Table of Contents

Introduction

As one of the most prevalent chronic diseases in the United States, diabetes has widely among people with various aging groups, even the young kids. Diabetes not only puts a heavy burden on the shoulders of patients and their family, impacts life quality, but also on the national economy that it costs about three hundred thousand dollars annually. The majority of prediabetics, according to the CDC, are ignorant of the risk and might pass up the ideal chance for early intervention. Early diabetes detection and classification is essential for improving the condition and implementing effective treatment.

Problem statement

- Predict if someone has diabetes using survey questions from the BRFSS.
- Find the risk factors that most accurately predict the risk of diabetes.

Dataset Selection

Data Description

We choose to analyze with publicly available survey data of BRFSS in 2022 which consist of 445132 records among 328 variables (https://www.cdc.gov/brfss/annual_data/annual_2022.html). While most of the features in the dataset are associated with survey background or health condition not related to diabetes, we are only using small segments of them.

Data Dictionary

The official code book for 2022 survey can be found in https://www.cdc.gov/brfss/annual_data/2022/zip/codebook22_llcp-v2-508.zip

We take relevant research paper findings as reference (https://www.cdc.gov/pcd/issues/2019/19_0109.htm) when selecting features.

Category	Variable	Description	Value
Response Variable	DIABETE4	Indicate Diabetes status of respondent	1: Yes, 2: Yes, but pregnant, 3: No, 4: Pre-diabetes
Demographic	_SEX	Indicate sex of respondent	1: Male, 2: Female
	_AGE5YR	13-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 older
	_EDUCAG	Level of education completed	1: Below high school, 2: High school, 3: Attend college, 4: College graduate
	_INCOMG1	Income categories	1: Less than 15,000, 2: 15,000-25,000, 3: 25,000-35,000, 4: 35,000-50,000, 5: 50,000-100,000, 6: 100,000-200,000, 7: 200,000 or more

General Health	GENHLTH	General health status	1: Excellent, 2: Very good, 3: Good, 4: Fair, 5: Poor
	MENTHLTH	Mental health status	Number of days mental health is not good in past 30 days as INT
	PHYSHLTH	Physical health status	Number of days physical health is not good in past 30 days as INT
BMI	_BMI5	Body Mass Index	BMI as INT
Smoke	_SMOKGRP	Smoking Group	1: Current smoker,20+/year, 2: Former smoker, 20+/year, 3: All smoker, 20-/year, 4: Never smoker
Alcohol Consumption	AVEDRNK3	Average drink amount in past month	Average number of alcoholic beverages consumed per week
Exercise	EXERANY2	Exercised in past month	1: Yes, 2: No
Chronic Health Conditions	_MICHD	Heart disease (CHD) or myocardial infarction (MI) status	1: Have, 2: Do not have
	_ASTHMS1	Asthma status	1: Current, 2: Former, 3: Never
	_DRDXAR2	Arthritis status	1: Diagnosed with arthritis 2: Not diagnosed with arthritis

Analysis

Descriptive statistics

We did a fundamental feature selection on the original dataset to get the columns shown above. The selected subset of data contains 15 columns and 445132 rows. Among these records, 20% of them are null value so we eliminated them. The second step we took to further deal with missing values is to drop the records that are shown as unknown or not applicable in the survey data since they cannot provide any information. After both steps, we have 291267 records remaining which is 65% of the original.

Among the remaining records, 83% of them are category as non-diabetes and only 17% had diabetes or prediabetes.

Two of the features we considered as continuous numerical variables which are BMI, body mass index, and total number of alcoholic beverages consumed per week. The range and mean of these two variables are 8560, 25000, and 2872, 327. We can conclude from the range and mean of average alcohol that there must be outlier on this feature, and we need to deal with it.

For those discrete variables, we can use label counts to gain information about the survey population. As the data shown, there are almost equal amounts of female and male participants, most of them are between age 65 and 69. People with bachelor or above education status and annual income between 50000-100000 make up the most part of the population. Less of them have other diseases like asthma, CHD, and arthritis than have no.

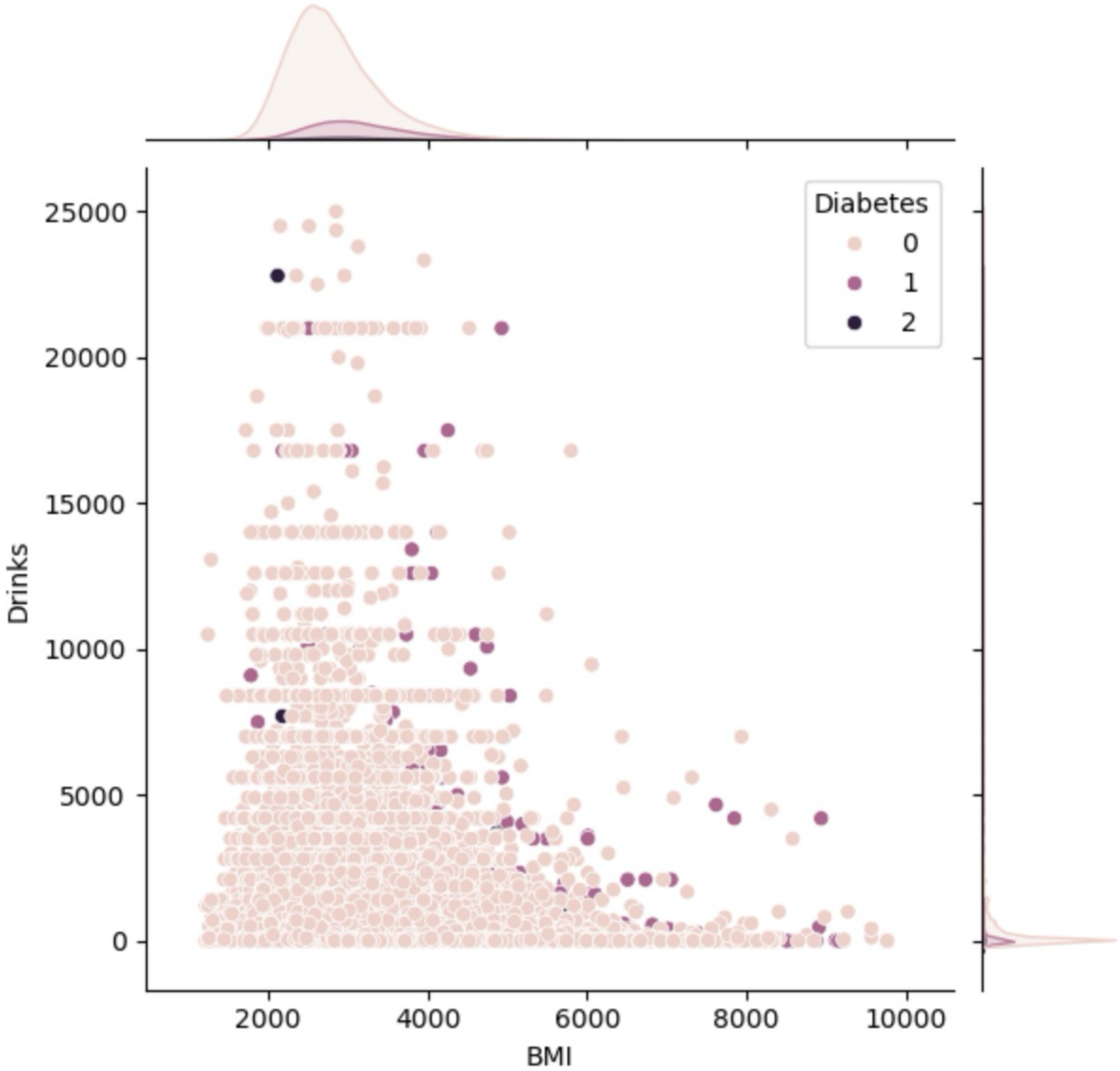
EDA (Data Visualization)

By creating visualization graph, we aim to do two things:

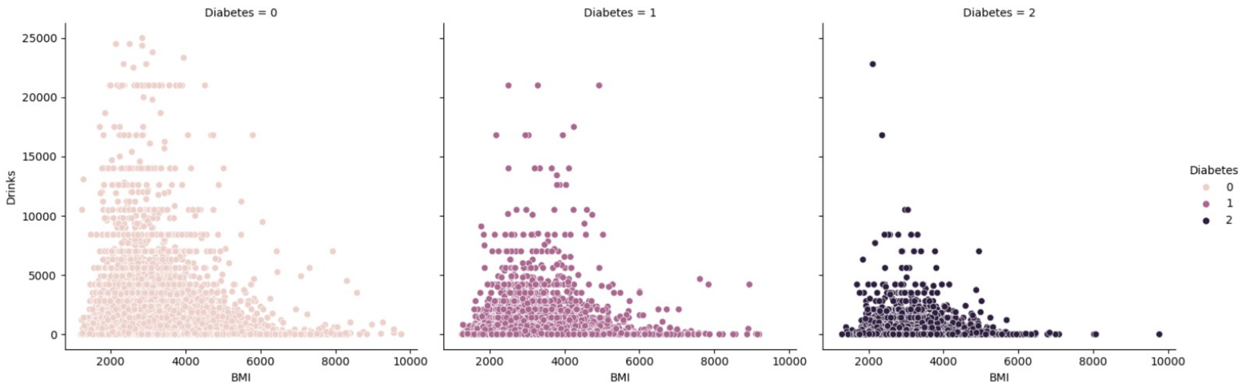
1) Explore the distribution of each feature across different diabetes conditions,
2) Extract the correlation between each feature and diabetes condition.

To achieve these goals, we used 5 different plots for continuous variables and discrete variables, scatter plot, histogram, violin plot, bar chart and heat map.

- Because there are only two numerical variables, we utilized a scatter plot to illustrate the relationship between them based on diabetes type.

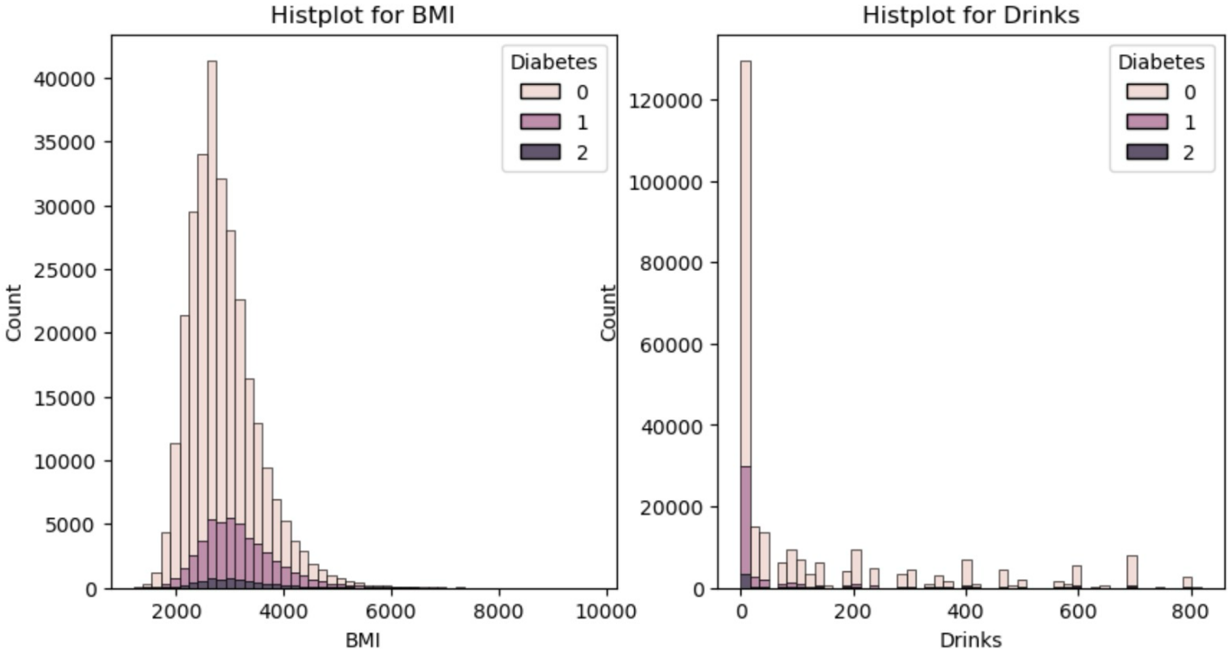


The graphic shows a nonlinear correlation between BMI and average alcohol intake. They fall along the y=1/x line, with most of the data indicating low BMI and alcohol usage. However, when we divide the sample by diabetes status and look at the scatter plot, there are little variations across the groups.

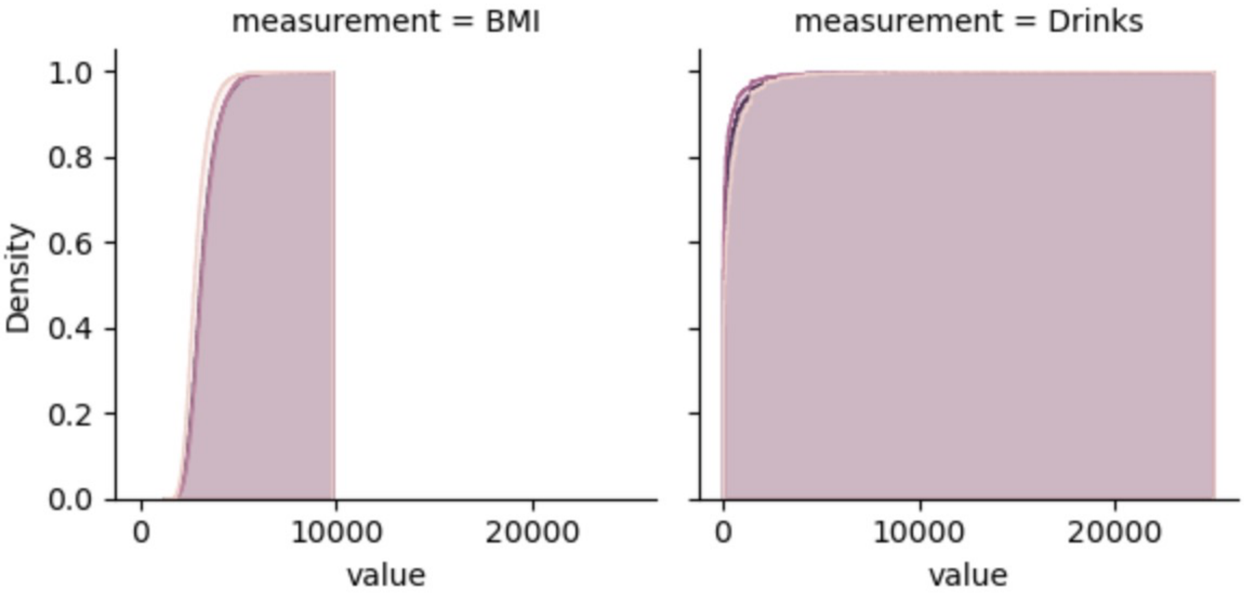


The no diabetes group have a more left leaned trend than the other two, indicating that Diabetes people tends to have a higher BMI value.

- To investigate the distribution of continuous variables, we created a histogram for BMI and average drinks.

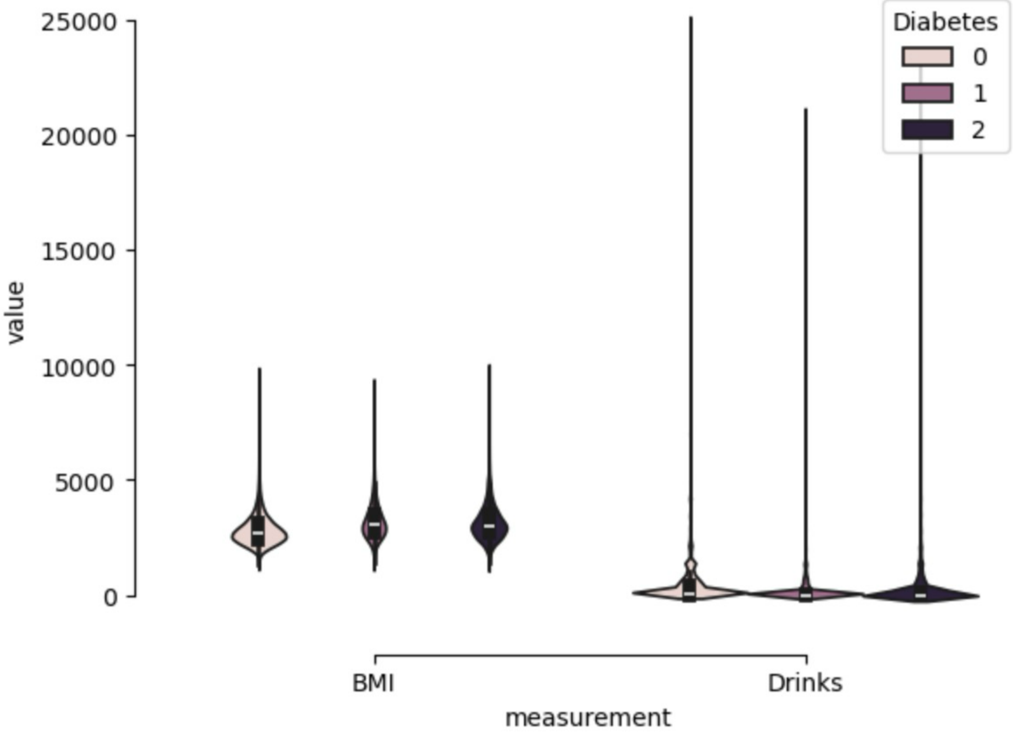


Because we discovered that the distribution is significantly left leaning, we plotted a cumulative distribution to have a better understanding of how they add up.



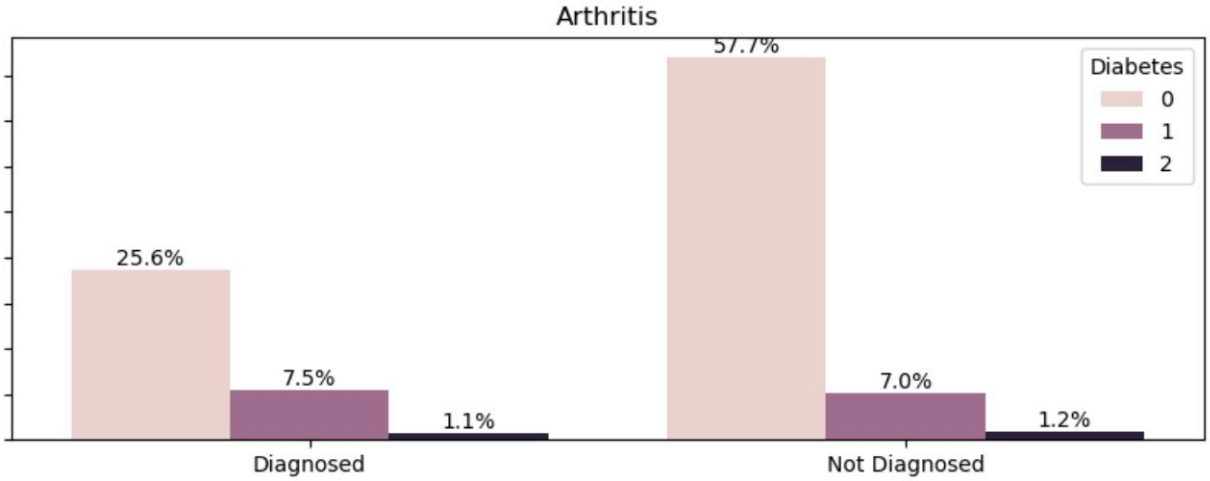
The result graph reveals that the distribution pattern for all three diabetes types is the same.

- The violin plot revealed the mean value and range of two numerical variables, indicating that people with diabetes or prediabetes had a higher average BMI value than those without. It is inverted for the average drink amount, since the diabetic group in general.



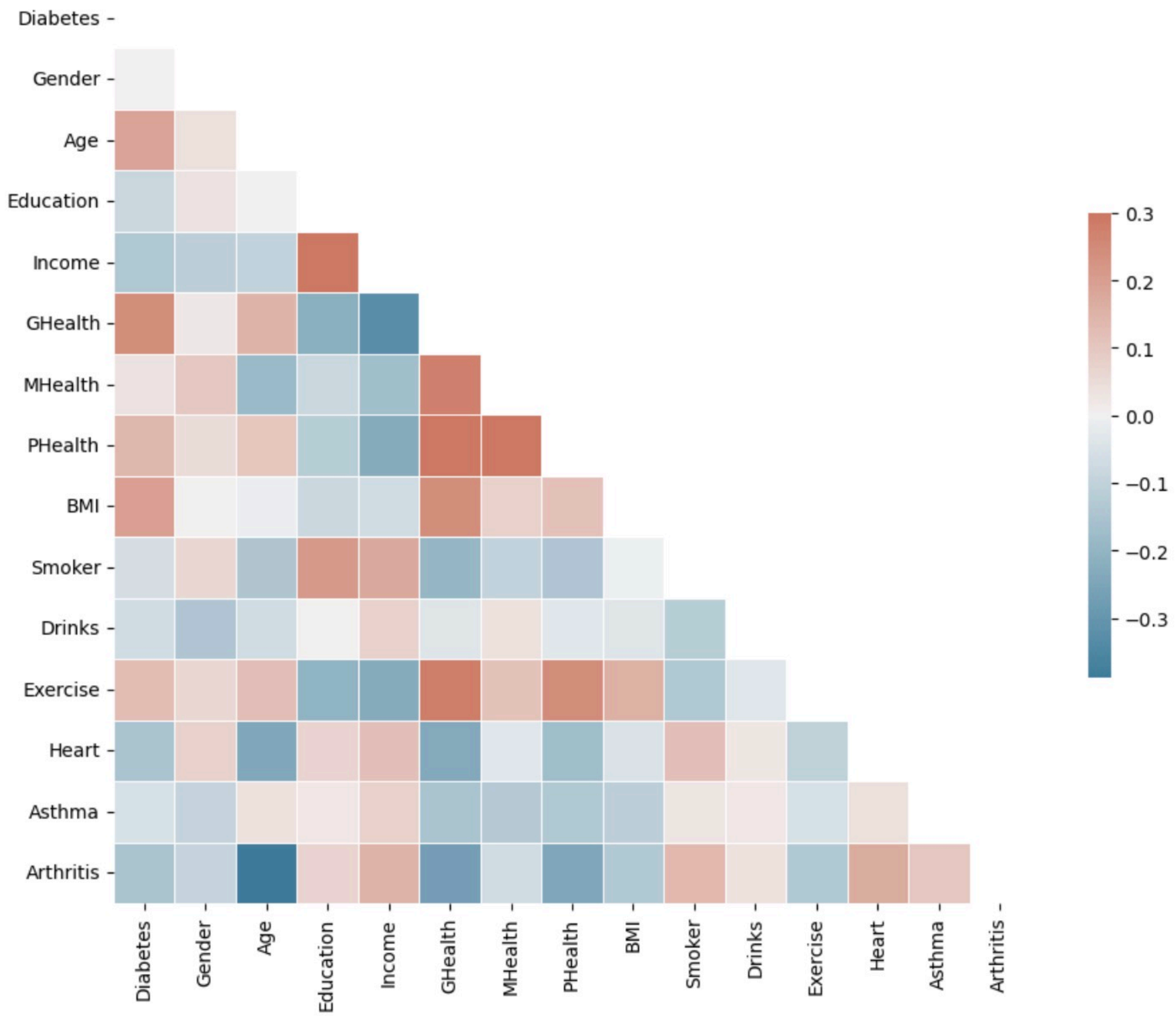
Recheck the distribution by histogram, cumulative plots, and violin plots, it seems like there is no relationship between BMI and Diabetes, or Drinks and Diabetes. The majority of these classes are all gathering at the low values of BMI or Drinks less.

- As the values for category variables are discrete, a histogram was not available. We developed two sets of bar charts to stand for separate distribution bases: the distribution of diabetes conditions across classes under each feature, and the distribution of class counts of each feature across diabetes conditions.



From all the plots we find out that Arthritis is unusual as it appears to have a very different distribution pattern between Diabetes and non-Diabetes. For those with Diabetes, we can see a balanced distribution between arthritis or not, while for non-diabetes there's a significant higher proportion on not diagnosed.

- To figure out the relationship between diabetic conditions and all other variables, we must first develop a correlation matrix from the preprocessed information and then produce a heat map to visually assess the correlation value.



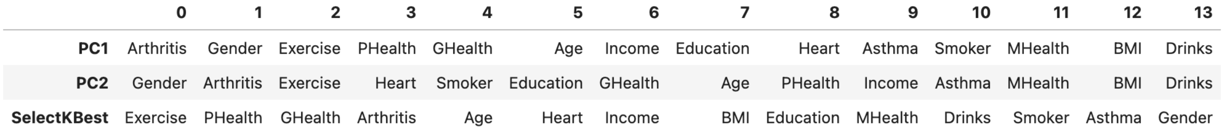
From the heatmap we look at the Diabetes column can find out some attributes that are highly correlated to the predict label. Self-evaluated general health attributes have the most positive correlation, meaning when people think they are health they tend to be health.

Feature engineering and selection

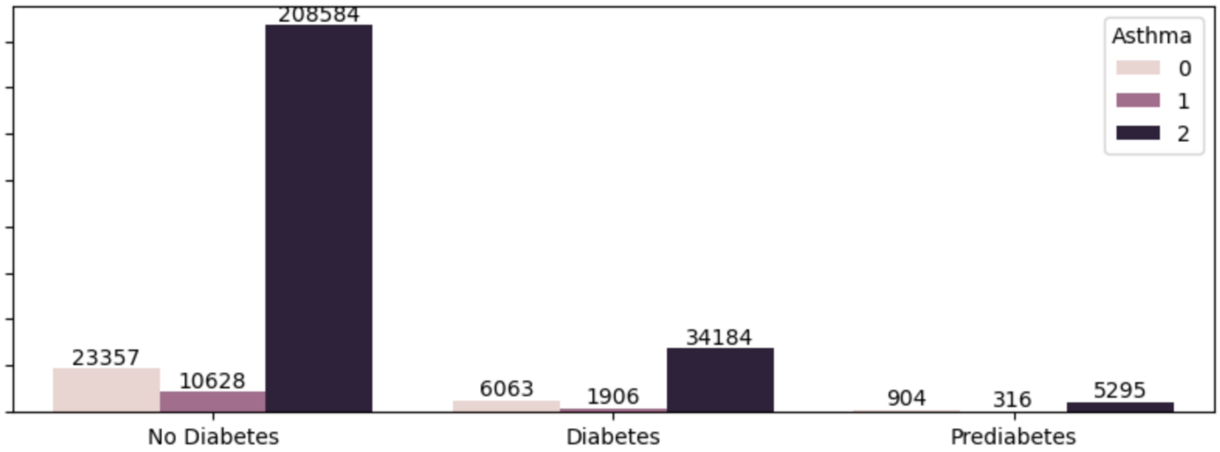
There are two steps of feature selection throughout the process. First, we chose the original unprocessed dataset of 16 features from 328 survey answer variables by referring to the published study report. The second step was carried out after examining chosen characteristics and selecting just those that are directly connected to the diabetes outcome with help of PCA and filter.

We used PCA analysis not to extract features, but to decide which factors contributed the most to the diabetes variable. Looking at the PC variable indices, we discover that BMI and number of drinks are the least associated variables, which supports the visualization's results. Other three non-related variables are mental health condition, smoking habit, and asthma history.

Another method used for feature selection is filter method. A subset of features is chosen by filter method in accordance with how they relate to the target variable, that PCA does not take into consideration. Selection is independent of any algorithm based on machine learning. Conversely, filter methods use statistical tests to find how "relevant" the features are to the output. The results show that Asthma and Gender are least related to the target variable, which also supports the visualization. Other three non-related variables are mental health condition, drinking and smoking habit.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13
PC1	Arthritis	Gender	Exercise	PHealth	GHealth	Age	Income	Education	Heart	Asthma	Smoker	MHealth	BMI	Drinks
PC2	Gender	Arthritis	Exercise	Heart	Smoker	Education	GHealth	Age	PHealth	Income	Asthma	MHealth	BMI	Drinks
SelectKBest	Exercise	PHealth	GHealth	Arthritis	Age	Heart	Income	BMI	Education	MHealth	Drinks	Smoker	Asthma	Gender



Recommended features

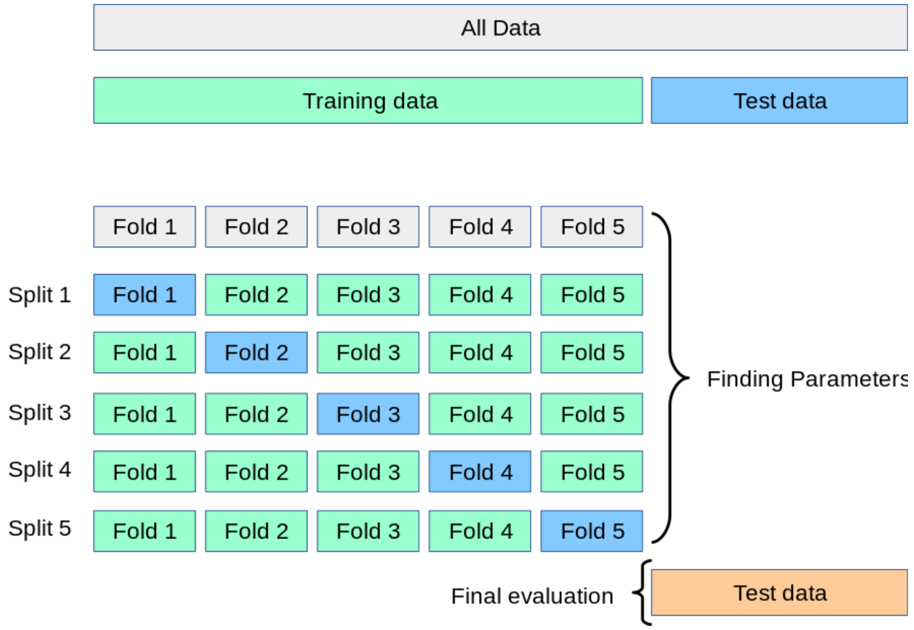
After applying PCA and Filter, we find a list of features that diabetes may depend on: **age, gender, education level, income level, exercise habit, self-recognized health status, heart disease and arthritis history.**

Modeling

In this part, we chose two models, Naïve Bayes, and Decision Tree that can solve nonlinear problems, for training and tuning their parameters by cross-validation (CV for short) which can solve the problem of overfitting.

The CV will loop for k times. During each iteration, the original training set will be divided into smaller sets, k-1 of which serve as training data, and the remaining fold is used as a validation set to calculate a performance score, such as accuracy. The cross_val_score function from scikit-learn is the simplest way to apply CV. The output of this function is a list of accuracy showing each iteration's model's performance on the kth fold (the validation set of each iteration). Thus, the output list length is k, and we will use the mean value as our output score.

And because of the splitting, a validation set is no need here, we will just use a training set for cross-validation and hyperparameter tuning, and a testing set for final evaluation of the model.



Naïve Bayes

Introduction

There are three main types of Naive Bayes model which aim for different attribute types.

- Gaussian Naive Bayes: This is used for continuous features (e.g., temperature, humidity). It assumes that each feature follows a Gaussian distribution and calculates the probability of each class based on the feature values.
- Multinomial Naive Bayes: This is used for categorical features (e.g., ham vs. spam emails). It assumes that each feature is independent and calculates the probability of each class based on the feature values.
- Bernoulli Naive Bayes: This is used for binary features (e.g., presence or absence of a word in a document). It is like the multinomial model but is more suitable for binary data.

Since our diabetes dataset holds a mixture of continuous, discrete, and binary attributes, we modeled all three types and chose the best performed one. In later optimization, we introduced a fourth Naive Bayes model specifically used for imbalanced dataset, called Complement Naive Bayes.

Implementation

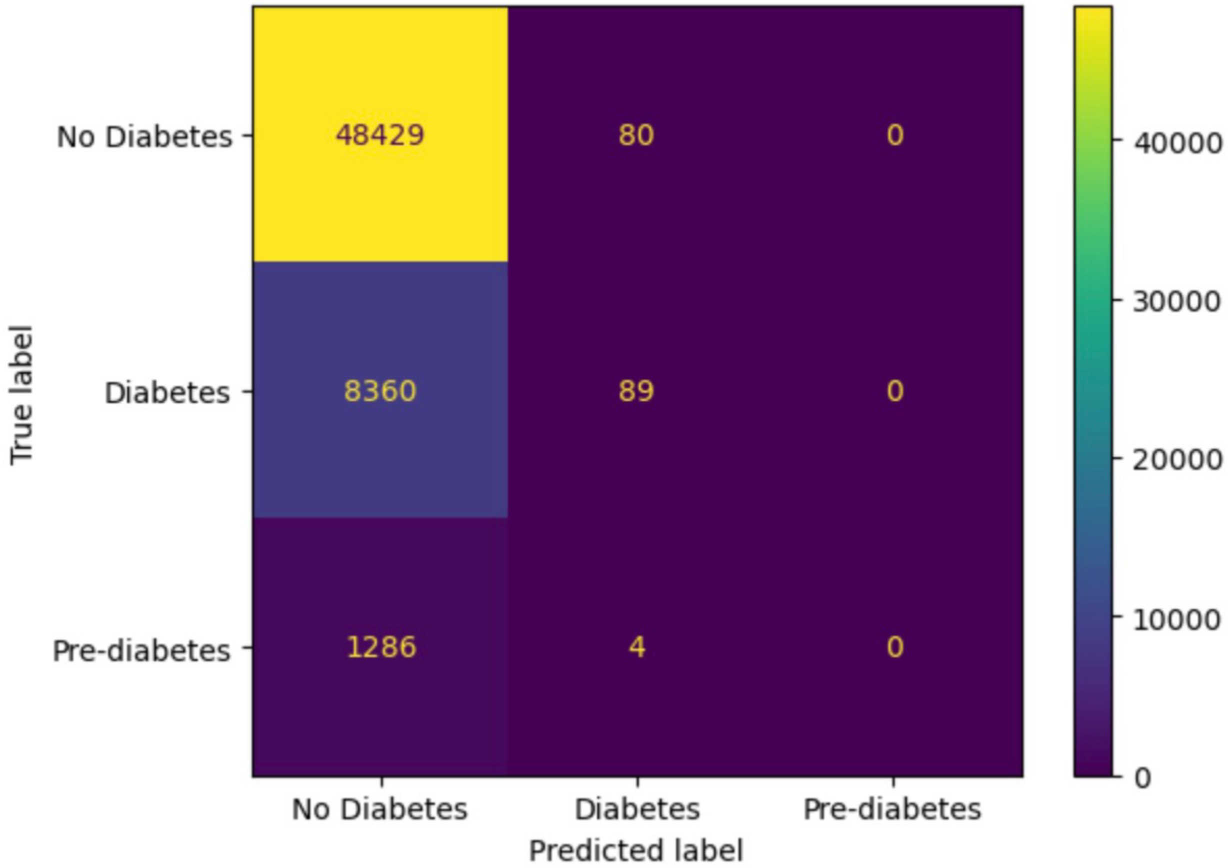
There is already a built-in model function in scikit-learn library so we will not build our own model instead just use the library function.

Given that there are only 2-4 parameters, the Naive Bayes model is considered the simplest model in building. Considering that we will hyperparameter tuning in later step, we choose to use the default parameter setting when building the basic model.

Cross-validation and Hyperparameter tuning

As we already had the cleaned dataset, it has then been used in building the three models. First, we use cross validation of the training dataset for each model to compare the accuracy of each model on default parameter setting.

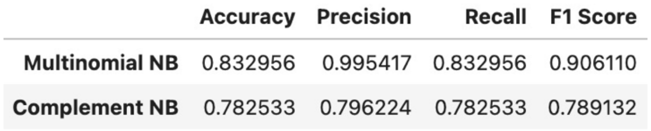
The result implicated that Gaussian is the worst model on our dataset but Bernoulli and Multinomial had similar performance scores which need to be evaluated further. Both models have an alpha parameter (also known as the smoothing parameter) used to address the issue of zero probabilities which is a hyperparameter that needs to be tuned. The turning result shows that after setting the alpha value to optimal, Multinomial is the model produced highest accuracy.



However, after we plotted the confusion matrix plot of the multinomial model on test dataset, we found out that there’s problem on the model result. The model is only good at classifying the majority label but performs bad on the minority label.

In the context of our classifying project, we are aiming to find out the Diabetes patients rather than people without diabetes. Therefore, the model does not fit our perspective even though it has a well performance score. We believe what causes the biased results is the imbalanced dataset.

There’s another model called Complement Naive Bayes which is specifically used for imbalanced dataset. We repeated the same process of cross validation and hyperparameter tuning on the model getting the best parameter values. Even though the overall performance score of this model is 20% lower than the Multinomial model, the classification result is surprisingly much more evenly distributed. Instead of classifying all the patients into no diabetes group, we can now identify some diabetes or pre-diabetes patients. To further raise the accuracy of classification, we can combine the complement model with other methods.



	Accuracy	Precision	Recall	F1 Score
Multinomial NB	0.832956	0.995417	0.832956	0.906110
Complement NB	0.782533	0.796224	0.782533	0.789132

Decision Tree

Introduction

Decision Tree (DT for short) is a popular supervised machine learning algorithm used for both classification and regression tasks. It partitions the data into subsets based on certain features, aiming to predict the target variable's value, a continuous prediction or discrete classification. It is a tree-like structure where each internal node represents a "decision" based on a feature, each branch represents the outcome of that decision, and each leaf node represents the final decision or outcome. The decision of which feature to split on and where to split is determined by a metric like Gini impurity or information gain (for classification) or variance reduction (for regression). There are many advantages of using DT:

- Easy to understand and interpret.
- Require little data preprocessing, such as normalization, compared to other algorithms.
- Able to handle both numerical and categorical data.
- Can capture non-linear relationships between features and the target variable.

Implementation

There's already a built-in decision tree model in the scikit-learn library therefore we used the existing library model instead of building our own for efficiency, accuracy, and future maintenance.

There are 13 parameters in total but only 4 of them are considered as important feature in improving model performance: the criterion function measuring the quality of a split, the maximum depth of the tree, the minimum number of samples required to split an internal node and the minimum number of samples required to be at a leaf node. The above parameters are later been tuned for optimization.

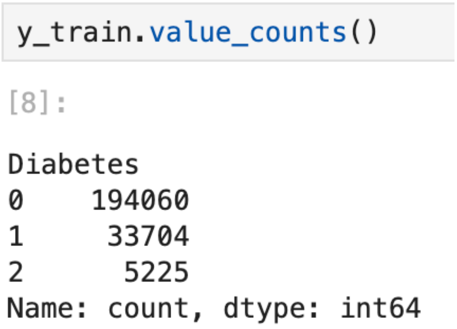
Cross-validation and Hyperparameter tuning

When building the basic decision tree, we choose to set all parameters as default and hyperparameter tuning them later. The result of the original model gives an accuracy of 0.88 on the training set and 0.8 on testing set which indicates we need to adjust each parameter for a better performance.

The steps here include two parts, first manually adjust the four parameters, criterion, max_depth, min_samples_split, min_samples_leaf, that are more important for DT model building than others, one by one. Then run grid search to find the best set of those four parameters based on the values got from the first step. At last, test the model and evaluate its performance on testing set.

Dealing with Super-Imbalance

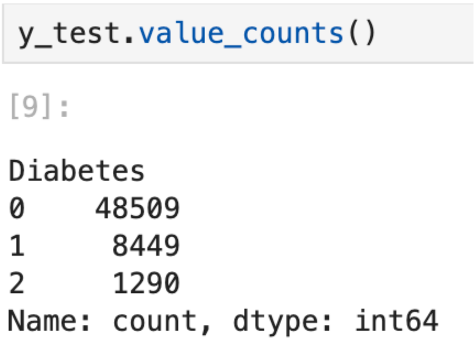
As shown in the previous discussion, the dataset is super imbalanced. Even though the accuracy is high, which due to the high proportion of non-diabetes, the model is not reliable enough to predict Diabetes correctly. To dealing with this situation, we first tried resampling the dataset, but the result was terrible. So, we decided to adjust the class weight, a parameter of decision tree model.



```
y_train.value_counts()
```

```
[8]:

Diabetes
0    194060
1     33704
2      5225
Name: count, dtype: int64
```

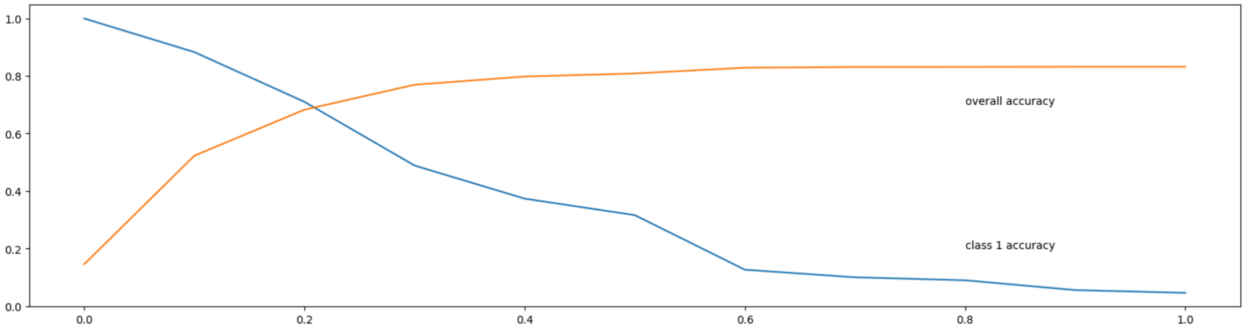


```
y_test.value_counts()
```

```
[9]:

Diabetes
0    48509
1     8449
2     1290
Name: count, dtype: int64
```

During testing different weight of class 0 (non-diabetes), we also apply the cross-validation, but manually separate the training set to training and validation set and calculate the accuracy of predicting class label 1 (diabetes), as well as the overall accuracy. Then we choose the weight of 0.2 at which those two lines across.



Our final Decision Tree model is built with parameters:

criterion	entropy'
max_depth	6
min_samples_leaf	10
min_samples_split	9
class_weight	{0:0.2}

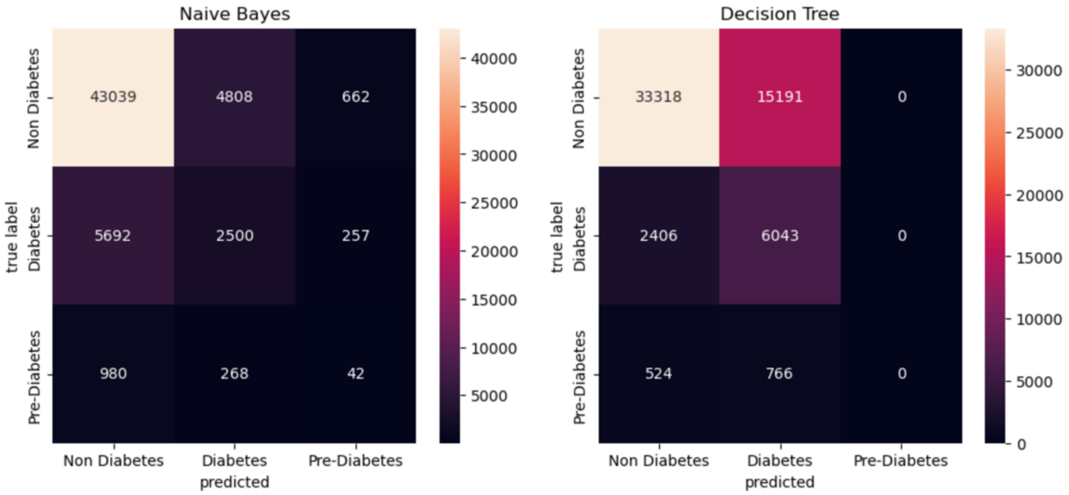
Performance summary

Comparation

	Naïve Bayes	Decision Tree
accuracy	**0.78**	0.68
precision - Diabetes	**0.33**	0.27
recall - Diabetes	0.30	**0.72**
f1-score - Diabetes	0.31	**0.40**

Naive Bayes has a higher accuracy compared to Decision Tree, and it demonstrates slightly higher precision for class label 1 compared to Decision Tree. Decision Tree exhibits significantly higher recall for class label 1 compared to Naive Bayes and it achieves a higher F1-score, indicating better overall performance in terms of precision and recall trade-off.

Considering the emphasis on detecting class label 1, the Decision Tree model appears to be more suitable due to its significantly higher recall and higher F1-score for this class. However, it's important to consider other factors such as computational resources, interpretability, and the specific requirements of the task before making a final decision.



```
Naive Beyes:
              precision    recall  f1-score   support

           0       0.87      0.89      0.88     48509
           1       0.33      0.30      0.31      8449
           2       0.04      0.03      0.04      1290

    accuracy                           0.78     58248
   macro avg       0.41      0.41      0.41     58248
weighted avg       0.77      0.78      0.78     58248

Decision Tree:
              precision    recall  f1-score   support

           0       0.92      0.69      0.79     48509
           1       0.27      0.72      0.40      8449
           2       0.00      0.00      0.00      1290

    accuracy                           0.68     58248
   macro avg       0.40      0.47      0.39     58248
weighted avg       0.81      0.68      0.71     58248
```

Best model

When reach to predict Diabetes, we debated about should we care about type I error more or type II error more. A type I error in medical area can lead to unnecessary treatments, surgeries, or emotional distress. In such cases, precision is crucial to minimize the number of false positives. While for diseases like sepsis, meningitis, or acute myocardial infarction, missing a diagnosis can be fatal. Recall is crucial to ensure that as many true positives as possible are detected.

In our case, since Diabetes is a relatively common disease, affecting over 400 million people worldwide. As such, the cost of false positives (e.g., unnecessary testing or treatment) is relatively low compared to the cost of false negatives (e.g., missing a diagnosis). This is because Diabetes is a serious disease that can lead to severe complications if left untreated. Missing a diagnosis can result in delayed treatment, which can lead to serious health consequences, including organ damage, blindness, and even death. Also having the fact that Diabetes treatment, such as medication and lifestyle changes, is generally safe and effective. As such, it is important to detect as many cases of diabetes as possible, even if some false positives occur.

Another argument occurred about should we focus on Diabetes class only or should we aim for an overall high performance matric. Trace back to our problem statement and dataset distribution, we aim to provide a basic insight on Diabetes prediction from an extremely imbalanced dataset where 83% of the records are no Diabetes. Considering the large amount of negative class record, there will be a high volume in false negative section, which we do not care about. Thus, it is better that we only compare the accuracy of Diabetes class when choosing model.

Having the idea that we want a model with high recall on classification output on class 1, we evaluate all our models by looking at classification report and choose the class-weighted decision tree to be our final model since it has the highest recall of classifying class 1.

Conclusion

From the feature engineering we found some factors that contribute to Diabetes result very much: age, Arthritis condition, education level, exercise routine, general health condition, gender, heart disease condition, income level and how the patients think their physical health condition are.

These factors can interact with each other and with diabetes in complex ways. For example: An older adult with arthritis may be more likely to develop diabetes due to chronic inflammation and reduced physical activity. Someone with a lower education level and lower income may face barriers to accessing healthcare and healthy food options, increasing their risk of developing diabetes. A person with a family history of heart disease and diabetes may be more likely to develop diabetes due to shared genetic risk factors.

In diabetes prediction, a high recall ensures that most individuals with diabetes are correctly identified, even if some individuals without diabetes are incorrectly identified as having the disease. This is particularly important in high-risk populations, such as those with a family history of diabetes or those who are overweight or obese.

Our final model is a class weighted decision tree which has a recall of 0.72 which means it can identify Diabetes patients in most of the cases even though we have a low precision of 0.27 at same time meaning many people might be identifying as Diabetes patients when they are not. Considering the above debate between precision and recall we conclude that our model has an overall good performance on achieving our problem objective of identifying Diabetes patients.

Reference

Centers for Disease Control and Prevention. (2023, November 15). *CDC - 2021 BRFSS survey data and Documentation*. Centers for Disease Control and Prevention. https://www.cdc.gov/brfss/annual_data/annual_2022.html

Xie, Z., Nikolayeva, O., Luo, J., & Li, D. (2019). Building Risk Prediction Models for Type 2 Diabetes Using Machine Learning Techniques. *Preventing chronic disease*, *16*, E130. https://doi.org/10.5888/pcd16.190109