Milestone 5: Final Project Paper and Presentation

Type 2 Diabetes Predictions Using Machine Learning

Monica Santana

Bellevue University

DSC 630: Predictive Analytics

Professor Andrew Hua

May 28, 2024

Introduction

My term project's analysis is on type 2 diabetes and factors that contribute. The dataset I will be using includes medical and demographic information of women to predict diabetes. To show the magnitude of diabetes in America, here are some statistics. 1 in 9 adult women which is roughly 15 million women in the United States are living with diabetes (Charles, 2022). Approximately 96 million American adults, more than 1 in 3, have prediabetes (Wergin, 2023). Out of those Americans, 80% don't know they have prediabetes since the symptoms are not clear and often goes undetected until type 2 diabetes shows up (Wergin, 2023).

This is why it is extremely important to help solve this problem early on before it is too late. Once one has type 2 diabetes, it is irreversible. There are no medications or treatments that can cure type 2 diabetes. The only thing one can do is have a healthy diet and increase exercise to lower and control the blood sugar level. If those two aren't enough, then one would have to take diabetes medications or insulin therapy for the rest of their lives. Diabetes also gradually creates long term complications the longer you have it. Possible complications include nerve damage (diabetic neuropathy), heart disease including heart attacks and strokes, kidney damage (diabetic nephropathy), eye damage (diabetic retinopathy), depression, and increase risk of dementia such as Alzheimer's disease (Pruthi, 2023). For women specifically, it can create complications during pregnancies and lead to complications for the baby (Pruthi, 2023). This includes excess baby growth, low blood sugar, a higher risk of developing obesity and type 2 diabetes later in the baby's lives and even death before or after the baby's birth (Pruthi, 2023).

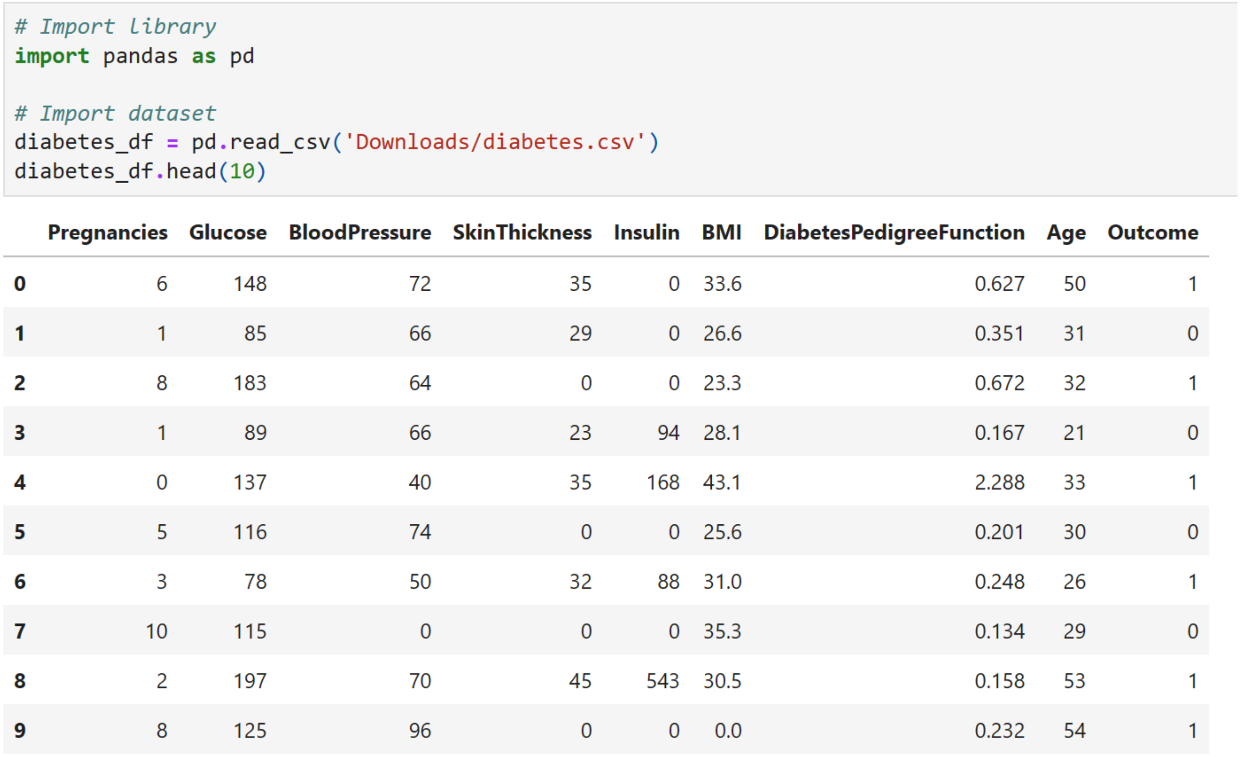
Those who would be interested in solving this problem are universal. My dataset only contains information on women, but mostly everyone has a woman in their life that is important to them. Therefore, anyone would be interested in solving this problem. I would try to sell this project to the healthcare system and hospitals who already have data on patients who are prediabetic or those who need to have a healthier lifestyle to avoid the fate of irreversible diabetes. It is crucial to spread the awareness of diabetes early on to help prevent it. So, this project can also be sold to the education system to teach students the importance of taking care of one's health early on to avoid being prediabetic when they are older.

My data was retrieved from Kaggle, and it is useful to help solve this problem because it focuses on different health related attributes that are associated with diabetes. The data is composed of 769 rows and 9 columns. The columns/features in the dataset are as follow:

1. Pregnancies: The number of times the woman has been pregnant.
2. Glucose: Glucose concentration in the woman's plasma.
3. Blood Pressure: Blood pressure measurement.
4. Skin Thickness: Thickness of skinfold at the triceps.
5. Insulin: Insulin levels in the blood.
6. BMI (Body Mass Index): A measure of body fat based on height and weight.
7. Diabetes Pedigree Function: A function that scores the likelihood of diabetes based on family history.
8. Age: Age of the woman.
9. Outcome: The target variable indicating whether the woman has diabetes or not (1 for diabetic, 0 for non-diabetic).

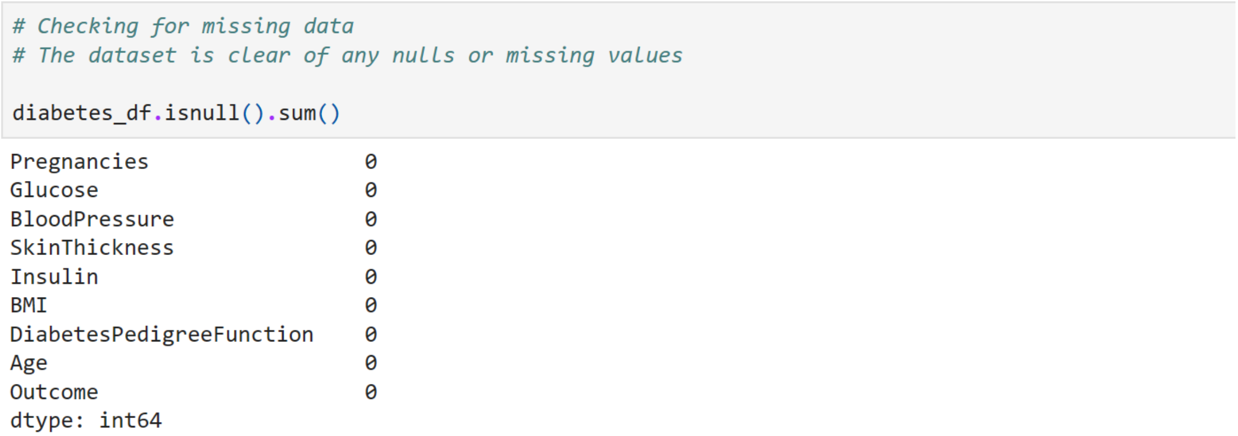
Methods/Results

The dataset didn't need any cleaning and/or transforming so the steps to prepare the data were very limited. It was already free of any nulls or missing data. The columns and the variable names were also adequate for analysis, so I didn't have to change those either. The important thing to note here is the outcome variable, which is 0 for non-diabetic diagnosis, and 1 for diabetic. The “outcome” column is my target variable for my models with the final diagnosis.



```
# Import library
import pandas as pd

# Import dataset
diabetes_df = pd.read_csv('Downloads/diabetes.csv')
diabetes_df.head(10)
```

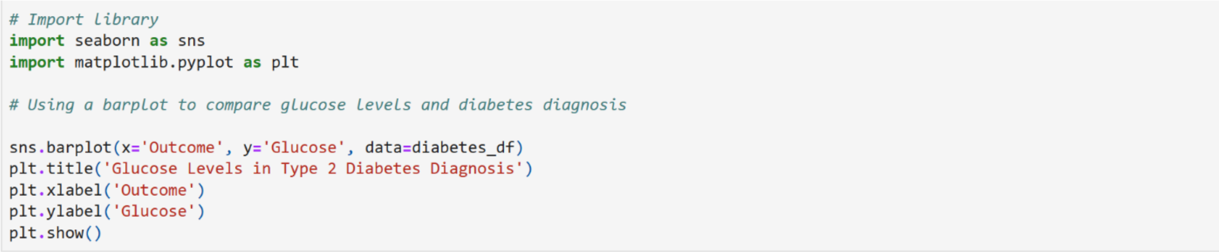
	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1



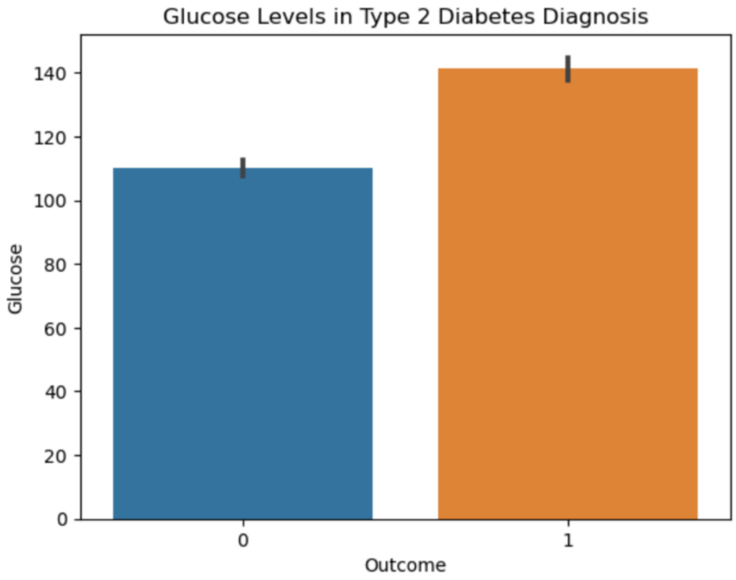
```
# Checking for missing data
# The dataset is clear of any nulls or missing values

diabetes_df.isnull().sum()
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

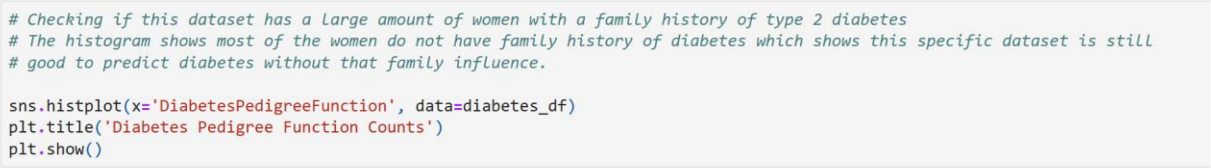
There are a few visualizations to help tell a story with the data. First, being the average glucose levels for the diabetics and non-diabetics. I created a bar plot to see what those levels are, and it showed the non-diabetic (outcome 0) average glucose levels were 110. The average glucose levels for diabetics (outcome 1) were greater, 140. This helps women catch diabetes early on as one can look at their own glucose levels and compare to the average levels from the dataset. Getting one's glucose levels checked regularly is a good habit to have.



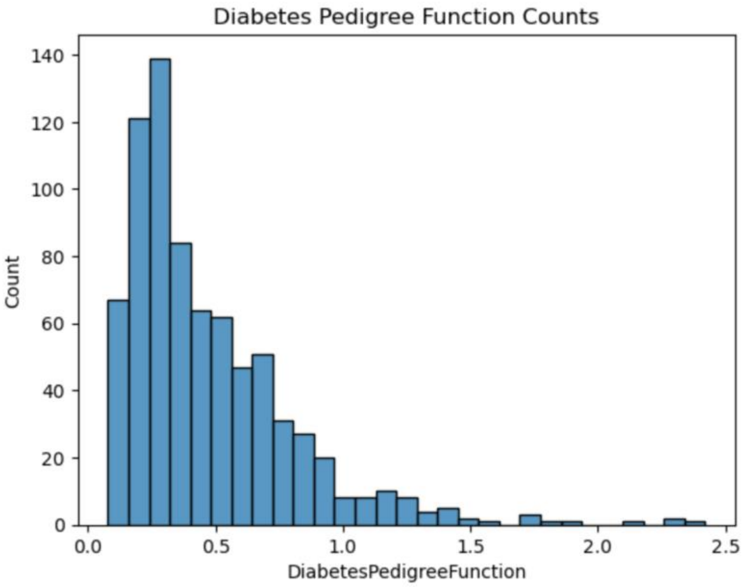
```
# Import library
import seaborn as sns
import matplotlib.pyplot as plt

# Using a barplot to compare glucose levels and diabetes diagnosis

sns.barplot(x='Outcome', y='Glucose', data=diabetes_df)
plt.title('Glucose Levels in Type 2 Diabetes Diagnosis')
plt.xlabel('Outcome')
plt.ylabel('Glucose')
plt.show()
```



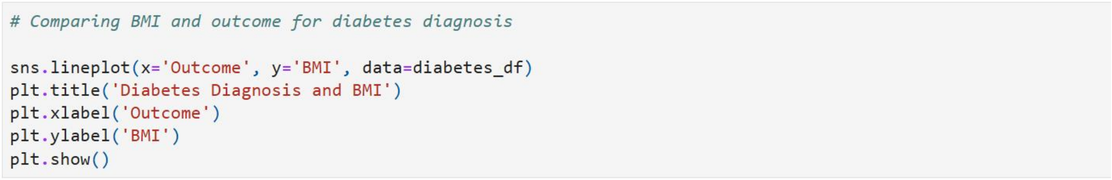
The next visualization that helps tell a story with the data is whether most of the women in the dataset were already predisposed to having diabetes from family history. This would be equivalent to the pedigree function. If they were, then it would create a bias and not account for those who got diabetes without previous family history. I created a histogram to take a better look at the count for the degree pedigree function variable, and it showed that most of the women in the data have a very low score which means they were not likely to have family history of diabetes. A low score means it is much closer to 0, which indicates no family history. The closer to 1 the diabetes pedigree function is, the greater chance of a family history of type 2 diabetes.



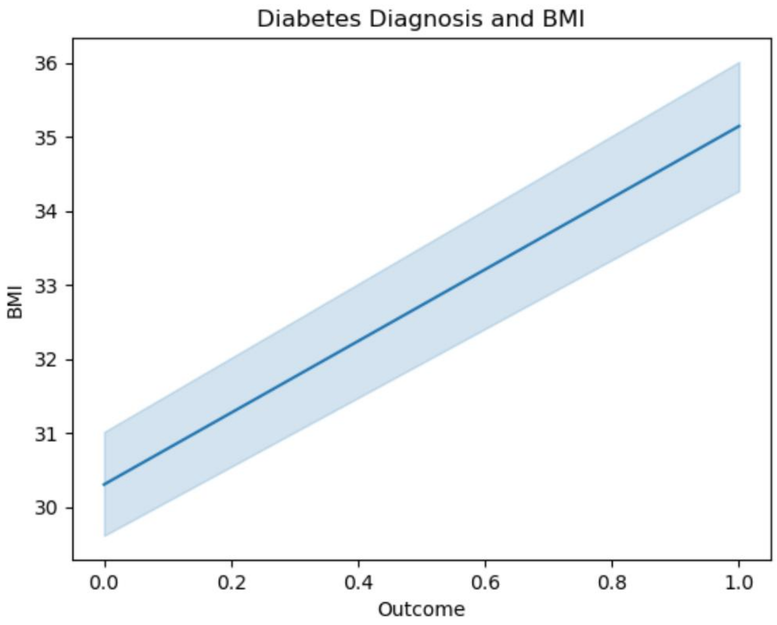
```
# Checking if this dataset has a large amount of women with a family history of type 2 diabetes
# The histogram shows most of the women do not have family history of diabetes which shows this specific dataset is still
# good to predict diabetes without that family influence.

sns.histplot(x='DiabetesPedigreeFunction', data=diabetes_df)
plt.title('Diabetes Pedigree Function Counts')
plt.show()
```



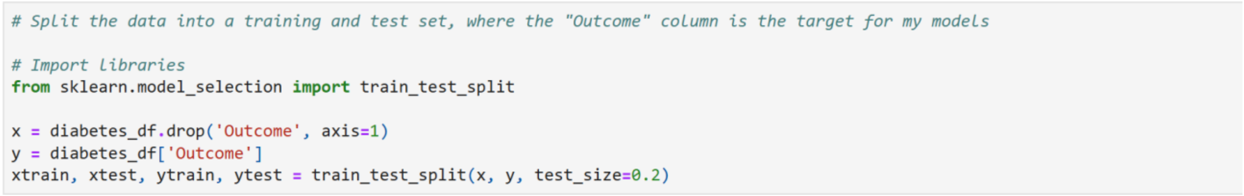
The last visualization that helps my analysis is whether body mass index has a correlation with being diabetic. I created a line graph to compare the two variables of BMI and outcome. BMI is an acronym for body mass index, a measure of body fat based on height and weight that applies to adult men and women. The line graph showed that the lower the BMI, the outcome was closer to 0 which is the non-diabetic. The diabetic outcome, closer to 1, the line graph showed a steep positive slope correlated to greater BMI. Previous research from outside sources does indicate that overweight people are more prone to diabetes. A healthy BMI is all dependent of sex, age, height, and weight so I cannot say what is considered healthy without knowing the women's heights which is not indicated in the dataset. The line graph does clearly show that regardless of what is considered a healthy BMI, the outcome of diabetes is more likely when the BMI is greater. Knowing whether you have high BMI or not can help one be weary of whether you are more prone to getting diabetes and make necessary changes to a current lifestyle.



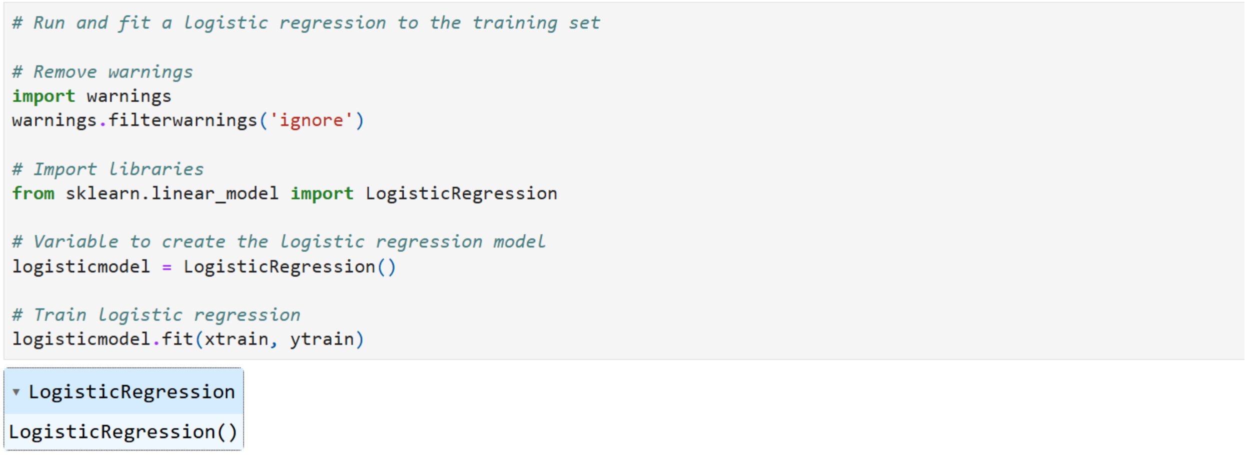
```
# Comparing BMI and outcome for diabetes diagnosis

sns.lineplot(x='Outcome', y='BMI', data=diabetes_df)
plt.title('Diabetes Diagnosis and BMI')
plt.xlabel('Outcome')
plt.ylabel('BMI')
plt.show()
```



My model building will go over 2 types of models. Since I am working with a binary categorical variable as my target, the best suited models will be logistic regression and random forest classifiers. The metrics I used are precision, recall, and f1-score. I chose these metrics because they are best used to evaluate the performance of binary classification models such as these. Precision is the measure of how many of the positive predictions made are correct which are equivalent to true positives. Recall is a measure of how many of the positive cases the classifier correctly predicted compared to all the positive cases in the dataset. F1-score is a measure combining both precision and recall, the harmonic mean of the two. First, I will be splitting the data into test and training sets to fit both models by dropping the “outcome variable”. The first model will be the logistic regression and what the metrics evaluated using the accuracy score function and a classification report.

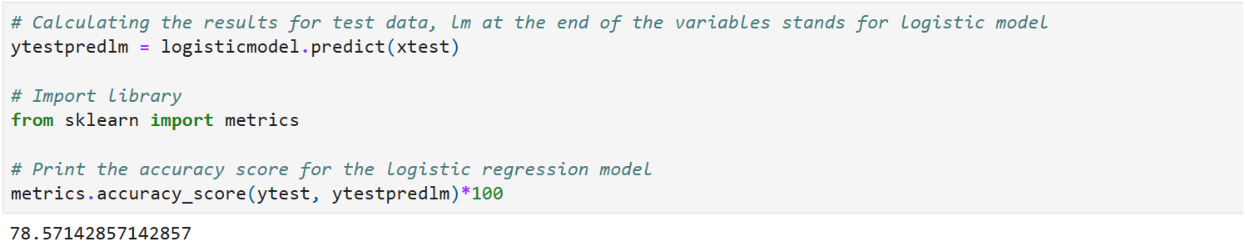


```python
# Split the data into a training and test set, where the "Outcome" column is the target for my models

# Import libraries
from sklearn.model_selection import train_test_split

x = diabetes_df.drop('Outcome', axis=1)
y = diabetes_df['Outcome']
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2)
```



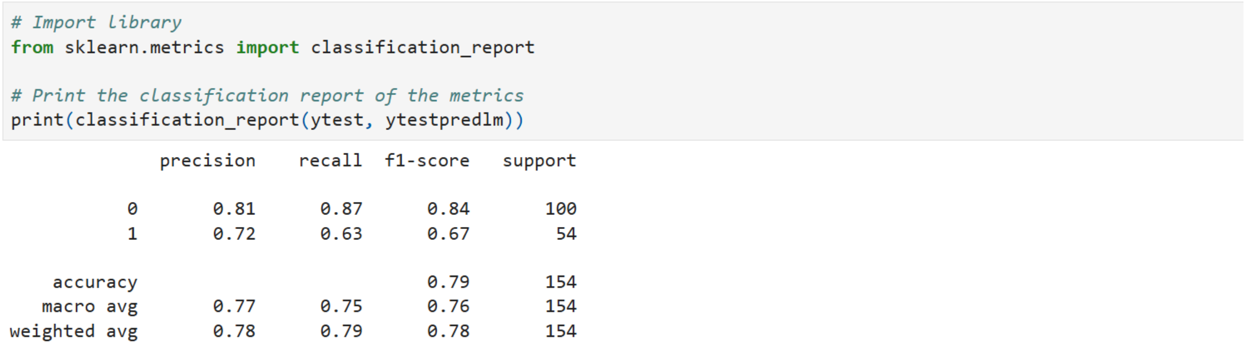
```python
# Run and fit a logistic regression to the training set

# Remove warnings
import warnings
warnings.filterwarnings('ignore')

# Import libraries
from sklearn.linear_model import LogisticRegression

# Variable to create the logistic regression model
logisticmodel = LogisticRegression()

# Train logistic regression
logisticmodel.fit(xtrain, ytrain)
```

```
▾ LogisticRegression
LogisticRegression()
```



```python
# Calculating the results for test data, lm at the end of the variables stands for logistic model
ytestpredlm = logisticmodel.predict(xtest)

# Import library
from sklearn import metrics

# Print the accuracy score for the logistic regression model
metrics.accuracy_score(ytest, ytestpredlm)*100
```

```
78.57142857142857
```

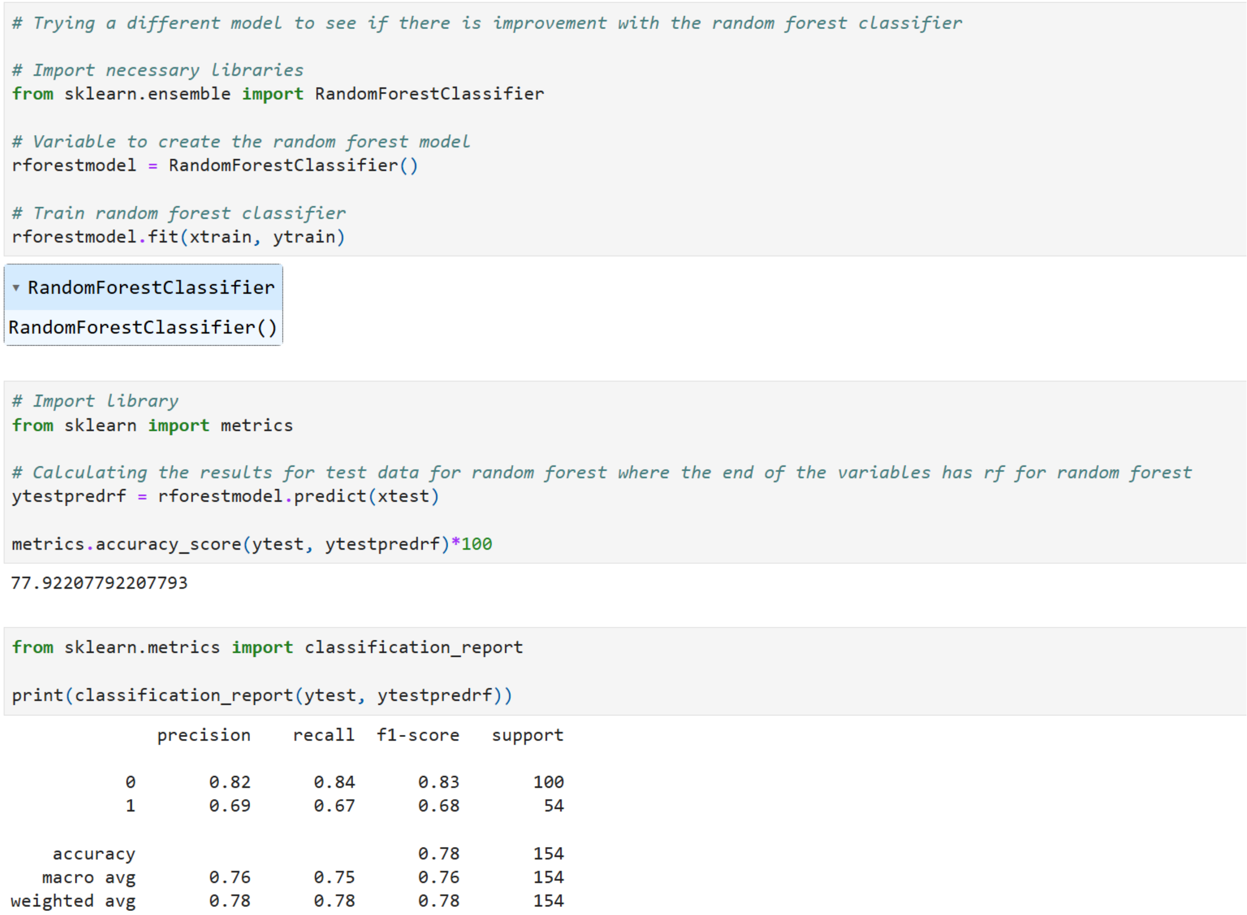


```python
# Import library
from sklearn.metrics import classification_report

# Print the classification report of the metrics
print(classification_report(ytest, ytestpredlm))
```

```
              precision    recall  f1-score   support

           0       0.81      0.87      0.84       100
           1       0.72      0.63      0.67        54

    accuracy                           0.79       154
   macro avg       0.77      0.75      0.76       154
weighted avg       0.78      0.79      0.78       154
```

Results here show the accuracy score is 78.57% with the accuracy score function. For the logistic regression model, the precision for the non-diabetic is 0.81, recall is 0.87, and the f1-score is 0.84. For the diabetic, precision score is 0.72, recall is 0.63, and f1-score is 0.67. These scores indicate that out of all the women that the model predicted would be diabetic, 72% were. Out of all the women that did get diabetes, the model predicted this outcome correctly for 63% of those women. Overall, my model is close to average at predicting whether women will get diabetes with a f1-score of 67%. The model is better at predicting the non-diabetics with precision of 81%. Out of all the non-diabetic women, the model predicted their outcome correctly for 87% of those women. The f1-score is also better for non-diabetic women of 84%.

Now the analysis for the random forest classifier and evaluation of the same metrics.



```
# Trying a different model to see if there is improvement with the random forest classifier

# Import necessary libraries
from sklearn.ensemble import RandomForestClassifier

# Variable to create the random forest model
rforestmodel = RandomForestClassifier()

# Train random forest classifier
rforestmodel.fit(xtrain, ytrain)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
# Import library
from sklearn import metrics

# Calculating the results for test data for random forest where the end of the variables has rf for random forest
ytestpredrf = rforestmodel.predict(xtest)

metrics.accuracy_score(ytest, ytestpredrf)*100
```

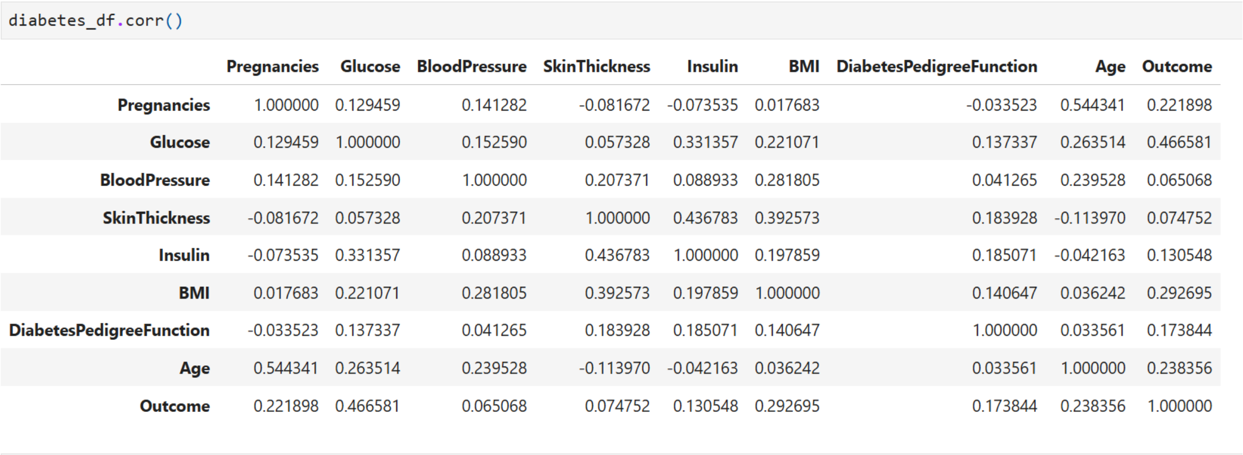
```
77.92207792207793
```

```
from sklearn.metrics import classification_report

print(classification_report(ytest, ytestpredrf))
```

```
              precision    recall  f1-score   support

           0       0.82      0.84      0.83       100
           1       0.69      0.67      0.68        54

    accuracy                           0.78       154
   macro avg       0.76      0.75      0.76       154
weighted avg       0.78      0.78      0.78       154
```

These results show the accuracy score is 77.92% with the accuracy score function. For the random forest classifier model, the precision for the non-diabetic is 0.82, recall is 0.84, and the f1-score is 0.83. For the diabetic, precision score is 0.69, recall is 0.67, and f1-score is 0.68. This indicates that out of all the women that the model predicted would be diabetic, 69% were. Out of all the women that did get diabetes, the model predicted this outcome correctly for 67% of those women. Overall, this model is also close to average at predicting whether women will get diabetes with a f1-score of 68%. The model is also better at predicting the non-diabetics with precision of 82%. Out of all the non-diabetic women, the model predicted their outcome correctly for 84% of those women. The f1-score is also better for non-diabetic women of 83%.

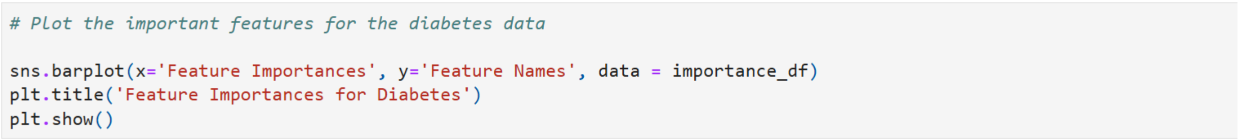
Between the logistic regression and random forest classifier, they are very close in results for all the scores. Next, I created a correlation matrix and a bar chart with the top features from the dataset to help further the analysis.



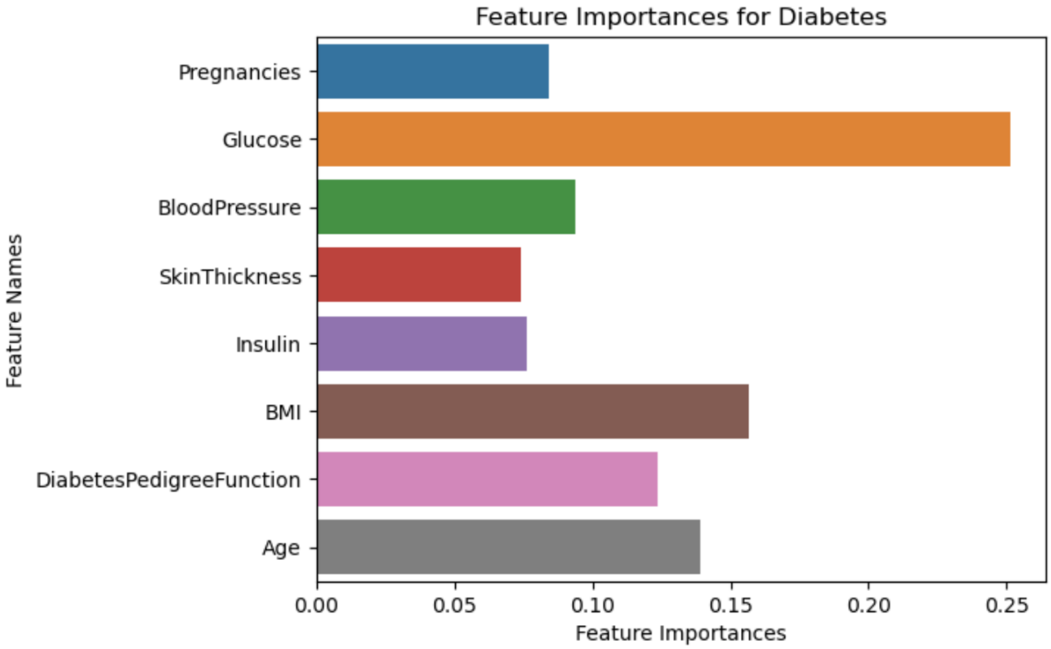
```
diabetes_df.corr()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017683	-0.033523	0.544341	0.221898
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221071	0.137337	0.263514	0.466581
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281805	0.041265	0.239528	0.065068
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392573	0.183928	-0.113970	0.074752
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197859	0.185071	-0.042163	0.130548
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000000	0.140647	0.036242	0.292695
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140647	1.000000	0.033561	0.173844
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036242	0.033561	1.000000	0.238356
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.292695	0.173844	0.238356	1.000000

The correlation matrix shows that the top 3 features correlated with the outcome (diagnosis) variable are the glucose levels, then body mass index, followed by age. Glucose is expected to be the top reason for diagnosis as high glucose levels are what causes a positive diagnosis. The bar chart I showed earlier in my analysis shows what those average glucose levels were for diabetics and non-diabetics. BMI as the second top feature was shown also in the line graph from earlier of BMI increasing with diabetes diagnosis. Age is expected as well, since the older you are, the more likely you are to have diabetes. The bar chart created below with the top features for diabetes reinforces the correlation matrix and it is easier to interpret visually for external audiences. It shows the least important in diagnosis is skin thickness, insulin, and pregnancies.



```
# Plot the important features for the diabetes data

sns.barplot(x='Feature Importances', y='Feature Names', data = importance_df)
plt.title('Feature Importances for Diabetes')
plt.show()
```



Overall, the logistic regression model shows to be a good prediction model for diabetic and non-diabetic women. Although the accuracy score is not in the 90-99% range, I feel confident in the model since I tried other ways to bring the accuracy percentage up but with no success. I tried a pipeline with a minmax scaler and a KNN-classifier to see if that would be a better model, but the scores did not improve more than the logistic regression or random forest classifier. I also tried a 5-fold cross-validation, hyperparameter tuning and a pipeline with the logistic regression and random forest classifier, yet the accuracy scores still did not improve. With that being said, I think this model is still a great start to help prevent diabetes early on and predict whether one is likely to have it or not based on the features. The model is limited in the sense it can't be the only reliable source to predict diabetes, but it can help. It is crucial to detect diabetes early on to prevent it from getting worse since once diagnosed, it is irreversible. The model shows that it is good at predicting the outcome of non-diabetics too, which I was not expecting. It is better to predict non-diabetics over diabetics, but the visualizations also help to understand the important features and what has caused diabetes in women.

Conclusion

Recommendations I can give based off my analysis are glucose levels and what the averages were for the diabetics and non-diabetics. I can recommend that BMI is correlated with a diabetes diagnosis, and that the models can predict whether a woman is diabetic or non-diabetic with above average accuracy. I learned that different model types alter the results and sometimes hyperparameter tuning or scaling will not improve scores. It is all dependent on the data and what works best. In my case, the logistic regression and random forest classifier were very close in results, but the logistic regression performed a little bit better. I learned that pregnancies are not that impactful when diagnosing women, and age is more important than I assumed. I would say both models are ready for deployment and testing. The f1-score especially for the non-diabetics is 84% for the logistic regression and 83% for the random forest classifier. The diabetic predictions were lower but still all above 65%. There still needs to be work done when it comes to diabetes all together. My model is just scratching the surface at model predictions, and the dataset only covers a small sample. Ethically, I need to consider the women’s data and their privacy since their medical and demographic information is included in the dataset/models. For the presentation, the statistics from outside sources must be correct to not mislead and spread inaccurate information. If the project was live in production, the models can raise ethical implications with pharmaceutical companies and diabetes medications. If diabetes becomes less apparent in Americans, it will create less revenue for those companies which cannot be mitigated. The goal does stand, however, that health is the upmost importance in one’s life.

References

Charles, S. (2022, September 27.). *Signs of type 2 diabetes*. Verywellhealth. https://www.verywellhealth.com/type-2-diabetes-in-women-6674445

Kishore, P. (2023, October). *Diabetes prediction dataset*. Kaggle. https://www.kaggle.com/datasets/pentakrishnakishore/diabetes-csv?select=diabetes.csv

Pruthi, S. (2023, September 15). *Diabetes.* Mayo clinic. https://www.mayoclinic.org/diseases-conditions/diabetes/symptoms-causes/syc-20371444#Complications

Wergin, A. (2023, September 6). *Diabetes: Prevention and warning signs.* Mayo clinic. Diabetes: Prevention & warning signs - Mayo Clinic Health System