 | 63959 |
| weighted avg | 0.91 | 0.70 | 0.77 | 63959 |

- Out 100 of people with heart disease, our model will classify 82 of them correctly i.e., 82% recall
- Final model performance is consistent with the cross-validation results

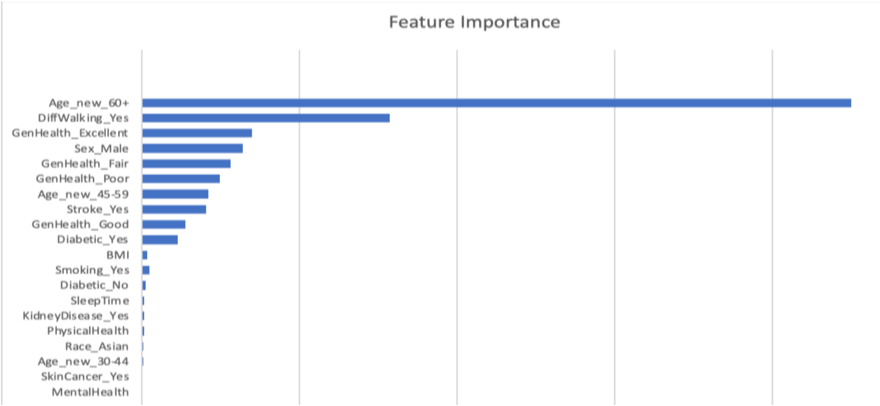


# Decision Tree Classification

```
Classification report:
              precision    recall  f1-score   support

           0       0.97      0.70      0.81     87649
           1       0.20      0.80      0.32      8290

    accuracy                           0.71     95939
   macro avg       0.59      0.75      0.57     95939
weighted avg       0.91      0.71      0.77     95939

AUROC: 0.821
```

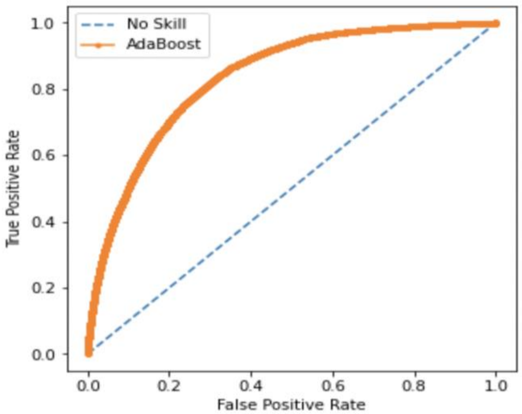


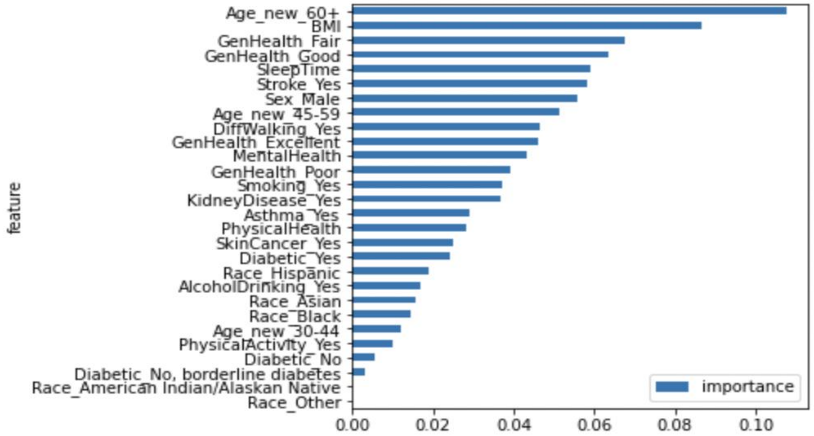

Test train split = 30%; no of trees =7




- Model was tested for various parameters (hyper parameter tuning) like depth of the tree=7  and the model gave best results
- The feature importance graph was plotted and "Age above 60" seem to be highly important
- Area Under Curve is 0.821 which means that the model can classify observations into classes well

# AdaBoost Classification



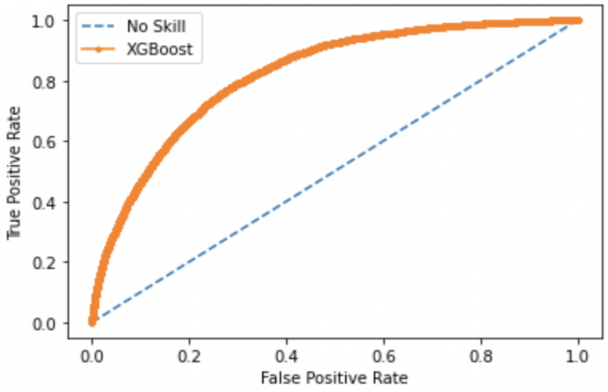




```
AUROC: 0.837
Accuracy = 0.8096410192266457
Precision Score = 0.20538475532108422
Recall Score = 0.8157514450867052
Specificity = 0.700941404689907
              precision    recall  f1-score   support

           0       0.98      0.70      0.82     87635
           1       0.21      0.82      0.33      8304

    accuracy                           0.71     95939
   macro avg       0.59      0.76      0.57     95939
weighted avg       0.91      0.71      0.77     95939
```

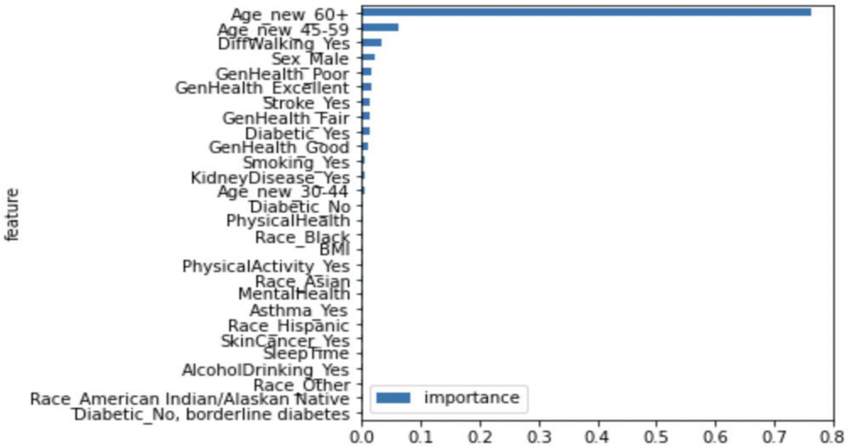
- Model was tested for various parameters (hyper parameter tuning) like depth of the tree, n_estimators and learning rate. The parameters on which the model was giving the best results was shortlisted.
- The feature importance graph was plotted shown below and was giving the expected results.

Test train split = 30%; max_depth=3, n_estimators = 50, learning_rate=0.2.

# XGBoost Classification



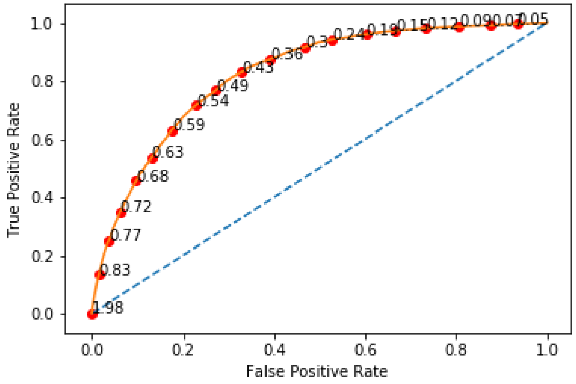

```
AUROC: 0.821
Accuracy = 0.8100071921074439
Precision Score = 0.2264117542927952
Recall Score = 0.7008219178082192
Specificity = 0.7758361261199644
              precision    recall  f1-score   support

           0       0.97      0.78      0.86     58484
           1       0.23      0.70      0.34      5475

    accuracy                           0.77     63959
   macro avg       0.60      0.74      0.60     63959
weighted avg       0.90      0.77      0.82     63959
```

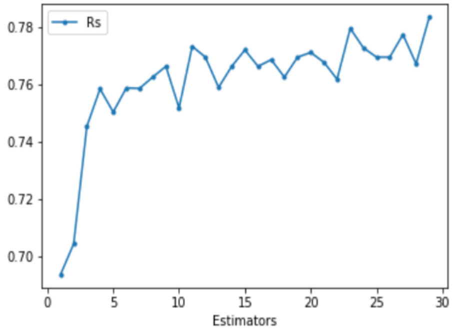


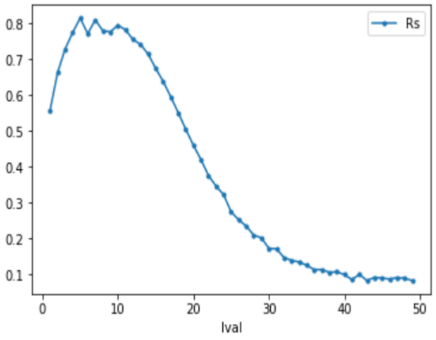

- Sample weights were taken to mitigate any effects of imbalance in datasets as number of zeroes were far greater than 1s in the 'HeartDisease_Yes' column of dataset.
- The feature importance graph was plotted shown below and was giving the expected results.
- The results of the XGBoost and AdaBoost classification were quite similar in nature.

Test train split = 30%; n_estimators = 100, max_depth= 20,alpha=10,learning_rate=0.1.

# Random Forest Classification



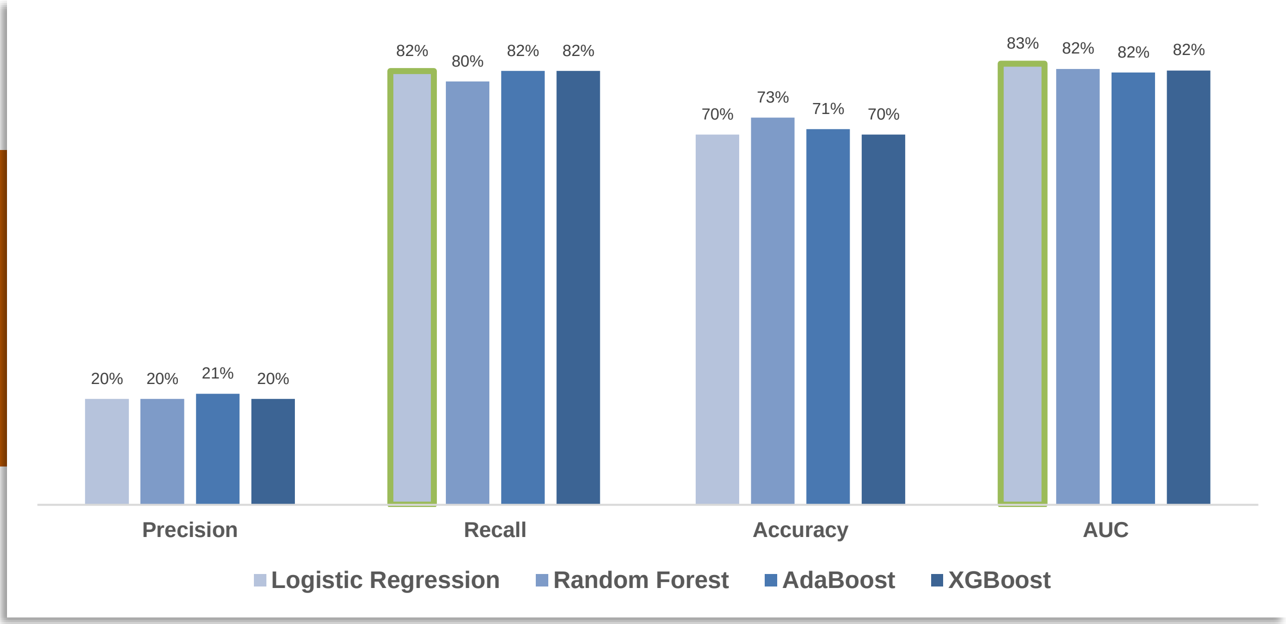

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.97 | 0.69 | 0.81 | 58367 |
| 1 | 0.20 | 0.80 | 0.32 | 5592 |
| avg / total | 0.91 | 0.70 | 0.77 | 63959 |

- Initially, Recall increases with increase in # of features and then the curve flattens post n=10
- Recall increases as depth of trees (k) increase until k ~12 and then decrease steeply with a further increase in k indicating possible overfitting for k>12

Test train split = 20%; Classification Threshold = 0.45. Max Tree Depth = 12, n_estimators = 10.

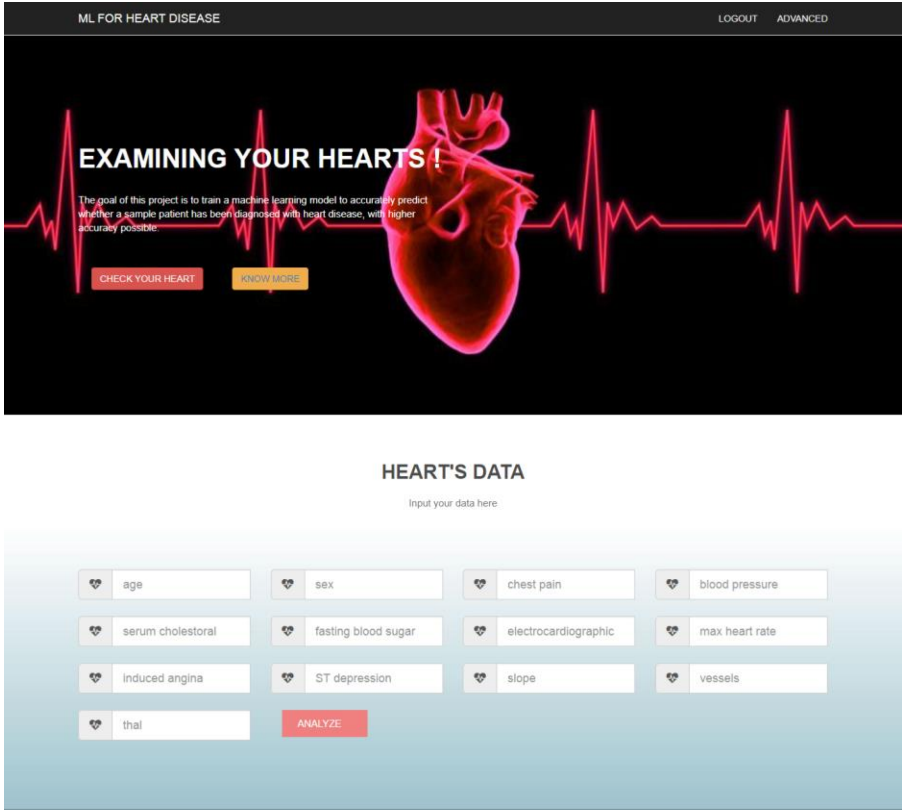
# Comparison across different models

Performance metrics across models are comparable, however, logistic regression has a better AUC and Recall score which is why we have selected it for prediction.



# Looking Forward…

- Including more granular features such as blood pressure, cholesterol levels, genetic factors etc. to improve the model accuracy
- Translate continuous variables such as BMI, Sleep time etc. into categorical variables to check their significance
- Exploring more advanced ML algorithms to improve the model
- An interactive app for predicting the chance of heart disease using the models



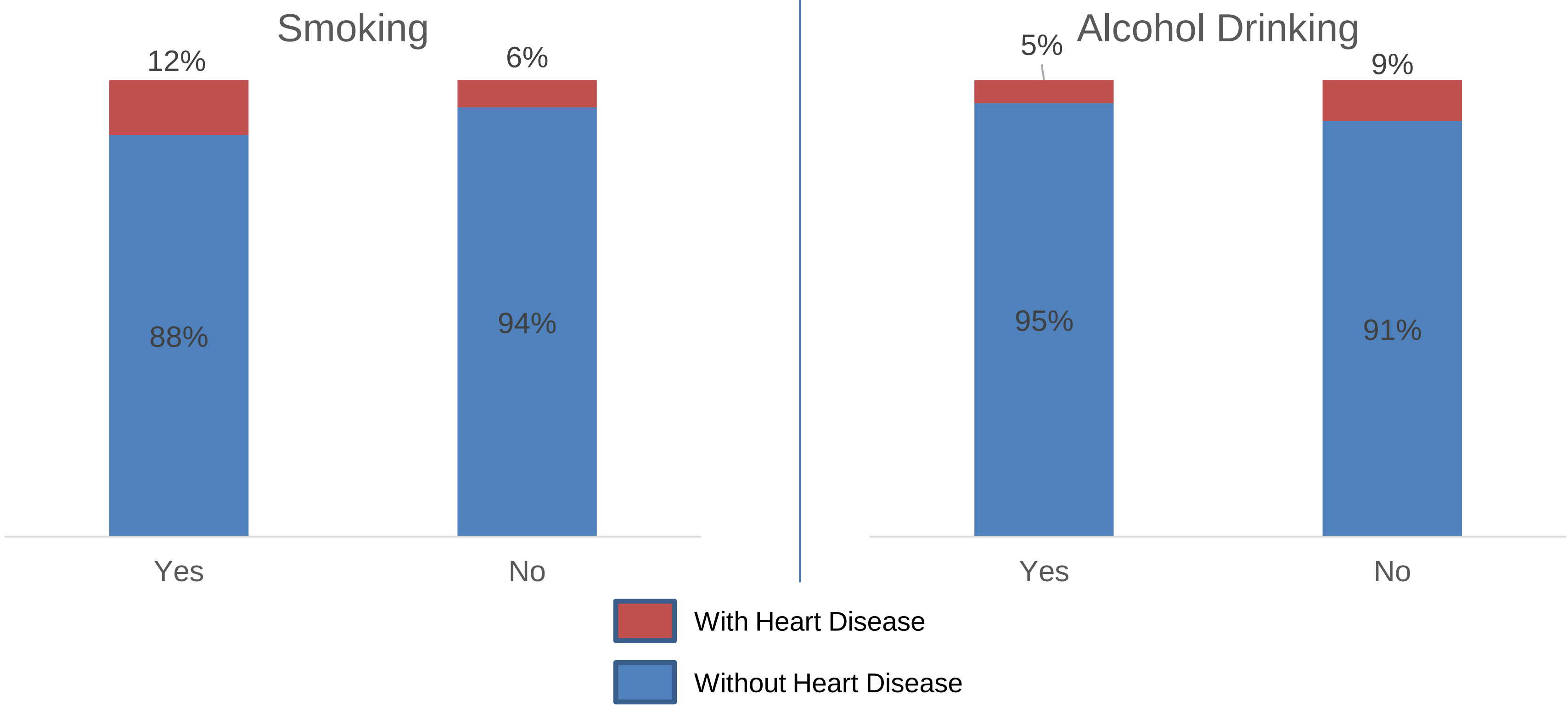
# Thank You!

# References

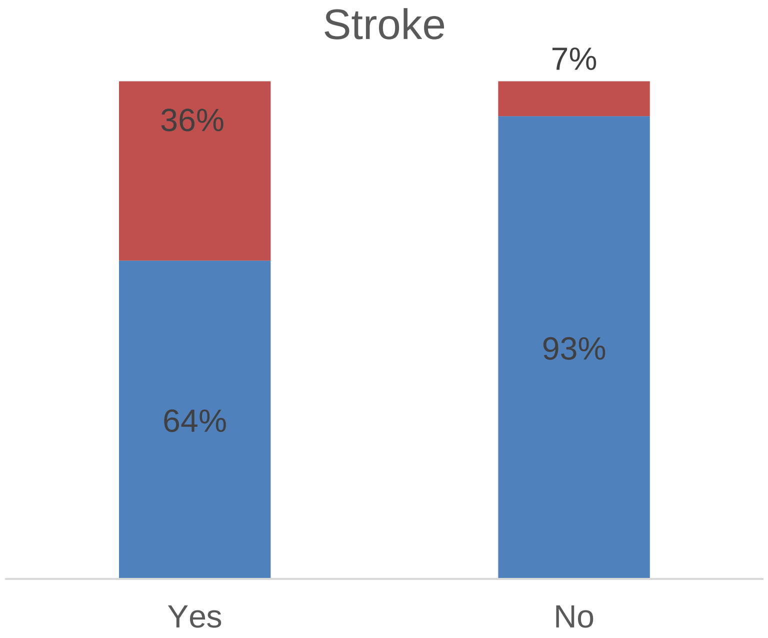

- https://www.cdc.gov/brfss/annual_data/annual_2020.html
- https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease
- https://towardsdatascience.com/how-to-effectively-predict-imbalanced-classes-in-python-e8cd3b5720c4
- https://www.datacamp.com/tutorial/random-forests-classifier-python
- https://medium.com/chinmaygaikwad/build-and-visualize-a-simple-decision-tree-using-sklearn-and-graphviz-84bda6b6b894

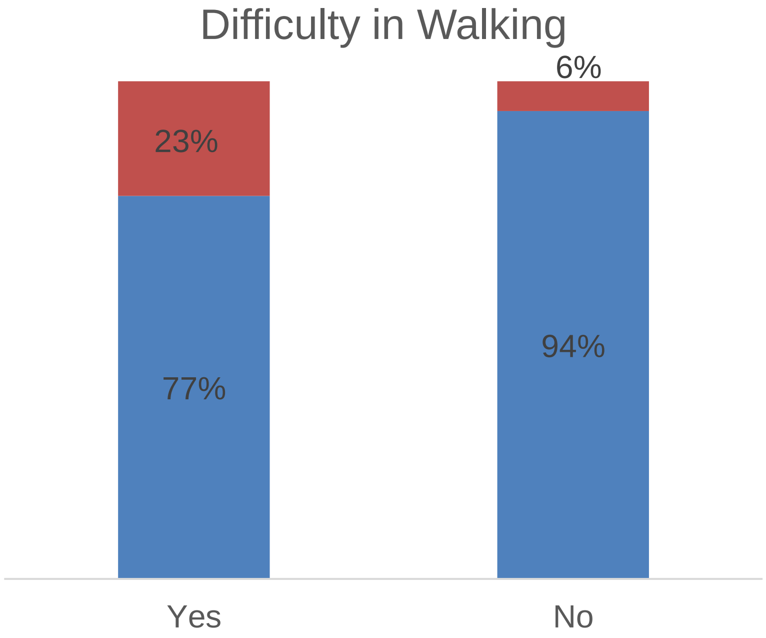
# Appendix

# Exploratory Data Analysis

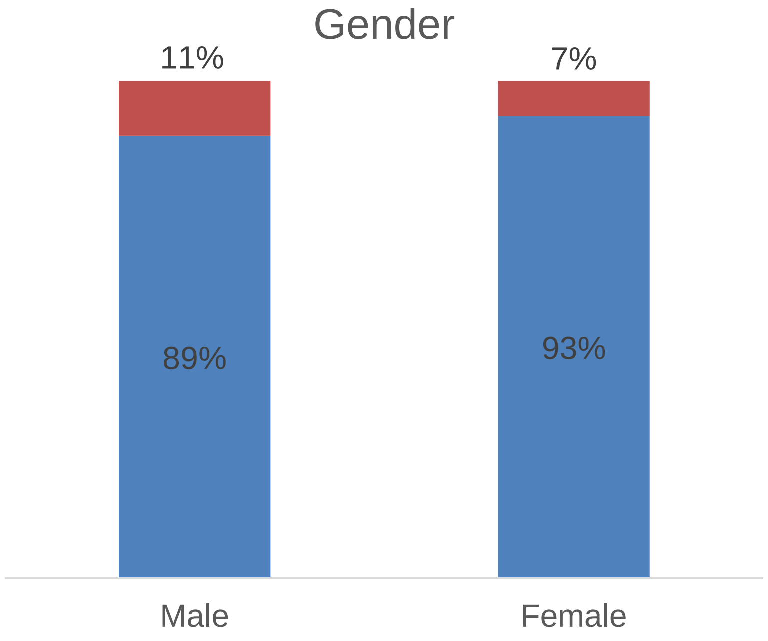


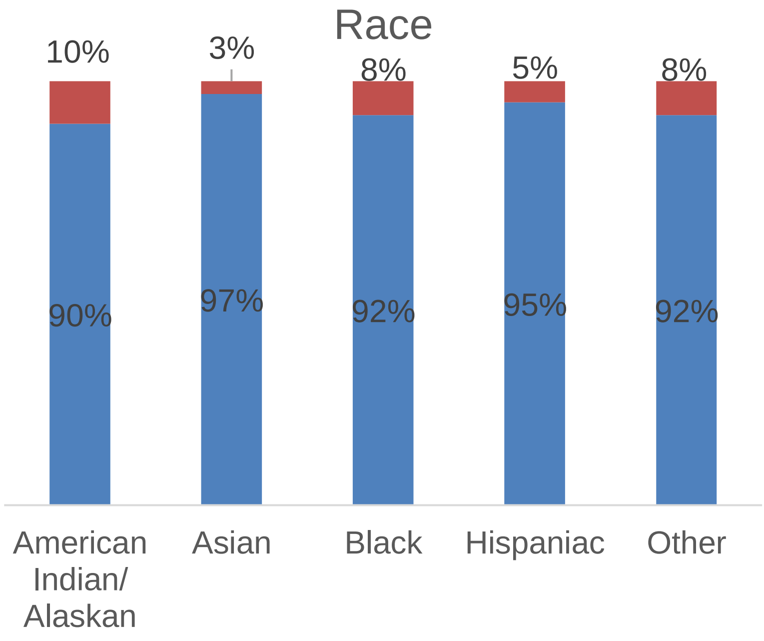
# Exploratory Data Analysis



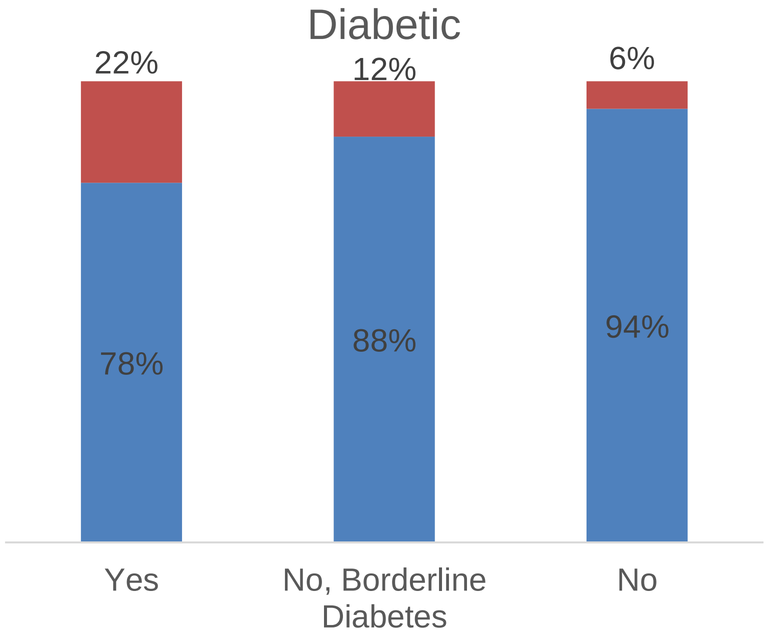



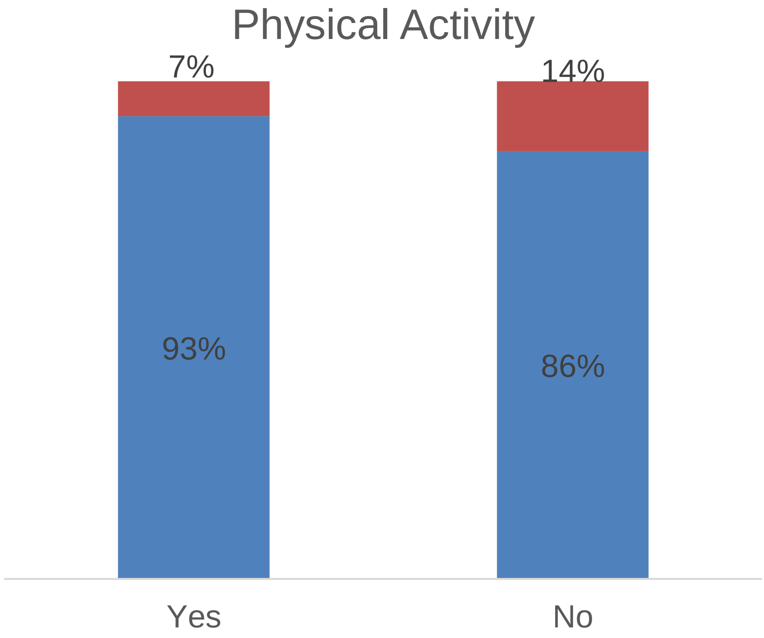
# Exploratory Data Analysis



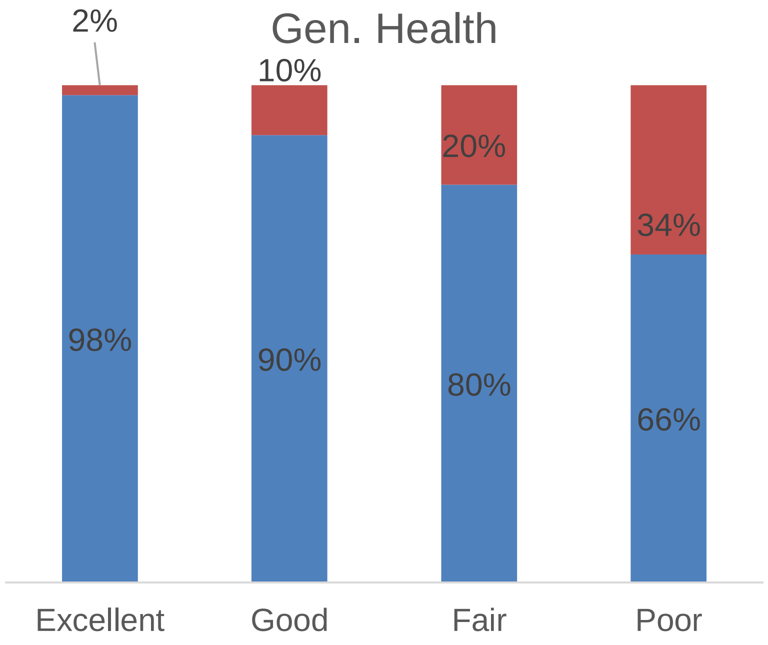



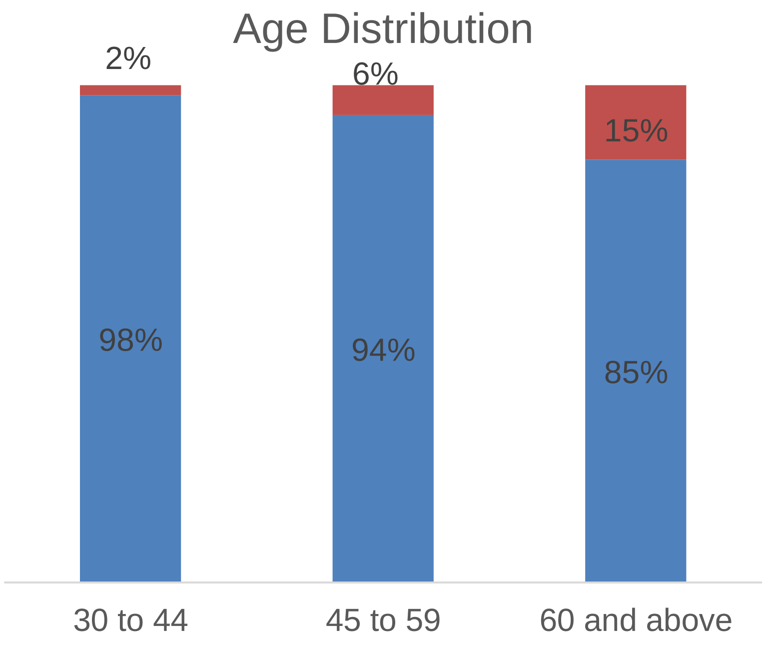
# Exploratory Data Analysis



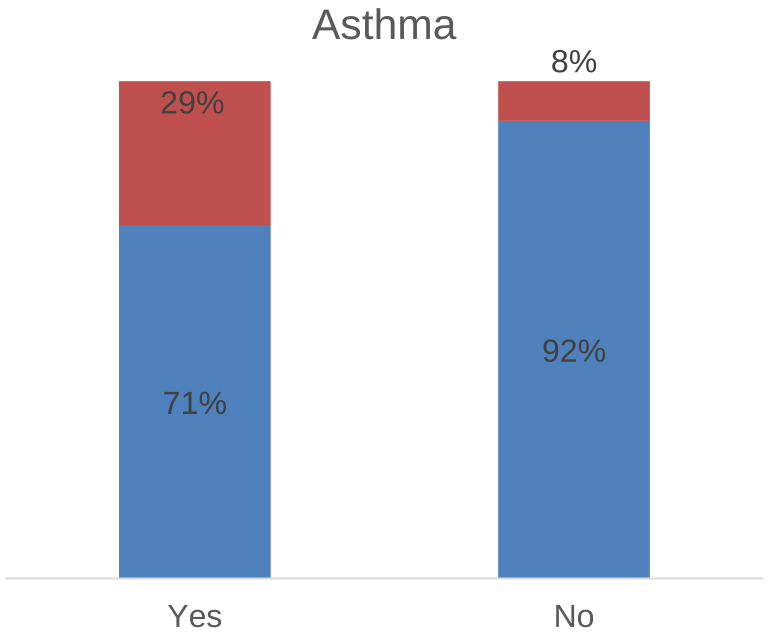



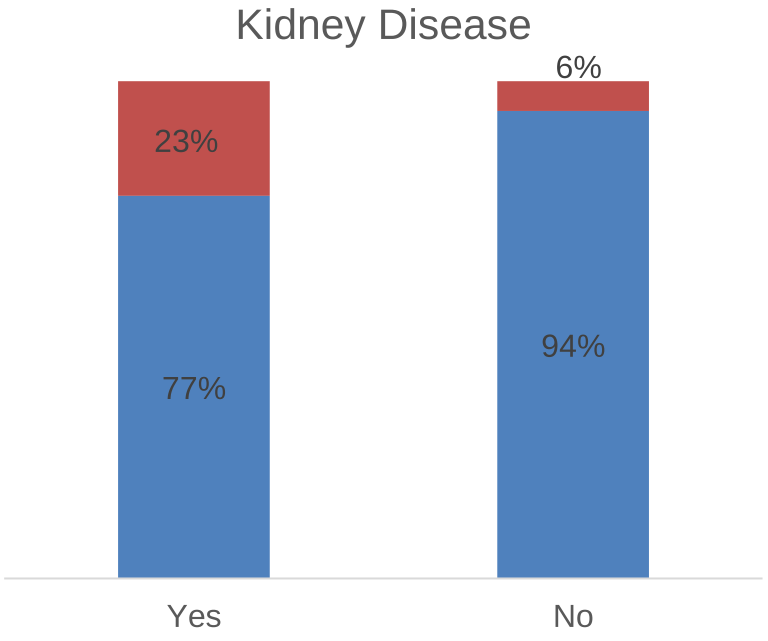
# Exploratory Data Analysis



# Exploratory Data Analysis



# Exploratory Data Analysis