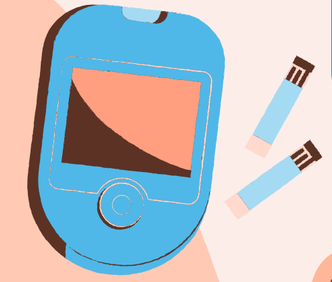
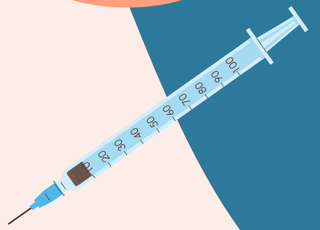
DIABETES DATA SET

By: Alejandro, Andrew, Janav





THE DS PROBLEM

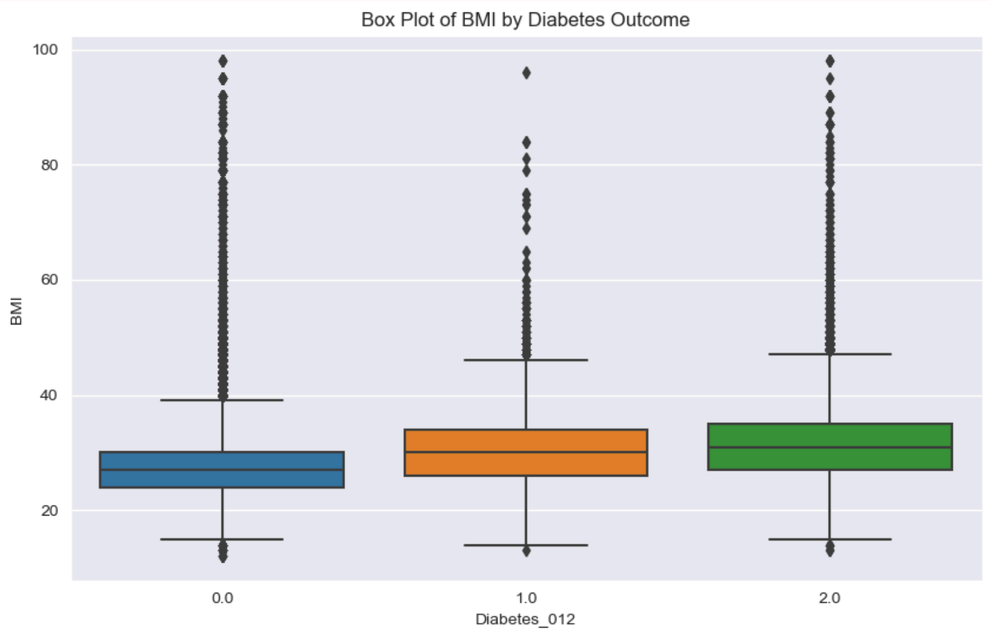
How can we create the best model, using the data and variables provided, to predict if someone has diabetes.

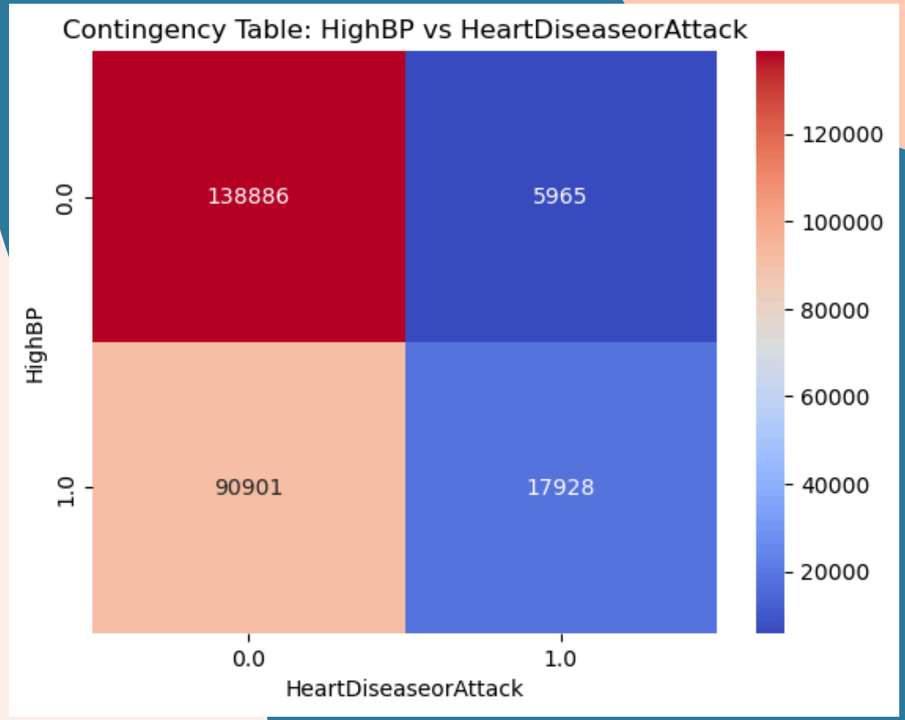
Data Cleaning Methods

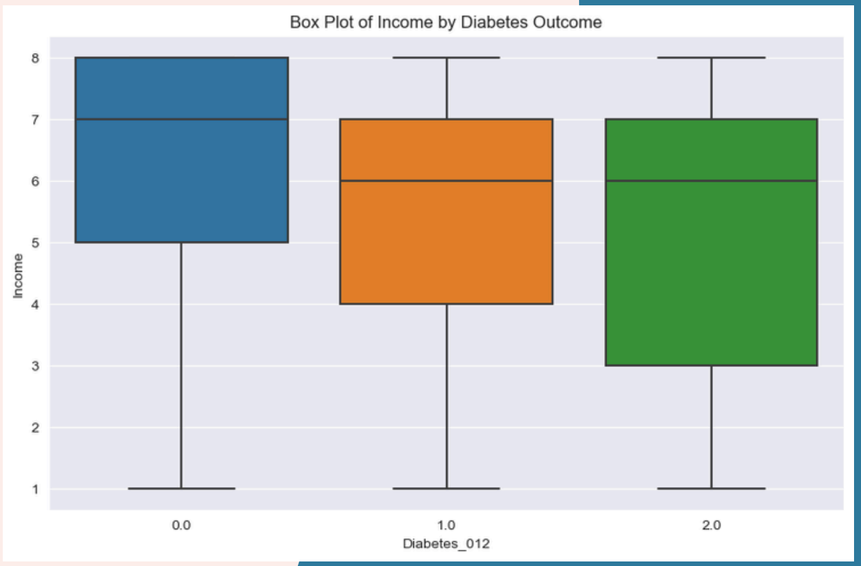
- Use .describe() to see potential min and max errors
- Convert data types to categoricals
- Check missing values .isnull().sum()
- Check general histogram for errors


Diabetes_012
HighBP
HighChol
CholCheck
BMI
Smoker
Stroke
HeartDiseaseorAttack
PhysActivity
Fruits
Veggies
HvyAlcoholConsump
AnyHealthcare
NoDocbcCost
GenHlth
MentHlth
PhysHlth
DiffWalk
Sex
Age
Education
Income

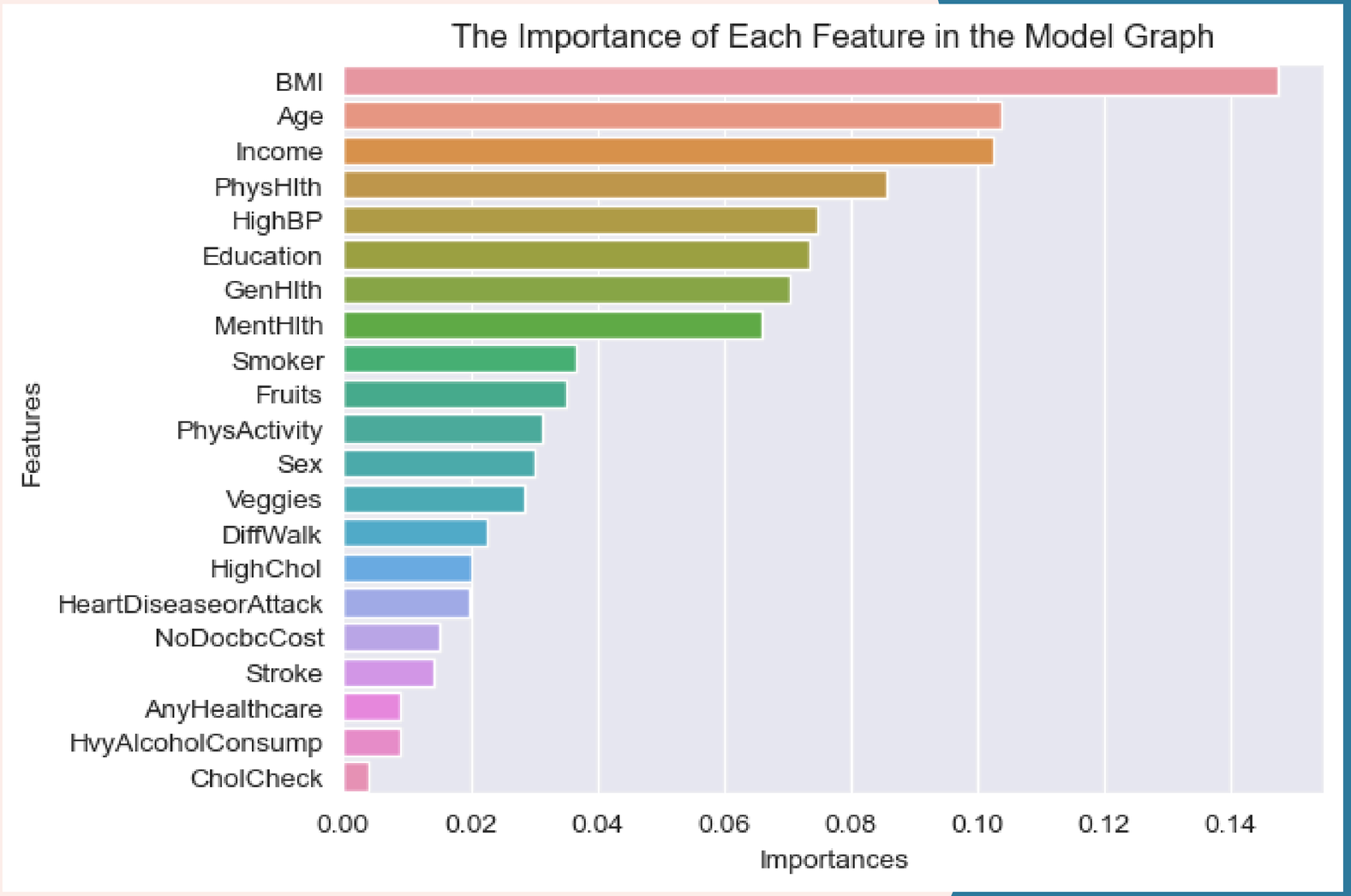
EDA GRAPHS


Box Plot of BMI by Diabetes Outcome
BMI
100
80
60
40
20
0.0
1.0
2.0
Diabetes_012

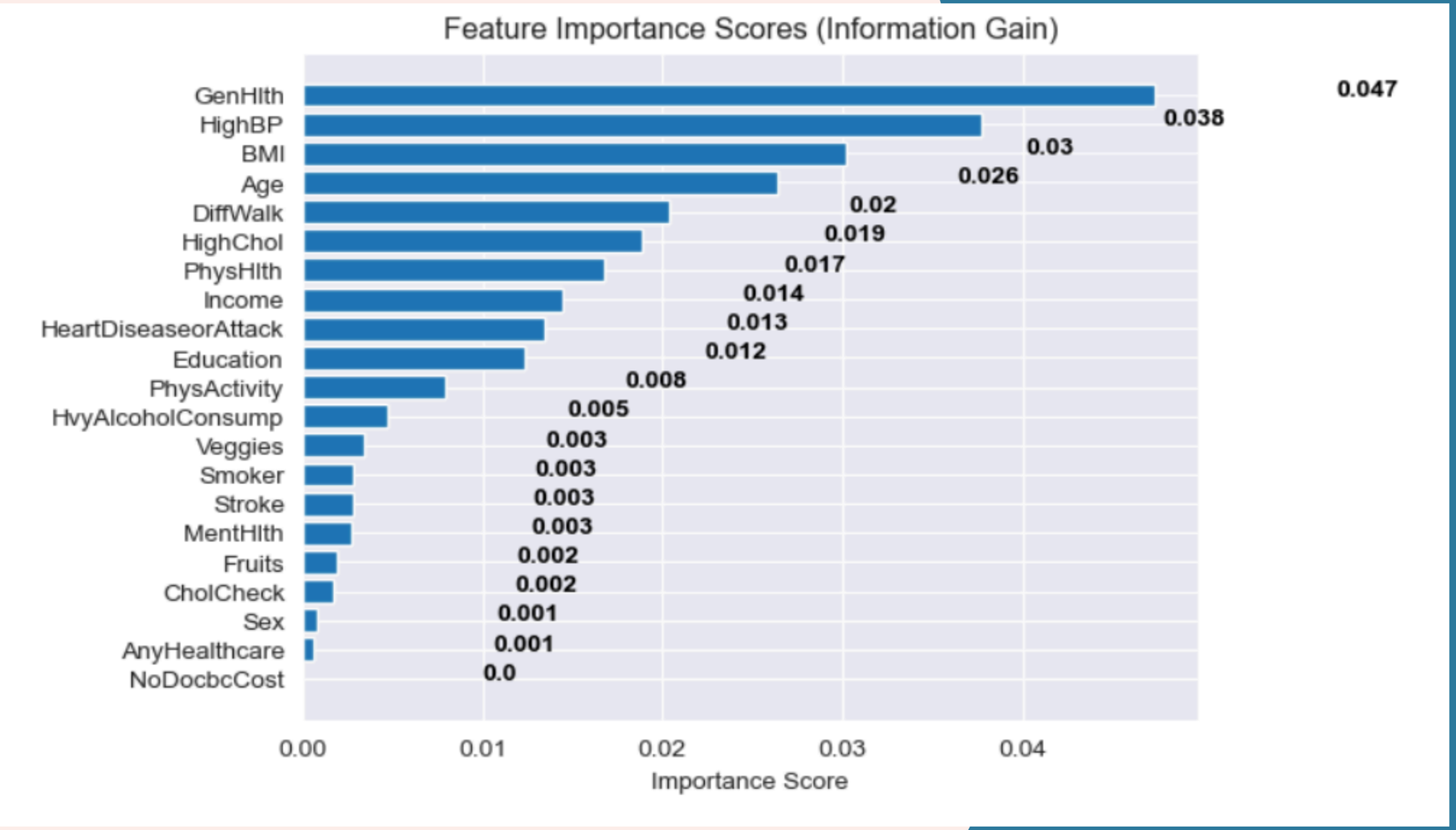

Contingency Table: HighBP vs HeartDiseaseorAttack
HighBP
0.0
1.0
138886
5965
90901
17928
0.0
1.0
HeartDiseaseorAttack
120000
100000
80000
60000
40000
20000


Box Plot of Income by Diabetes Outcome
Income
8
7
6
5
4
3
2
1
0.0
1.0
2.0
Diabetes_012

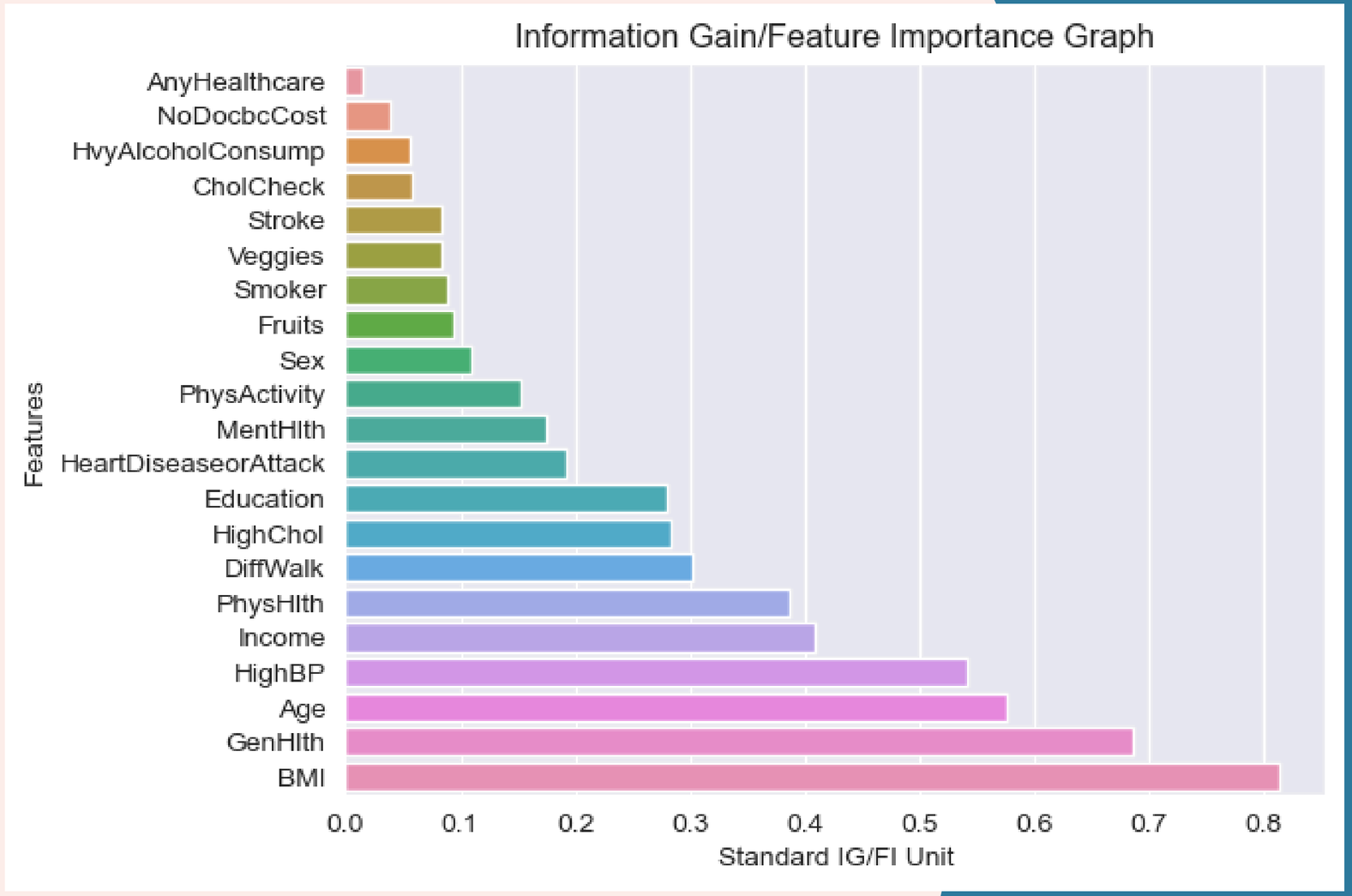
FEATURE IMPORTANCE (RANDOM FOREST)



INFORMATION GAIN



STANDARDIZED AVERAGE OF INFORMATION GAIN AND FEATURE IMPORTANCE



VARIABLES WE CHOSE

Top 9 features in Information Gain/Feature Importance graph
(We chose 9 because there was a fall off after variable 9)

(BMI, GenHlth, Age, HighBP, Income, PhysHlth, highchol, diffwalk, education)

Keeps strong correlated features while also factoring useful information in a classification. Reduces dimensionality, which will reduce computing time

CHI-SQUARED SIGNIFICANCE TESTING

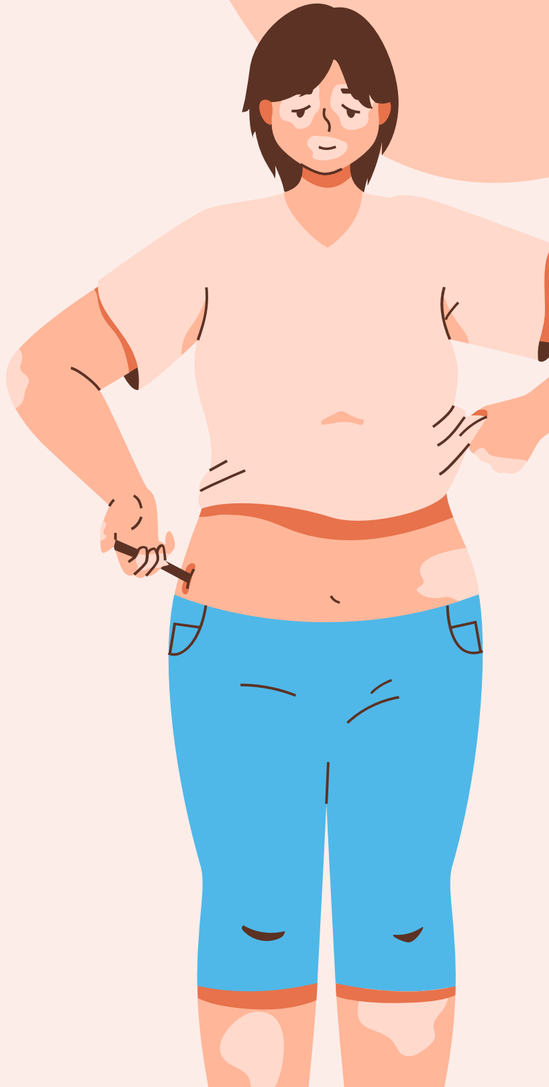
significance level = .05

-Sex and Diabetes_012, rejected null hypothesis

p = 3.38e-55

-AnyHealthcare and Diabetes_012, rejected null hypothesis

p = 1.00e-15



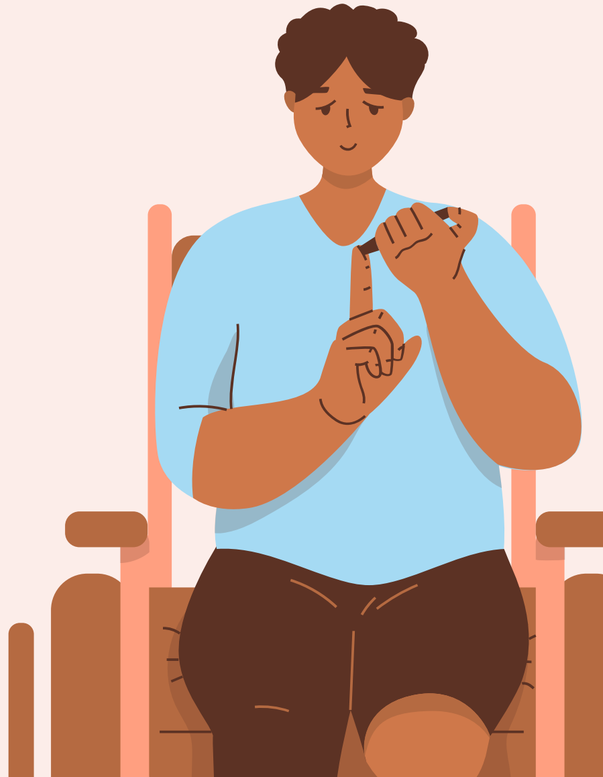
THESE ARE DEPENDENT VARIABLES!

MODELS

Models

We used three models that have the ability to handle non-linear data:

1. KNN Model
2. SVC Model
3. Decision Tree Model



Dataset

- We used the best features based on p values. The best ones were encoded and normalized for use in the model.
- get_dummies( ) was used for the encoding.
- StandardScaler( ) was used to normalize.
- We then split the data into X and y variables.

Results

KNN Model

Test Set Accuracy (before tuning): 80%

Best Parameters: {'leaf_size': 1, 'n_neighbors': 1}

Test Set Accuracy (after tuning): 83.7%

SVC Model

Test Set Accuracy (before tuning): 84%

Best Parameters: {'C': 3, 'kernel': 'rbf'}

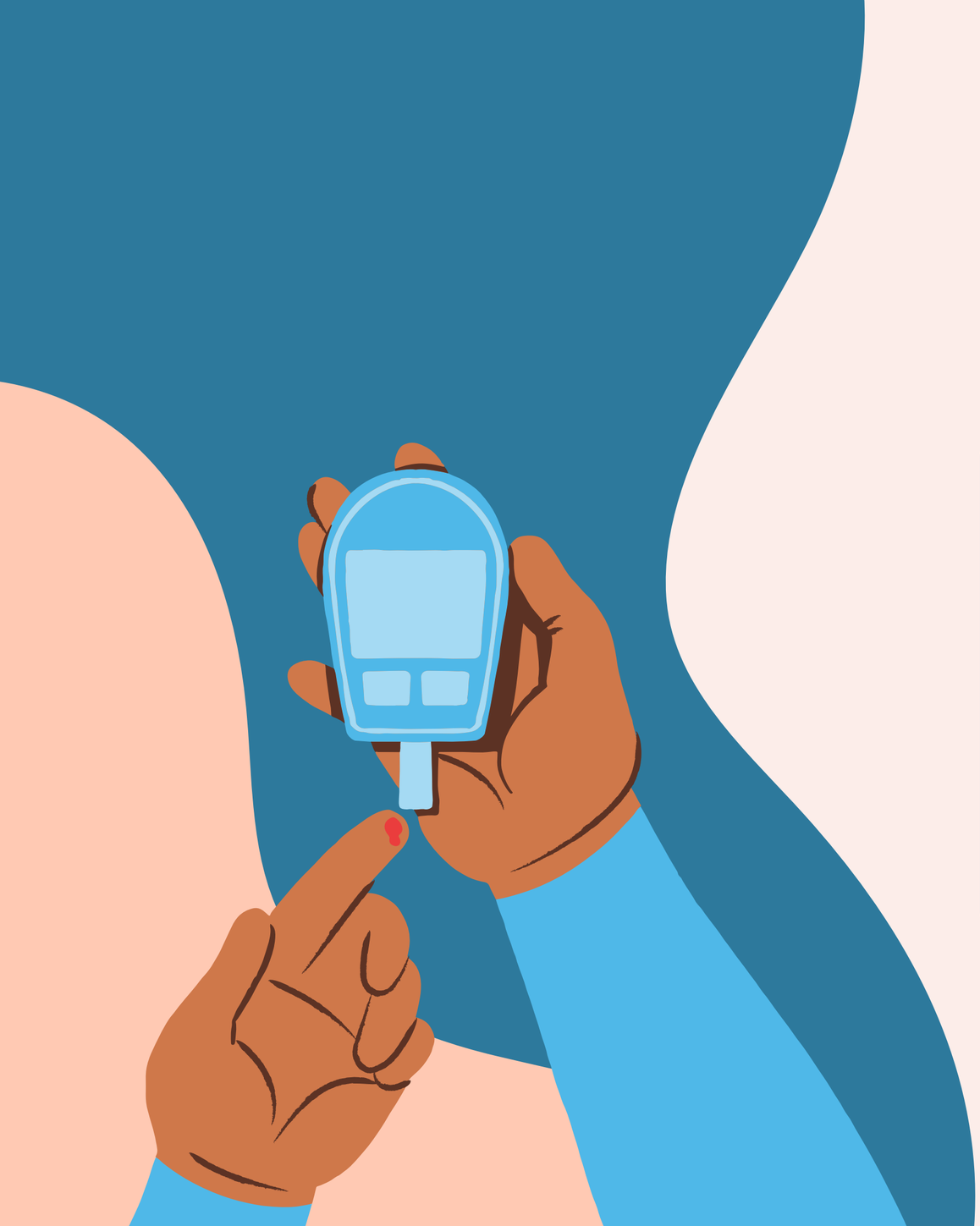
Test Set Accuracy (after tuning): 84.4%

Decision Tree Model

Test Set Accuracy (before tuning): 55%

Best Parameters: {'min_samples_split': 2, 'min_samples_leaf': 3, 'max_depth': 5, 'criterion': 'gini'}

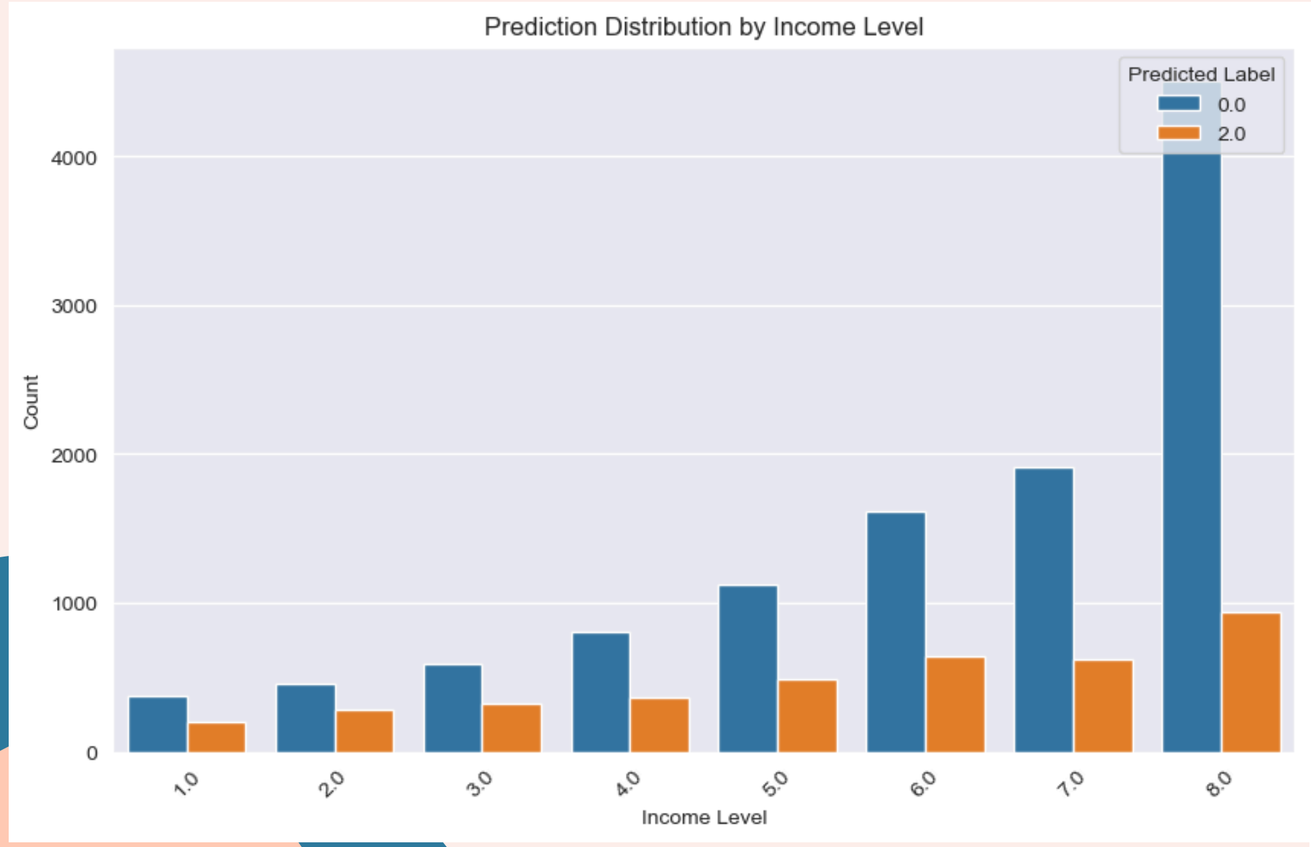
Test Set Accuracy (after tuning): 75%

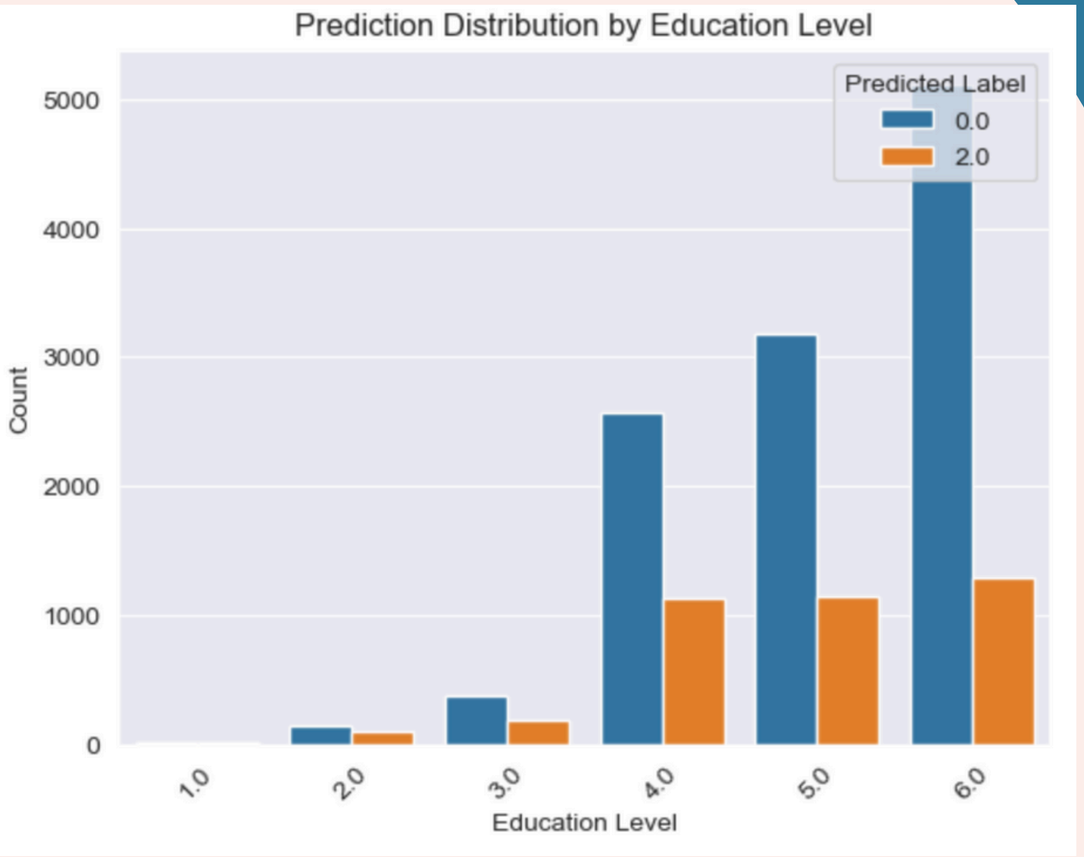


BEST MODEL

Upon training, evaluating and tuning all three models, we've concluded that the SVC model is the best one with an accuracy of 84.4%

BIAS TESTING





THANK YOU!

Thank you so much for watching our presentation! Do you have any questions, comments, or suggestions?