Diabetes is a common health problem with serious complications if not appropriately diagnosed and managed. The purpose of this project is to take health data with a number of identified risk factors and to predict with accuracy if a person might have diabetes or is at risk of diabetes. We would also like to identify what risk factors are the most predictive of diabetic risk.

Data

After investigation, we have identified BRFSS 2015 data as a candidate for use as our data to create our predictive model. The dataset is from Kaggle. This project will be using only the diabetes_binary_5050split_health_indicators_BRFSS2015.csv. This is a clean dataset of 70692 survey responses to CDC's BRFSS2015. It has a 50-50 split of respondents with no diabetes and with either prediabetes or diabetes. The target variable Diabetes_binary has 2 classes. Zero is for no diabetes and one is for prediabetes or diabetes. It has twenty-two feature variables and is a balanced dataset.

Based on the diabetes disease research regarding factors influencing diabetes disease and other chronic health conditions, only select features are included in this analysis.

Research in the field has found the following as important risk factors for diabetes and other chronic illnesses like heart disease. The following are the selected subset of features from BRFSS 2015.

- diabetes binary (0 = no diabetes; 1 = prediabetes/diabetes)
- high blood pressure (0 = no high BP; 1 = high BP)
- high cholesterol (0 = no high cholesterol; 1 = high cholesterol)
- cholesterol check (0 = no cholesterol check in 5 years; 1 = yes cholesterol check in 5 years)
- BMI (body mass index)
- smoker (smoke at least 100 cigarettes in entire life; 5 packs = 100 cigarettes; 0 = no; 1 = yes)

- stroke (ever told had a stroke; 0 = no; 1 = yes)
- heart disease or attack (coronary heart disease (CHD) or myocardial infarction (MI); 0 = no; 1 = yes)
- physical activity (in past 30 days not including job; 0 = no; 1 = yes))
- fruits (consumed 1 or more times per day; 0 = no; 1 = yes)
- veggies (consumed 1 or more times per day; 0 = no; 1 = yes)
- heavy alcohol consumption (adult men =>14 drinks per week; adult women =>7 drinks per week; 0 = no; 1 = yes)
- any healthcare (any kind of healthcare coverage including health insurance, prepaid plans (HMO, etc.); 0 = no; 1 = yes)
- no doctor because of cost (past 12 months when needed to see doctor but could not because of cost; 0 = no; 1 = yes)
- general health (general health 5 scale; 1 = excellent; 2 = very good; 3 = good; 4 = fair; 5 = poor)
- mental health (days of poor mental health 1-30 days)
- physical health (physical illness or injury days in past 30 days scale of 1-30)
- difficulty walking (have serious difficulty walking or climbing stairs; 0 = no; 1 = yes)
- sex (0 = female; 1 = male)
- age (14 level age category; 1 = 18-24; 2 = 25-29; 3 = 30-34; 4 = 35-39; 5 = 40-44; 6 = 45-49; 7 = 50-54; 8 = 55-59; 9 = 60-64; 10 = 65-69; 11 = 70-74; 12 = 75-79; 13 = 80 or>; 14 = not know/not sure/refused/missing)
- education (scale 1-6, 9); 1 = never attended school or only kindergarten; 2 = elementary; 3 = some high school; 4 = high school graduate; 5 = some college; 6 = college graduate or more; 9 = refused)
- income (scale 1-8; 1 = <10K; 2 = <15K; 3 = <20K; 4 = <25K; 5 = <35K; 6 = <50K; 7 = <75K; 8 = >75K or more; 77 = not know/not sure; 99 = refused)

The selected subset of features (columns from the dataset) from BRFSS 2015 was further investigated for more information in order to understand the meaning of the data gathered from the BRFSS 2015 codebook from the survey.

Additional information about the BRFSS 2015 codebook and relevant paper are as follows:

- BRFSS 2015 Codebook
- Relevant Research Paper using BRFSS for Diabetes ML
- Kaggle Dataset

Data Cleaning

Data Cleaning Jupyter Notebook

The BRFSS2015 codebook was reviewed to understand the dataset in depth. We checked the column names, shape, info, summary statistics for each of the columns and the unique values of the dataset. The data was checked for missing values, NaN values and duplicates. The data did not have any missing values, NaN's or any duplicates.

Looked into the value counts for each feature to understand the dataset and made a histogram for the data to see the distribution of features and see any useful values that may need to be investigated further. The value counts for each of the features gave a picture on how many people were diabetic or

not, how many male or female, how many had high BP or high cholesterol and so on which may be of use in later analysis.

Also looked at the mean, median and mode of the data to see if it showed anything useful in the categorical and non-categorical features of the data. The mode of the data is of interest in that most of the features will help in analyzing which features would be good predictors for diabetic risk. Lastly, we grouped the data by Diabetes_binary to further see if the features will be of use in predicting risk factors for diabetes. The data showed some association with the different risk predictors for diabetes, but nothing conclusive.

EDA

EDA Jupyter Notebook

We assessed the dataset's statistical summary and followed with visualizations of our features in histograms. We saw most of the data are binary and are categorical. In addition, we generated a Pandas profiling report which showed us no additional information, because most of our data are categorical and would need a different method to represent the data relationships.

Other visualizations like the boxplot and heatmap were created and only produced more questions regarding feature interactions. Correlations of binary and categorical data are not valid and performing a correlation would not provide us with any useful information at this time.

We explored the relationship between each feature and Diabetes_binary. We also simplified the categorical features (PhysHlth and MentHlth) to six categories instead of the original thirty one. The BMI data undergone min/max scaling to make it consistent with the categorical data to prevent it from dominating the future models. We also performed one hot encoding on the categorical data.

Hypothesis testing was performed to decide if there is a difference between diabetics and non-diabetics. We used the two-sided test, z-test and p-value calculation and found that most of the features are significantly different between the diabetics and non-diabetics which would need more investigation. We also performed Permutation test on each of the features and the results support our z-test results.

We also performed z-test and p-value calculation for the BMI and results showed that the non-diabetics have lower BMI values than the diabetics. This may indicate that BMI could be a predictor of diabetes.

It seems that almost all the features would still likely have some impact in the modeling based on the relationships between the features and the response variable.

Preprocessing

Preprocessing Jupyter Notebook

We changed BMI scaling of the data using MinMaxScaler, instead of the calculation used in the prior section. We changed to MinMaxScaler as a more robust scaler. Then we fitted and transformed the feature in preparation for splitting the dataset. OneHotEncoding was already performed in the EDA to allow for hypothesis testing.

We chose the Diabetes_binary as our dependent variable. Train test split method was applied to develop the training and testing data. We used the 80/20 split in the model development dataset.

Algorithms and Machine Learning

Modeling Jupyter Notebook

In the diabetes dataset, we want to predict the accuracy if a person might have diabetes or is at risk of diabetes by finding out what risk factors are the most predictive of diabetic risk

The dataset is a binary classification. We are interested in the ACCURACY which is the proportion of the total number of correct predictions that were correct and RECALL measurement, because we would like a high positive conclusion even if it gives us a large number of false positives. A high false positive rate would indicate that we consider someone who is a diabetic or who isn't. This can be resolved by additional medical testing. We prefer a high false positive rate as opposed to high false negative rate, due to the problems of not being diagnosed as at risk for diabetes.

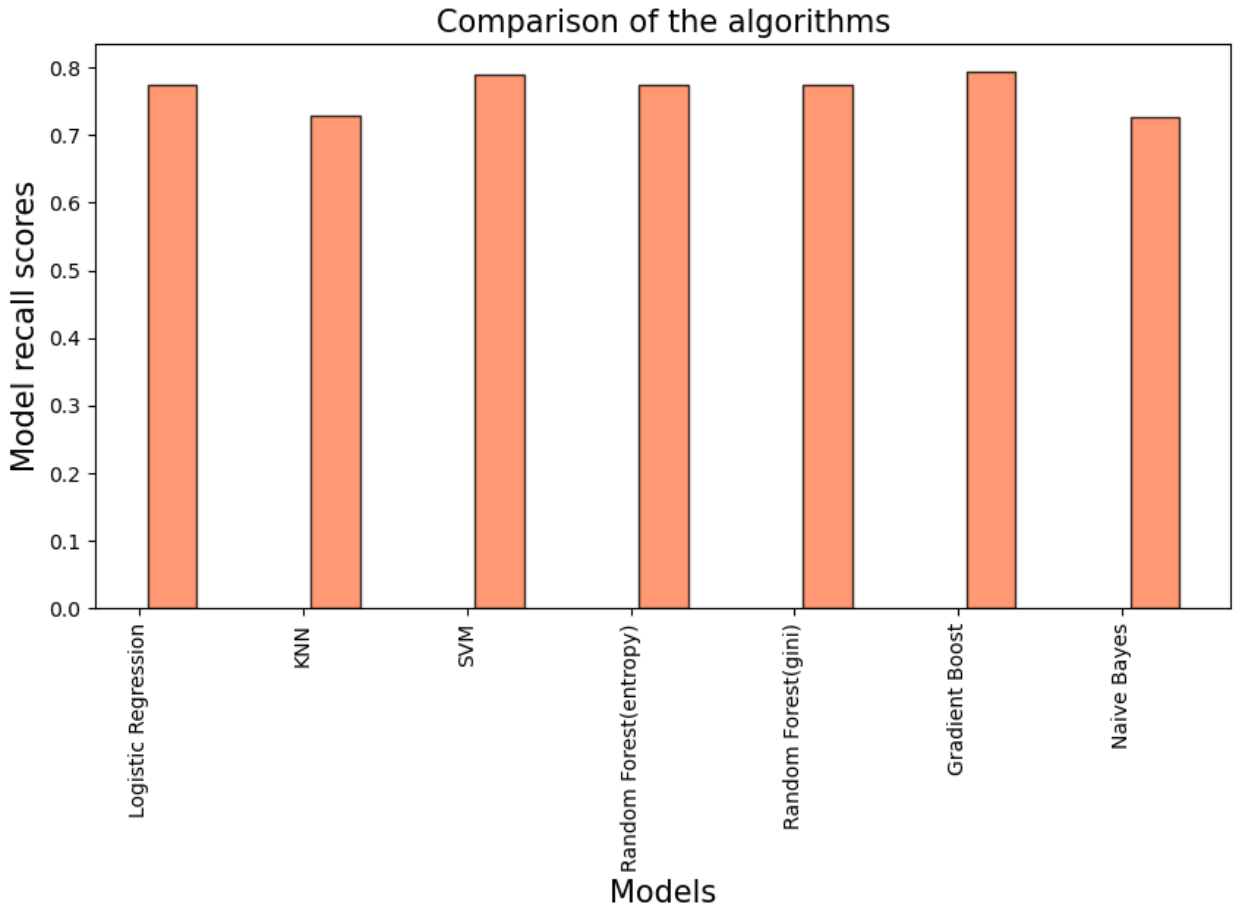
Here are the following classification models we will be using:

- Logistic Regression
- K-Nearest Neighbor (KNN)
- Gradient Boost
- Support vector machine (SVM)
- Random Forest (entropy and gini)
- Naive Bayes

Comparison of the models:

This shows that **Gradient Boost** and **SVM** are the top two best performing models. Both models are ensemble and based on decision trees. We will perform a gridsearchCV for hyperparameter tuning for the two ML models.

Algorithm	Accuracy score	Recall score	F1 score
Gradient Boost	0.751821	0.794872	0.762793
SVM	0.749204	0.788673	0.759463
Logistic Regression	0.748568	0.774725	0.755720
Random Forest(gini)	0.734847	0.774444	0.745710
Random Forest(entropy)	0.732018	0.773457	0.743449
KNN	0.704788	0.728092	0.712336
Naive Bayes	0.708607	0.726965	0.714681



Hyperparameter Tuning for Gradient Boost

After hyperparameter tuning with gridsearchCV, the following results were found:

```
Fitting 3 folds for each of 81 candidates, totalling 243 fits
GradientBoostingClassifier(learning_rate=0.04, max_depth=9,
max_leaf_nodes=90, n_estimators=140, random_state=5, subsample=0.8)
GridsearchCV best score: 0.7943217218918154
```

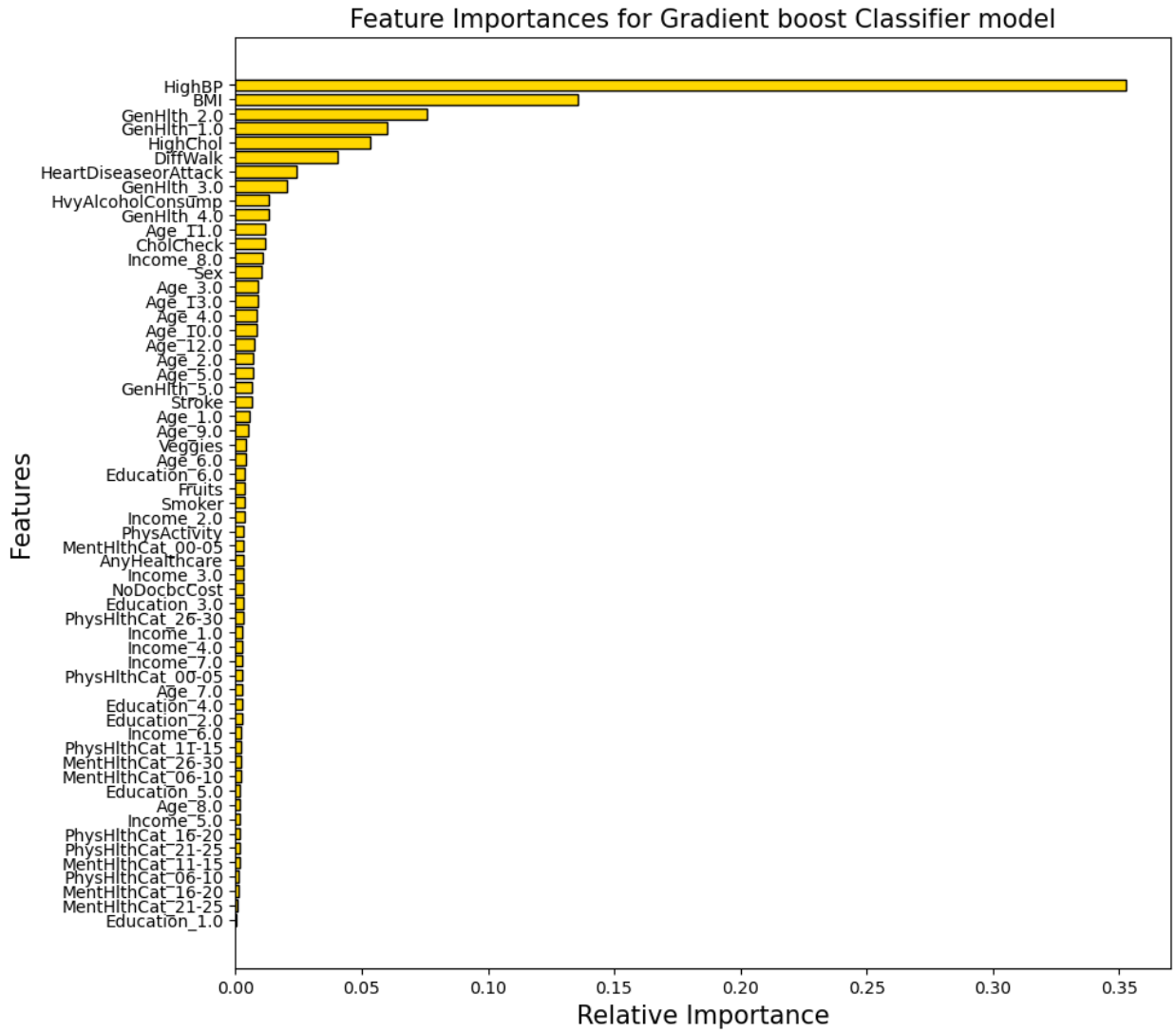
Our new results after fitting the Gradient Boost model with the optimal hyperparameters:

```
Accuracy: 0.7518919301223566
  Recall: 0.7945900253592562
      F1: 0.7627806329456316
```

Compared with the original scores, recall went down just a little bit.

Next, we calculated the important features and found:

Features	Importance scores
HighBP	0.352958
BMI	0.135629
GenHlth_2.0	0.075922
GenHlth_1.0	0.060004
HighChol	0.053260



Based on the table and chart (above), the top five features that are predictive of having a risk of diabetes are High BP, BMI, General Health 2.0(very good), General Health 1.0(excellent) and High cholesterol.

Hyperparameter Tuning for SVM:

After hyperparameter tuning with gridsearchCV, the following results were found:

```
Best Estimator Results
{'`C': 100, 'gamma': 0.001, 'kernel': 'rbf'}
SVC(C=100, gamma=0.001)
```

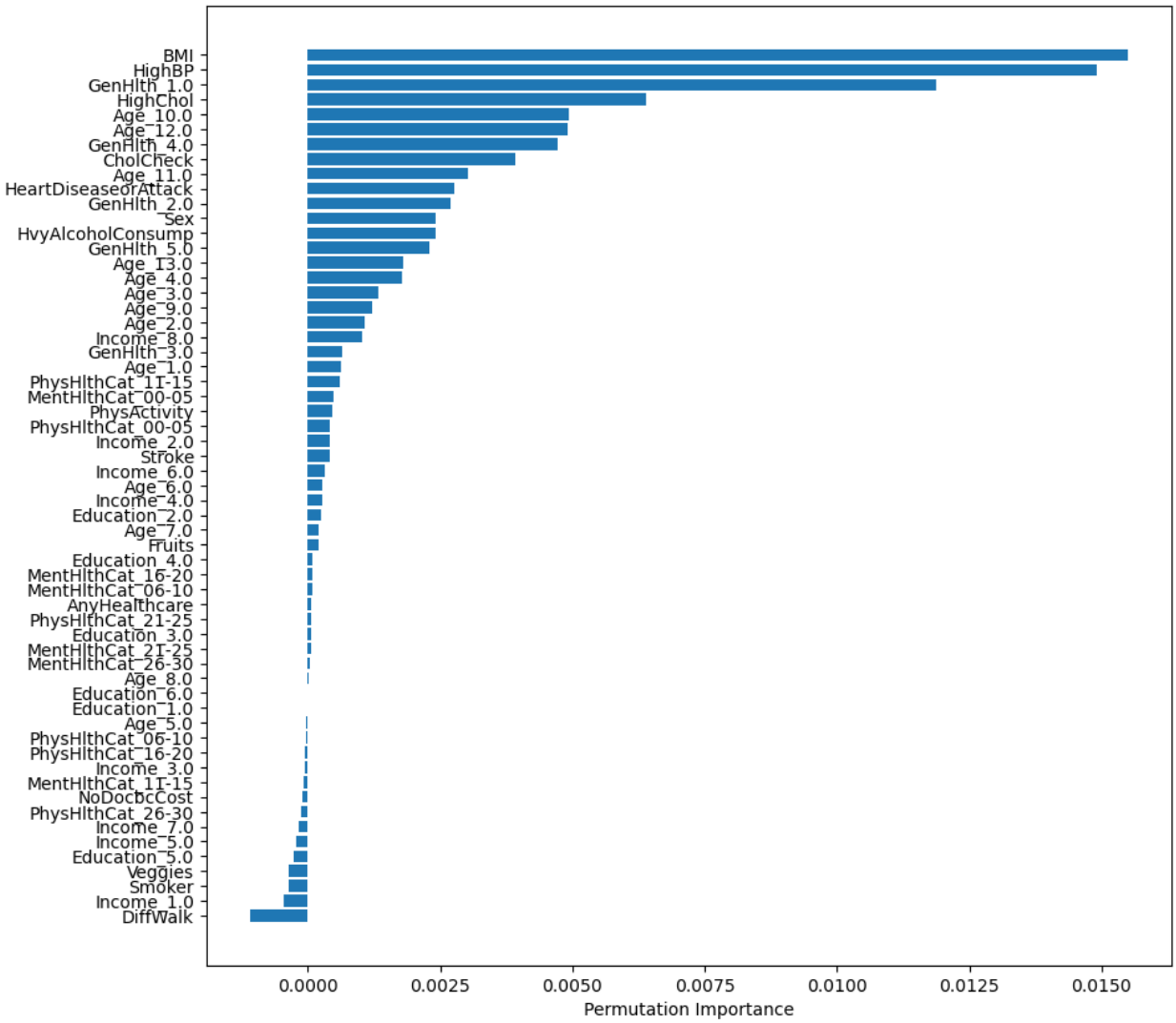
Our new results after fitting the SVM model with the optimal hyperparameters:

```
Accuracy: 0.7477898012589292
  Recall: 0.7945900253592562
      F1: 0.7598006196955409
```

Compared to the original scores, recall improved a little bit.

Next, we calculated the important features and found:

Features	Importance scores
BMI	0.015489
HighBP	0.014909
GenHlth_1.0	0.011882
HighChol	0.006394
Age_10.0	0.004937



Based on the table above, the top five features that are predictive of having a risk of diabetes are BMI, High BP, General Health 1.0, High cholesterol, General Health 4.0.

Final Modeling Results

After hyperparameter tuning the two best models, the final and the best model for this project is the Gradient Boost Model. It had better accuracy and F1 score than the SVM Model.

The top five features of the Gradient Boost Model that are predictive of having a risk of diabetes are High BP, BMI, General Health 2.0(very good), General Health 1.0(excellent) and High cholesterol.

Based on the series of steps done to arrive at this conclusion, there is still room for improvement. We found a model that has the best result to make the prediction of finding out what risk factors are the most predictive of diabetic risk and gave us the top 3-5 risk factors of high interest.

Future Investigation

There is always room for improving the model. We can consider the following for improvements:

1. Trying more feature engineering to extract more relevant data is a big possibility.
2. Using models that have less computational complexity and less maintenance cost is something to be highly considered as well.
3. Use randomsearchCV, instead of gridsearchCV. This would allow us to search a larger parameter space, while still yielding very good parameters.
4. Use a hybrid approach, starting with randomsearchCV to search a larger space to narrow down the optimal parameters, then use gridsearchCV to refine the parameters further.