Predicting of Diabetes and Prediabetes Given Various Health and Lifestyle Factors

Daniel Angel, Rajib Ratan Samanta, and Venkataraghavan Thottiyam Venkatakrishnan

Master of Science in Data Science Program, Bellevue University

Predictive Analytics DSC630-T301

Professor: Andrew Hua

Introduction:

Diabetes is a serious issue in society. Diabetes is a disease which affects the body's ability to process sugar. The disease and its effects are especially concerning in the United States of America. Statistics from the Centers for Disease Control and Prevention state that over 38 million Americans have the disease and nearly 30% are living with the disease but are unaware and undiagnosed (CDC 2022). Furthermore, nearly 100 million American adults are pre-diabetic which is over one-third of the population (CDC 2022).

There are severe risks to the life and well-being of those living with diabetes. Diabetes has significant risks of developing complications which include loss of limbs, disease of the kidneys, and potentially most concerning heart disease (Teboul 2021). If untreated, diabetes can be deadly. Diabetes does not have a cure but with a motivated effort involving exercise, nutritional diet, weight loss and medical intervention the most severe risks can be mitigated (Teboul 2021). Prediabetes is a state where blood sugar levels are concerning but not yet at the level of diabetes (CDC 2022). Prediabetics can actually prevent the disease from progressing to a diabetes diagnosis by following these same lifestyle changes (CDC 2022).

There is not only implications for an individual's health but also economic considerations as well. The estimated additional health care costs related to diabetes care for individuals is $12,000 per year (CDC 2022). The total societal cost in the United States is over $400 billion! (CDC 2022)

Clearly, this is something that we as a society should attempt to address. Luckily, the US government has been collecting data related to this for nearly 40 years (Teboul 2021). Since 1984, a survey has been conducted and data has been collected in the form of the BRFSS, the Behavioral Risk Factor Surveillance System (BRFSS).

Data Selection and Project Proposal:

Our term project will focus on one year's worth of data collected from this survey (2015, to be specific). This data set contains 253,680 survey responses and can be found on Kaggle.com at the URL https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset . The target variable we are most interested in is Diabete_012 which contains 3 classes: 0,1 and 2 corresponding to non-diabetic, prediabetic, and diabetic, respectively.

To give a general idea of features contained in the data set, it includes information related to BMI (Body Mass Index), cholesterol and blood pressure levels, and a history of heart attacks. It also has features for survey responses about diet, exercise, and self-evaluation of one's physical and mental health. Finally, the data set contains demographics such as age, sex, education and income.

Our group aims to produce a predictive model with the power to forecast whether one is at risk of being prediabetic or having diabetes based on a variety of factors. Being able to accurately diagnose this disease, diabetes, or a predisposition to it, prediabetes, with a non-trivial level of certainty without actually needing physical access to a patient could theoretically save lives and associated health care costs!

Model Selection:

Selecting a model is an iterative process and is dependent on several things. After considering the facts surrounding our data set and accounting for what we aim to accomplish with our models, we have arrived at the conclusion that we should use the SVM classification model from Python's scikitlearn library.

Reasoning for Model Selection:

Firstly, we must narrow down our choice of model since our target is a classification problem rather than a regression or clustering one. Also, the size of the data set should impact our eventual selection. In our case, we have over 100,000 rows in our array. Finally, we know that our data is labeled.

SVMs, support vector machines, are effective in capturing complex, non-linear relationships in the data. The risk factors for diabetes often exhibit non-linear patterns, and SVMs can model these relationships more flexibly compared to linear models. In healthcare data, especially when considering risk factors like BMI, cholesterol levels, and blood pressure, the feature space can be high-dimensional. SVMs perform well in high-dimensional spaces and can handle a large number of features effectively.

Model Evaluation:

Our belief is that model evaluation should be done in two phases. Firstly, we should identify whether our initial selection of SVM Classifier was truly the best suited. We can compare it against other Classifiers using k-fold cross-validation and measure which is most appropriate based on basic metrics such as Accuracy and F1 Score. This should be done after certain crucial steps such as data cleaning, and feature engineering and selection.

Once we are in fact certain that our selection is the best we will need to determine how good our model is actually. Since this is a classification question we will certainly need to produce a confusion matrix to visually understand the effectiveness of our model. Again, we will look at metrics such as Accuracy and F1. However, now that our hyperparameters have been tuned we will want to place additional emphasis on Recall and AUC-ROC. These two metrics, in particular, might be the most important for evaluating our model. Recall, as opposed to Precision, is best suited because False Negatives are of higher concern than False Positives. We would rather misdiagnose and have someone adopt healthy lifestyle changes than have someone slip through the cracks and miss out on a potentially life-saving medical intervention. Finally, AUC-ROC is a good summary statistic for our model due to the fact that it measures the classifier’s ability at class distinctions.

Desired Outcomes:

The aim of our project is to address a few key questions.

Which behaviors are most closely linked with a diabetes or prediabetes diagnosis?

Can we accurately predict diabetes within an individual, knowing certain characteristics?

Is the BRFSS, as currently constructed, the best remote diagnosis tool at our disposal?

Could we remove questions from the BRFSS or add entirely new ones to create an even better or just as good model for prediction?

Risks:

With any predictive model there are several risks. One such risk could possibly be insufficient or bad data. There is also some likelihood that the model we have initially selected is inappropriate or simply not the best suited for our project. Both of these risks must be accounted for, and we should have a contingency plan in place in the event of this occurrence. Also, there is a risk that other environmental factors have changed in the 8 years since the survey and so it would no longer work in the current time period (2023).

There is also the unfortunate risk that even with a well-trained model we are not able to achieve our goals of producing an accurate predictor.

Ethical Implications:

Patient data, especially health information, is sensitive. Ensuring privacy and data security is crucial to prevent unauthorized access or misuse. It is imperative that individuals are adequately informed about how their data will be used and obtain their consent for participation in the study.

Biases in the training data or the model itself can lead to unfair outcomes, particularly in healthcare where disparities may already exist. Therefore, it is essential that the model's predictions align with clinical knowledge and contribute meaningfully to patient care. It is crucial to meticulously assess the enduring long term consequences of these predictions on individuals' lives, encompassing the possible effects of stigma and psychological implications.

Contingency Plan:

The first step involves a diagnostic assessment to pinpoint the specific reasons for failure, considering factors such as data quality, feature engineering challenges, and algorithm suitability. Diversifying modeling approaches by experimenting with alternative algorithms, architectures, and hyperparameters is crucial. Feature engineering and data enhancement should be revisited to address any deficiencies in the dataset. Hyperparameter tuning and the exploration of ensemble methods, which combine predictions from multiple models, offer avenues for improved performance. Opting for an alternative model, incorporating a broader range of years in the dataset, selecting distinct years for analysis, integrating supplementary data sources, or even opting for an entirely different project topic are viable strategies for refining the approach.

Ability to Answer Questions:

The dataset has been evaluated and determined that it provides sufficient information to answer the primary research questions related to predicting the risk of prediabetes or diabetes. It has been ensured that key variables, such as those related to health indicators and lifestyle factors, are present and informative. The source of our data is the Centers for Disease Control and Prevention, a U.S. government agency and so we can be fairly certain that it is trustworthy. Upon inspection of the data, no missing values were found which is further confirmation that this dataset should be well suited to our needs. Finally, owing to the large size of the dataset which contains over a quarter million rows, even if we were to discover that the dataset contains bias or skew it is still possible to take a sizable subset which would be balanced and or normal and would still satisfy our requirements and give us what we need to address our questions.

Useful Visualizations:

The best visualization to understand the distribution of our target variable, Diabetes_012, is the histogram. Based on the amount of patients within those classes we will able to analyze what percent of each category (Non-diabetic, pre-diabetic, and diabetic) exist within the whole of our data. Scatter plot will be helpful for dependent variables impact analysis.

Visualizations such as bar charts, box plots, Correlation Matrix , ROC curves are particularly useful for explaining the data. Bar charts can show the makeup of the individual variables and box plots will visually describe center tendencies and summary statistics of each variable. Correlation Matrices visually represent the correlation between different variables. They help identify the strength and direction of relationships, which is crucial for feature selection and model building.

Once it is time to deploy our model we can also employ a particular visualization to evaluate our model. ROC curves are used to visualize the trade-off between true positive rate and false positive rate at different classification thresholds. They help assess the performance of binary classification models which our project happens to be.

Data and Driving Questions Adjustment:

Based on the initial exploration we don't consider it is necessary to adjust the data and or driving questions. We find that the dataset and the data contained within align with the goal of predicting diabetes risk. We feel fairly confident that the data as it currently stands should give us the ability to answer our questions and therefore no change is needed at this time. Our driving questions were informed by the dataset and as we are not making changes to our data we will leave our queries as is.

Model and Evaluation Choices Adjustment:

Initially, we had considered using a Support Vector Machine model as the model for our term project to help us answer our questions. SVM works best with classification so it's appropriate in that regard. Also, SVM models work best with complex and smaller datasets. Our dataset is definitely complex but it might be hard to argue it is smaller.

Sometimes simpler is better so I think our workflow should involve starting with Logistic Regression. If it doesn't perform well, then we try SVM. Regardless of model used, we plan to apply Stochastic Gradient Descent (SGD) algorithm for tuning. It's good with large datasets and its stochastic nature allows it to update model parameters based on small random subsets of the data, making it computationally efficient and scalable to handle large amounts of training data.

By producing a confusion matrix we can visually understand the effectiveness of our model and in this way perform the first step in evaluating our model.

By reassessing the model & based on the insights gained, there might be a need to adjust the choice of algorithm, hyperparameters, or evaluation metrics for the sake of optimization. It should also be considered if the selected model is suitable for capturing the complexity of the relationships in the data. As we learn more about Ensemble models perhaps that is an approach we could take. However, since our problem is still of a classification type we should basically be using the same evaluation methods as discussed in Milestone 2 such as Accuracy and the F1 Score but with a special emphasis given to Recall and AUC-ROC and with particular importance given to Recall because of the risk associated with False Negatives in health care and disease identification.

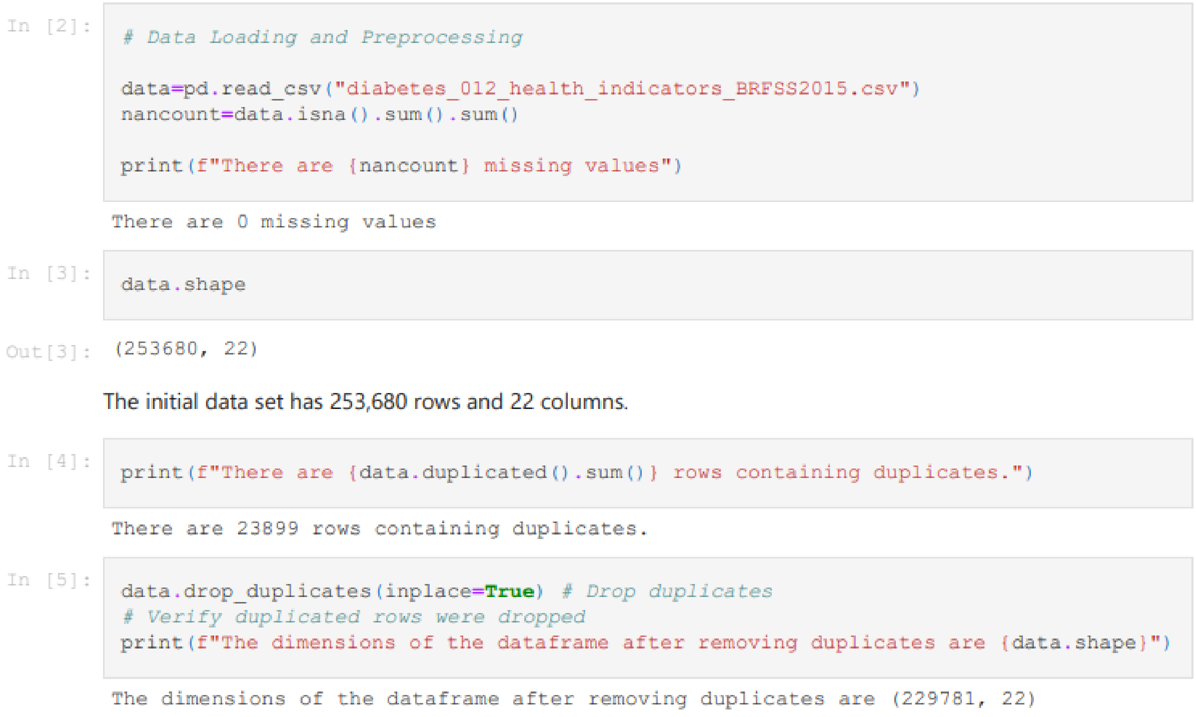
Reasonableness of Original Expectations:

Our original expectation is to detect diabetes or prediabetes at an early stage. Early identification allows for timely intervention and lifestyle modifications to prevent the progression to full-blown diabetes. Our models can contribute to patient engagement by raising awareness about diabetes risk. Providing individuals with information about their risk status encourages proactive health management and preventive measures.

We believe our original expectations are still reasonable based on the patterns observed in the data. If the data reveals unexpected trends or challenges, we will be open to adjusting the expectations and hypotheses.

Explain your process for prepping the data

As discussed in Milestone 3 the data set contained no missing data so one data preparation step was not needed. We did find duplicates which were discarded to produce a new, smaller data set containing no duplicates.

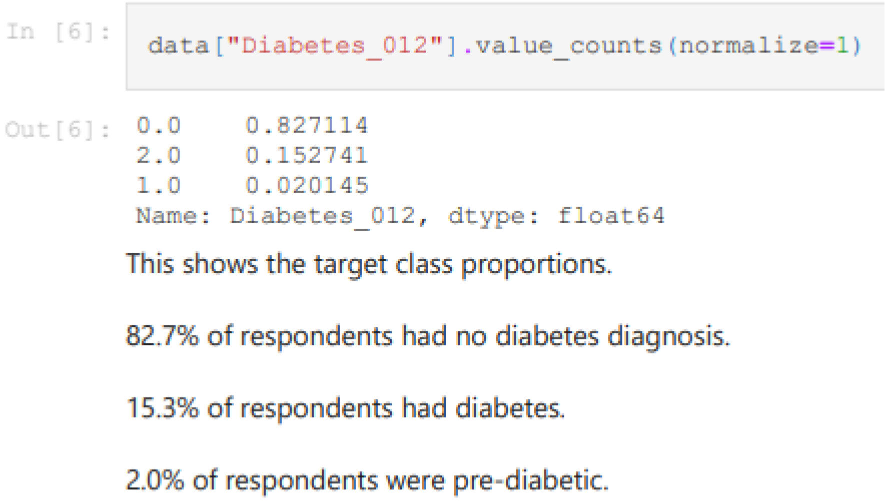


```
In [2]: # Data Loading and Preprocessing

        data=pd.read_csv("diabetes_012_health_indicators_BRFSS2015.csv")
        nancount=data.isna().sum().sum()

        print(f"There are {nancount} missing values")

        There are 0 missing values

In [3]: data.shape

Out[3]: (253680, 22)
```

The initial data set has 253,680 rows and 22 columns.

```
In [4]: print(f"There are {data.duplicated().sum()} rows containing duplicates.")

        There are 23899 rows containing duplicates.

In [5]: data.drop_duplicates(inplace=True) # Drop duplicates
        # Verify duplicated rows were dropped
        print(f"The dimensions of the dataframe after removing duplicates are {data.shape}")

        The dimensions of the dataframe after removing duplicates are (229781, 22)
```

Next, we displayed the proportions of our target classes.



```
In [6]: data["Diabetes_012"].value_counts(normalize=1)

Out[6]: 0.0    0.827114
        2.0    0.152741
        1.0    0.020145
        Name: Diabetes_012, dtype: float64
```

This shows the target class proportions.

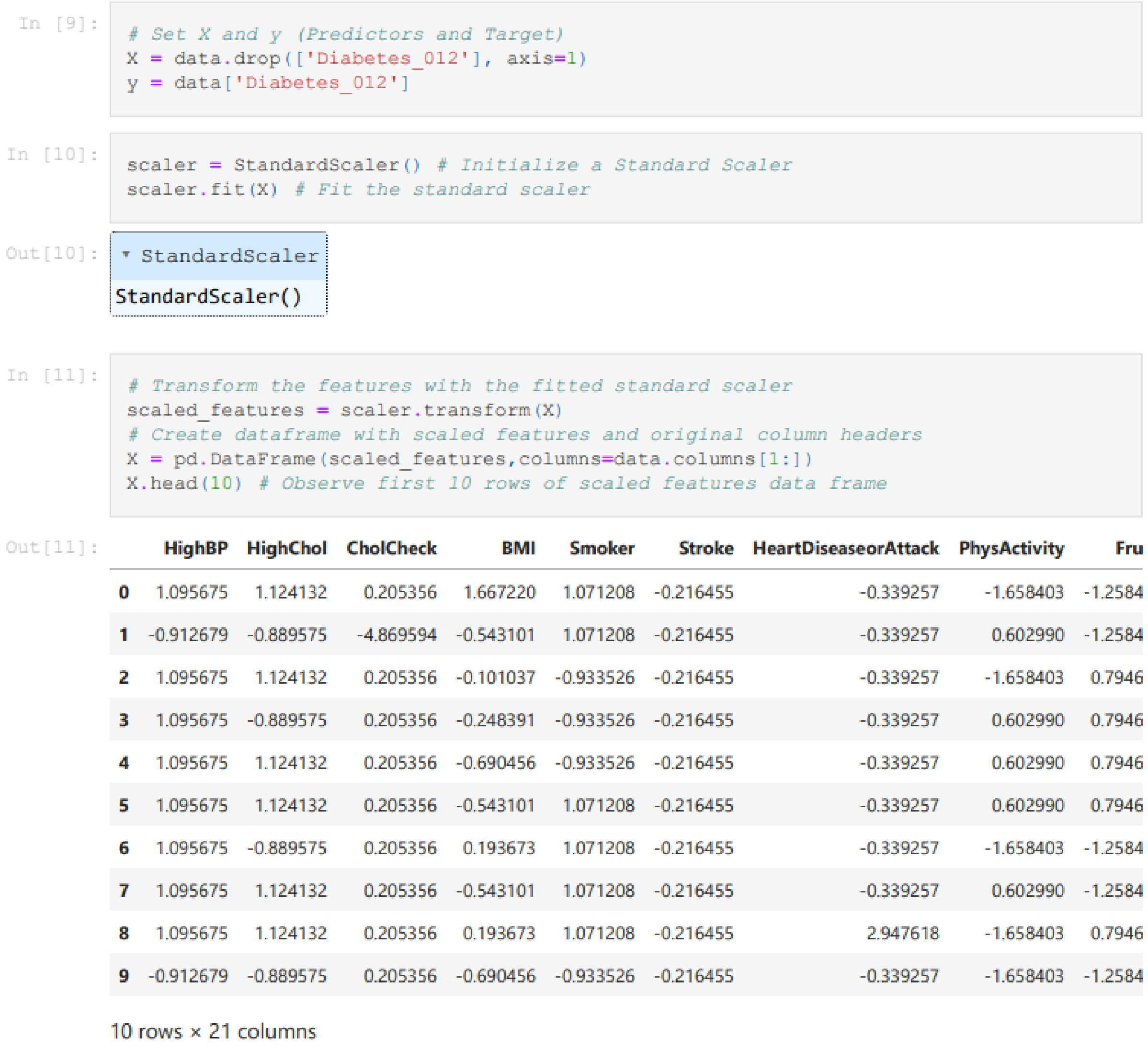
82.7% of respondents had no diabetes diagnosis.

15.3% of respondents had diabetes.

2.0% of respondents were pre-diabetic.

We notice that we have class imbalance. Dealing with the imbalanced classes of our target variable will be one of our data preparation steps.

Before resampling we performed a couple of additional data preparation steps. First, we separated our data frame into predictor variables (X ; all features except Diabetes_012) and target variable (y ; Diabete_012). Then we applied a scaler to our predictor variables, specifically a Standard scaler.



```
In [9]:  # Set X and y (Predictors and Target)
         X = data.drop(['Diabetes_012'], axis=1)
         y = data['Diabetes_012']
```

```
In [10]: scaler = StandardScaler() # Initialize a Standard Scaler
         scaler.fit(X) # Fit the standard scaler
```

```
Out[10]: ▾ StandardScaler
         StandardScaler()
```

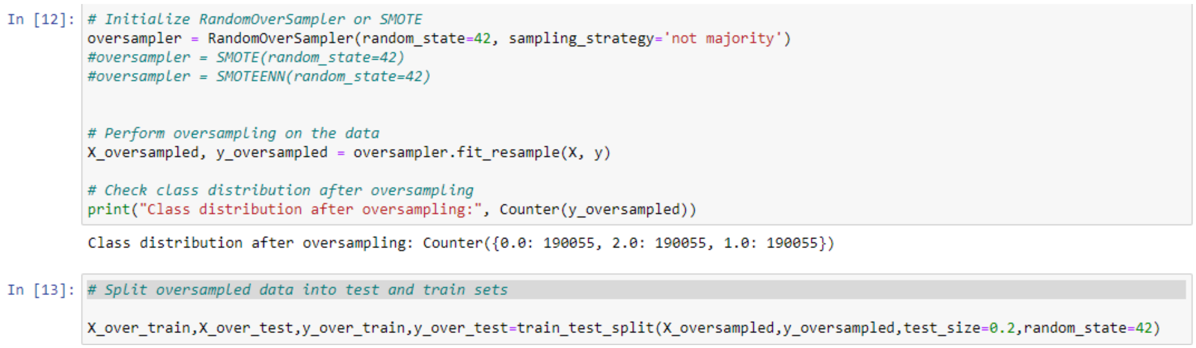
```
In [11]: # Transform the features with the fitted standard scaler
         scaled_features = scaler.transform(X)
         # Create dataframe with scaled features and original column headers
         X = pd.DataFrame(scaled_features,columns=data.columns[1:])
         X.head(10) # Observe first 10 rows of scaled features data frame
```

Out[11]:

	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fru
0	1.095675	1.124132	0.205356	1.667220	1.071208	-0.216455	-0.339257	-1.658403	-1.2584
1	-0.912679	-0.889575	-4.869594	-0.543101	1.071208	-0.216455	-0.339257	0.602990	-1.2584
2	1.095675	1.124132	0.205356	-0.101037	-0.933526	-0.216455	-0.339257	-1.658403	0.7946
3	1.095675	-0.889575	0.205356	-0.248391	-0.933526	-0.216455	-0.339257	0.602990	0.7946
4	1.095675	1.124132	0.205356	-0.690456	-0.933526	-0.216455	-0.339257	0.602990	0.7946
5	1.095675	1.124132	0.205356	-0.543101	1.071208	-0.216455	-0.339257	0.602990	0.7946
6	1.095675	-0.889575	0.205356	0.193673	1.071208	-0.216455	-0.339257	-1.658403	-1.2584
7	1.095675	1.124132	0.205356	-0.543101	1.071208	-0.216455	-0.339257	0.602990	-1.2584
8	1.095675	1.124132	0.205356	0.193673	1.071208	-0.216455	2.947618	-1.658403	0.7946
9	-0.912679	-0.889575	0.205356	-0.690456	-0.933526	-0.216455	-0.339257	-1.658403	-1.2584

10 rows × 21 columns

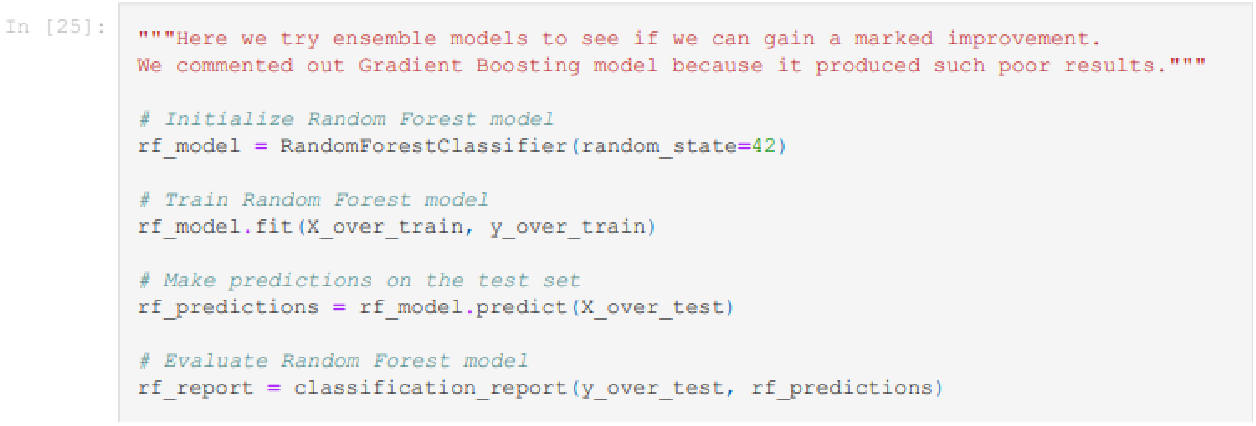
Finally, we perform resampling. We chose Oversampling as opposed to undersampling as it is considered the superior method. We tried 3 oversamplers: RandomOverSampler, SMOTE, and SMOTEENN. Coincidentally, the best performing oversampler was also the fastest – RandomOverSampler. After oversampling with RandomOverSampler we displayed the class distribution to determine if the classes were balanced now, which they were. Lastly, just prior to model building and deployment we need to split our data set. We split our oversampled data set into training and testing sets using the 80-20 Train-Test split.



```
In [12]: # Initialize RandomOverSampler or SMOTE
         oversampler = RandomOverSampler(random_state=42, sampling_strategy='not majority')
         #oversampler = SMOTE(random_state=42)
         #oversampler = SMOTEENN(random_state=42)


         # Perform oversampling on the data
         X_oversampled, y_oversampled = oversampler.fit_resample(X, y)

         # Check class distribution after oversampling
         print("Class distribution after oversampling:", Counter(y_oversampled))

         Class distribution after oversampling: Counter({0.0: 190055, 2.0: 190055, 1.0: 190055})

In [13]: # Split oversampled data into test and train sets

         X_over_train,X_over_test,y_over_train,y_over_test=train_test_split(X_oversampled,y_oversampled,test_size=0.2,random_state=42)
```

Build and evaluate at least one model

We actually built and tested three separate models. These were DecisionTreeClassifier (94% accurate), XGBClassifier (92% accurate), and RandomForestClassifier (96% accurate). For the XGBClassifier, we used a RandomizedSearch Cross Validation to determine the optimal hyperparameters. We used the same metrics for evaluating all of our models. This included a classification report containing Precision, Recall, F1 Score, and Accuracy. We also looked at a Confusion Matrix for each which showed the amount of patients with correctly predicted classes and also the count of each type of incorrectly predicted patient. We concluded by producing a ROC-AUC curve for each model also. Even without tuned hyperparameters, the RandomForestClassifier performed superiorly in almost every aspect.



```
In [25]: """Here we try ensemble models to see if we can gain a marked improvement.
         We commented out Gradient Boosting model because it produced such poor results."""

         # Initialize Random Forest model
         rf_model = RandomForestClassifier(random_state=42)

         # Train Random Forest model
         rf_model.fit(X_over_train, y_over_train)

         # Make predictions on the test set
         rf_predictions = rf_model.predict(X_over_test)

         # Evaluate Random Forest model
         rf_report = classification_report(y_over_test, rf_predictions)
```

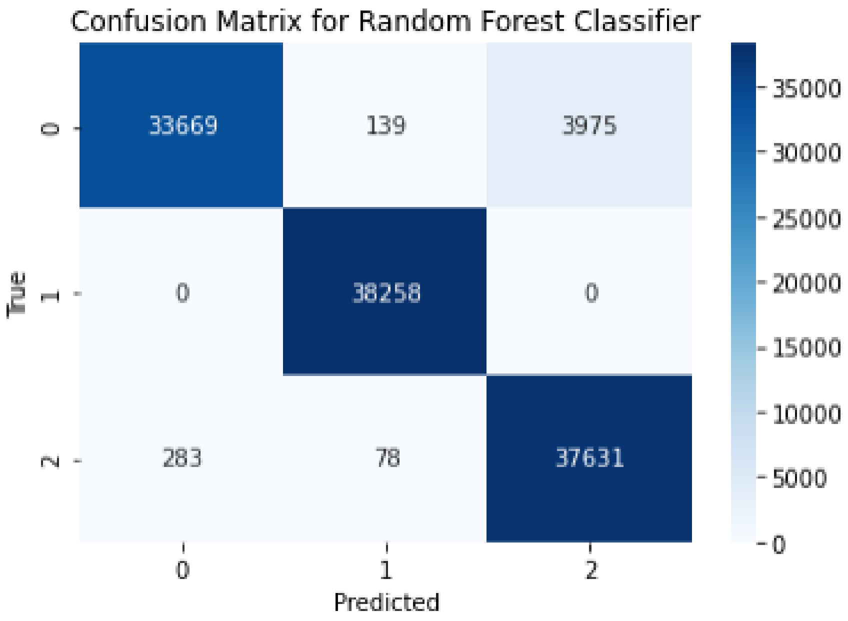
Our trained RandomForestClassifier model performed with an astonishingly high accuracy of 96%. Perhaps even more impressive than this level of accuracy is the Recalls of 1 and .99 for

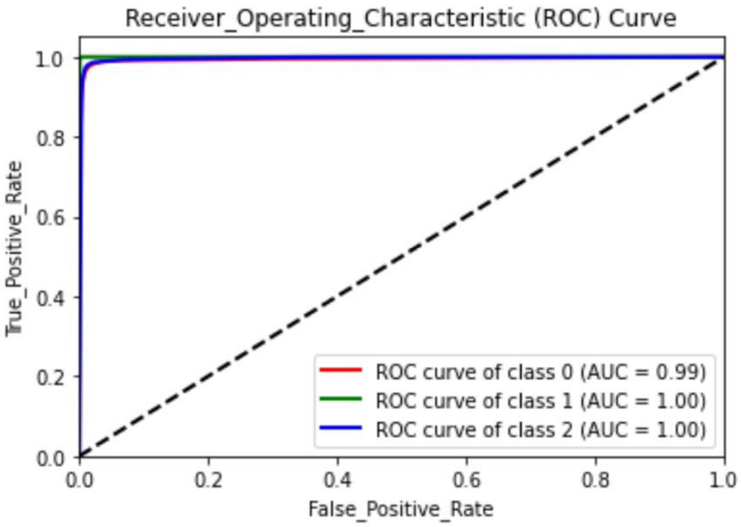
Diabetes_012=1 (Prediabetic) and Diabetes_012=2 (Diabetic), respectively. It might be possible to obtain a higher accuracy but it would be highly unlikely to achieve better Recall. The reason why Recall is prioritized over other metrics is, as discussed in Milestone 3, due to dealing with medical diagnosis. Medical diagnosis is a case where False Negatives should be minimized as much as possible. A false negative in medical diagnosis can mean a patient potentially missing lifesaving medical care or attention. Other than correct predictions, the next highest count in our confusion matrix belongs to actual 0’s predicted as 2’s (i.e. patients without diabetes predicted to have diabetes). This is okay because a simple test can determine if a patient has diabetes or not.

```
print("Random Forest Classification Report:")
print(rf_report)

# print("\nGradient Boosting Classification Report:")
# print(gb_report)
```

```
Random Forest Classification Report:
              precision    recall  f1-score   support

         0.0       0.99      0.89      0.94     37783
         1.0       0.99      1.00      1.00     38258
         2.0       0.90      0.99      0.95     37992

    accuracy                           0.96    114033
   macro avg       0.96      0.96      0.96    114033
weighted avg       0.96      0.96      0.96    114033
```

```
In [26]:
cmatrix = confusion_matrix(y_over_test, rf_predictions)
sns.heatmap(cmatrix, annot=True, fmt='g', cmap=plt.cm.Blues)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix for Random Forest Classifier')
plt.show()
```


Confusion Matrix for Random Forest Classifier



Interpret your results

We interpret that our model has resulted in fairly accurate prediction of diabetes in patients given a detailed medical history. The high accuracy should raise red flags that perhaps the model is overfitted. The size of the test and training sets lowers this probability but the chance still exists. Under some circumstances our XGBClassifier that used tuned hyperparameters found from a randomized search cross validation was an improvement over the Decision Tree Classifier we first tried but generally it was not and it definitely did not show an improvement which was proportional to the increased runtime.

Only 361 prediabetics or diabetics out of 76,250 total were missed by our model. In other words, this accounts to less than half a percent (.0047) or 47 out of 10,000. This represents a very small risk.

Begin to formulate a conclusion/recommendations

To ensure our model is fair, equitable and useful we should perform additional tasks. We could use Principal Component Analysis to see if it is an improvement. Also, could try a few more models to see if any extra improvement is possible. Tests should be performed to ensure our model isn’t overfitted. A potential solution could be using Train/Test/Validate sets instead of solely Train/Test as we have been.

To better understand the impact of the feature variables on the predictions it would be advisable to display the features’ importance both numerically and visually. This could serve to improve the quality of the health survey as well as helping to understand which particular factors are most closely related to diabetes and which lifestyle changes could best help healthy individuals from crossing into prediabetes or diabetes, help avoid prediabetes from progressing to diabetes, and help in directly reversesing diabetes.

Without taking any of these steps the model and its results can only be taken as guidance not as gospel. To use the model as definitive at this point could be unethical and potentially dangerous.

References

"By the Numbers: Diabetes in America." Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 25 Oct. 2022, www.cdc.gov/diabetes/health-equity/diabetes-by-the-numbers.html.

"Choosing the Right Estimator." Scikit, scikit-learn.org/stable/tutorial/machine_learning_map/. Accessed 7 Dec. 2023.

"Prediabetes - Your Chance to Prevent Type 2 Diabetes." Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 30 Dec. 2022, www.cdc.gov/diabetes/basics/prediabetes.html.

Teboul, Alex. "Diabetes Health Indicators Dataset." Kaggle, 8 Nov. 2021, www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset.