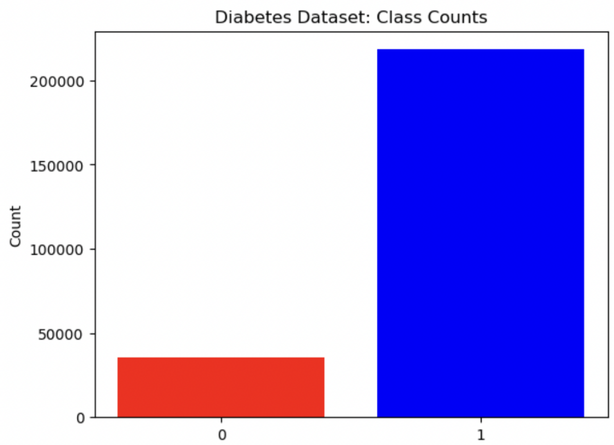
REPORT : HW3 FML

Full Name	Net ID	Class	Professor
Aya El Mir	ae2195	FML	Pascal Wallisch

Each answer should contain these elements:

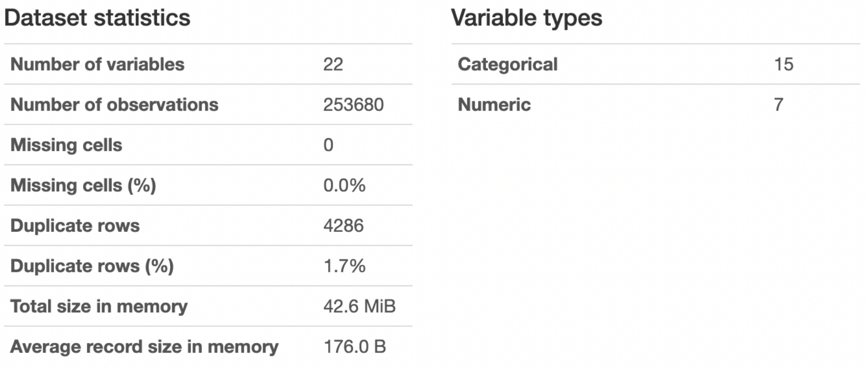
1. **A brief statement (~paragraph) of what was done to answer the question (narratively explaining what you did in code to answer the question, at a high level).**
2. **A brief statement (~paragraph) as to why this was done (why the question was answered in this way, not by doing something else. Some kind of rationale as to why you did x and not y or z to answer the question – why is what you did a suitable approach?).**
3. **A brief statement (~paragraph) as to what was found. This should be as objective and specific as possible – just the results/facts. Do make sure to include numbers and a figure (=a graph or plot) in your statement, to substantiate and illustrate it, respectively.**
4. **A brief statement (~paragraph) as to what you think the findings mean. This is your interpretation of your findings and should answer the original question.**

Data Preprocessing:



Class imbalance:
It is important to consider class imbalance when training machine learning (ML) models because the models can be biased towards the majority class. In a highly imbalanced dataset, the minority class may not have enough representative examples to be learned properly. This can lead to a model that predicts the majority class accurately but performs poorly on the minority class. As we see in this plot for our diabetes dataset, we can clearly see the class imbalance with a majority of non diabetic patients. Therefore, to account for that I am always passing class_weight = 'balanced' as an argument for the models I am running in this hw. Therefore, the algorithm will give more weight to this class during training to help mitigate the impact of class imbalance on the model's performance.

Variable types:
Considering the various nature of variables (Categorical and numeric) we have in our dataset, it is important to consider that when assessing the performance of models to predict diabetes. Specifically for the predictor Age Bracket, which is given as different bins, to be more accurate in my analysis I divided it into 13 sub predictors as show in the code submitted.



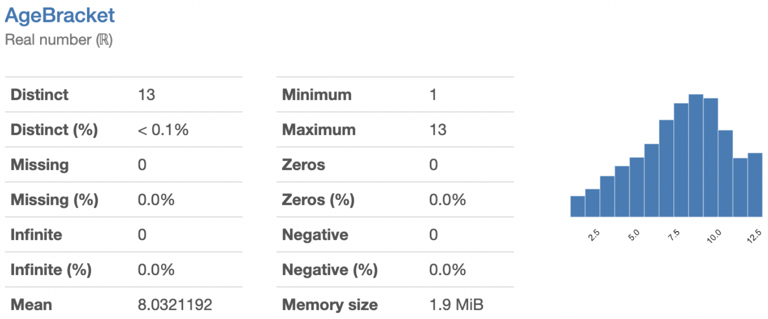
Dataset statistics

Number of variables	22
Number of observations	253680
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	4286
Duplicate rows (%)	1.7%
Total size in memory	42.6 MiB
Average record size in memory	176.0 B

Variable types

Categorical	15
Numeric	7

–



AgeBracket
Real number (ℝ)

Distinct	13
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	8.0321192

Minimum	1
Maximum	13
Zeros	0
Zeros (%)	0.0%
Negative	0
Negative (%)	0.0%
Memory size	1.9 MiB

QUESTION 1: Build a logistic regression model. Doing so: What is the best predictor of diabetes and what is the AUC of this model?

1. What was done to answer the question?

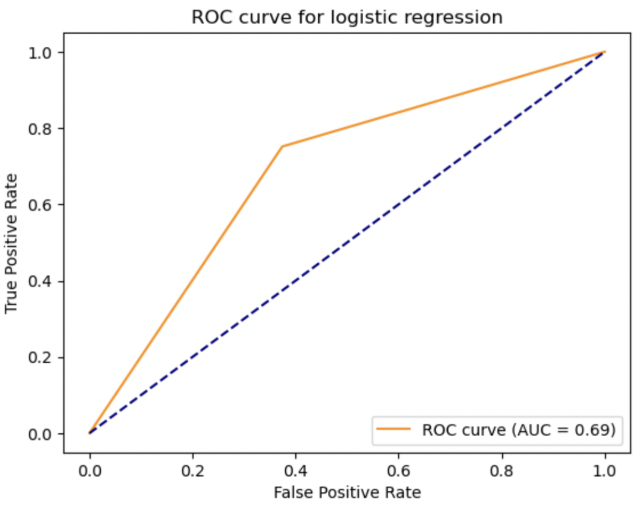
In order to find the best predictor for diabetes using logistic regression I utilized the approach of checking which predictor out of the 21 predictors we have gives the highest increase in AUC score. More specifically, the code computes the AUC score for each of the predictors while storing the results in an array and computing the difference between the sorted predictor values.

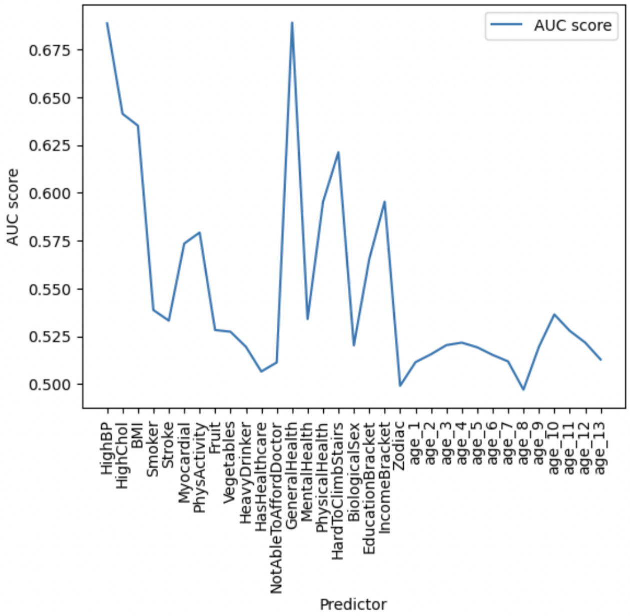
I first start by creating a dictionary to store the AUC stores for each predictor, then inside a for loop that goes through all the columns of X, I each time select the current predictor and slit the data. Then fit a logistic regression model. Considering how our data is strongly unbalanced (way less nondiabetes cases), I am passing the parameter class_weight = 'balanced' to the Logistic Regression function to make the model pay more attention to the diabetes cases. Then, I predict the testing set and calculate the evaluation metrics which I append to the metrics dictionary. Outside the for loop, in order to visualize our findings, I am plotting the learning curve for each predictor, then sort the predictors by their AUC score in descending order, and finally find the best predictor that gives the highest increase in AUC score.

2. Why this was done?

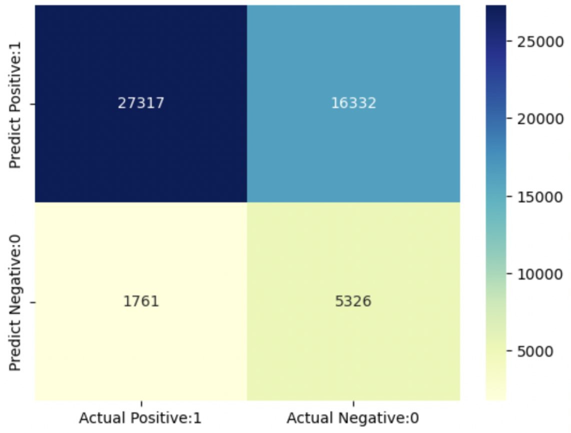
AUC is a common evaluation metric used in binary classification problems like diabetes prediction. It provides an overall measure of the model's ability to discriminate between positive and negative examples, taking into account both sensitivity (true positive rate) and specificity (true negative rate).

In the context of diabetes prediction, a higher AUC would indicate that the model is better at correctly identifying patients with diabetes and patients without diabetes. Indeed, we are more concerned with identifying true positives (people who have diabetes and are correctly identified as having diabetes) and minimizing false negatives (people who have diabetes but are incorrectly identified as not having diabetes). Therefore, utilizing the AUC score as a metric to compare the performance of the logistic regression model across the 21 different predictors is the best way to predict the best predictor in the context of this question and dataset.





```
The best predictor is: HighBP
```



3. What was found?

As shown in the Jupyter notebook attached with the submission, I was able to find that the predictor that gives the highest increase in AUC score is **HighBP** with an AUC score of 0.6886751071423851. In order to clearly visualize my findings I also plotted the learning curve for each predictor.

4. What do you think these findings mean?

In the plotted ROC curve, we see that HighBP is indeed the predictor with the highest AUC score (0.6886751071423851). An AUC score of 0.6886751071423851 indicates that the model's ability to discriminate between positive and negative instances of diabetes is only slightly better than random guessing. Indeed, this might be deemed acceptable if the model considering the model has a high sensitivity, which can be assessed by examining the ROC curve or the confusion matrix of the model. We can examine the shape of its ROC curve, the curve is close to the upper left corner, it suggests that the model has a high true positive rate, and hence, high sensitivity.

We note from the plot that General Health also has a very high AUC score showing its importance in predicting diabetes as well. Additionally, the plot allows us to get a good visualization of each predictor's performance for predicting Diabetes; notably, for instance, we are also able to see that Zodiac is the least good predictor for Diabetes which indeed makes sense and is shown in our findings as well. I was also able to visualize the confusion matrix and the ROC curve in order to get the general performance of the ROC curve which consists of the TPR and FPR at various threshold levels. The ROC curve helps us to choose a threshold level that balances sensitivity and specificity for a particular context. By looking at the ROC curve we find that the ROC AUC of our model approaches 1, thus, we can conclude that our classifier does a good job of classifying diabetes. However, compared to other models we will get the chance to analyze further in our analysis, the performance of logistic regression is not really the most optimal compared to other models.

QUESTION 2: Build a SVM. Doing so: What is the best predictor of diabetes and what is the AUC of this model?

1. What was done to answer this question?

The code first split the data into training and testing sets, trained an SVM classifier on the training data, predicted the labels for the test data, computed the accuracy of the model, identified the best predictor of diabetes, and calculated the AUC score.

First, The dataset was split into training and testing sets using the train_test_split function from sklearn.model_selection. The testing set was set to be 20% of the data, and a random state of 42 was used to ensure reproducibility. Then, an SVM classifier was trained using the linear support vector classifier (LinearSVC) from sklearn.svm. The fit method was used to train the SVM on the training data. Following that, the labels for the test data were predicted using the predict method of the SVM classifier. The accuracy of the SVM model was then computed using the accuracy_score function from sklearn.metrics. The decision scores and feature names were obtained for all samples using the decision_function method of the SVM classifier. The absolute values of the decision scores were sorted in descending order to identify the best predictor. The index of the best predictor (feature) was found using np.argmax(np.abs(svm.coef_)) and the name of the best predictor was obtained using the feature_names list. The area under the receiver operating characteristic curve (AUC) was computed using the roc_auc_score function from sklearn.metrics.

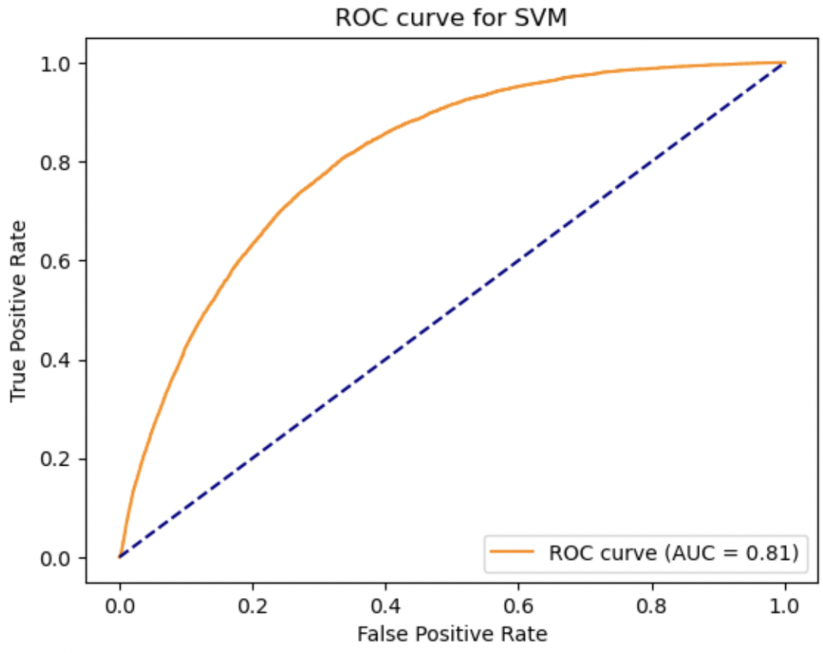
2. Why this was done?

I chose to use a Linear Support Vector Machine (LinearSVM) because it is a commonly used algorithm for binary classification problems, such as predicting the presence or absence of diabetes. LinearSVM is a type of SVM that works well when there is a clear margin of separation between the classes, which might is with the diabetes dataset.
Additionally, as mentioned in the previous question; the AUC score because it is a widely used metric for evaluating the performance of binary classification models, especially in cases where the classes are imbalanced. The AUC score is a measure of the overall quality of a binary classifier that takes into account both sensitivity and specificity, which can be useful in medical applications such as predicting the presence or absence of a disease like diabetes.

3. What was found?

We found that the best predictor is Feature 22 (recall the data processing which increased the number of columns in the feature matrix X) corresponding to **HighBP** predictor. The best predictor thus so far for both logistic regression and SVM is the same. Indeed, the performance is different across the 2 models. Note that a high AUC score indicates good performance, while a score of 0.5 suggests that the model is no better than random guessing.

For this model, an AUC score of 0.8059388380964961 was found, thus, we can say that the Linear SVM model has better discriminatory power than the logistic regression model. As for the last question, I also plotted the ROC curve.



4. What do you think these findings mean?

An AUC score of 0.8059388380964961 suggests that the Linear SVM model has moderate to good discriminatory power in distinguishing between positive and negative instances of diabetes.
A linear SVM model with an AUC score of 0.8059388380964961 performs better than a logistic regression model with an AUC score of 0.6886751071423851. This suggests that the SVM model is better at distinguishing between patients who have diabetes and those who do not. A higher AUC score for the SVM model indicates that it has a higher true positive rate and a lower false positive rate compared to the logistic regression model. Therefore, the SVM model is more accurate in identifying patients with diabetes and minimizing the misclassification of patients without diabetes.
Regarding the interpretability of the ROC curve, we can clearly see the better performance from the curve itself. Indeed, the curve for a linear SVM model is closer to the top-left corner of the graph, indicating a higher true positive rate and a lower false positive rate. The curve is further away from the diagonal line, which represents random guessing, and is more concave upward. Therefore, the SVM model has a higher ability to correctly classify positive instances (patients with diabetes) while minimizing the false positive rate (patients without diabetes classified as having diabetes). In general, a higher AUC score corresponds to a more accurate and effective model, with the curve approaching the top-left corner of the graph.

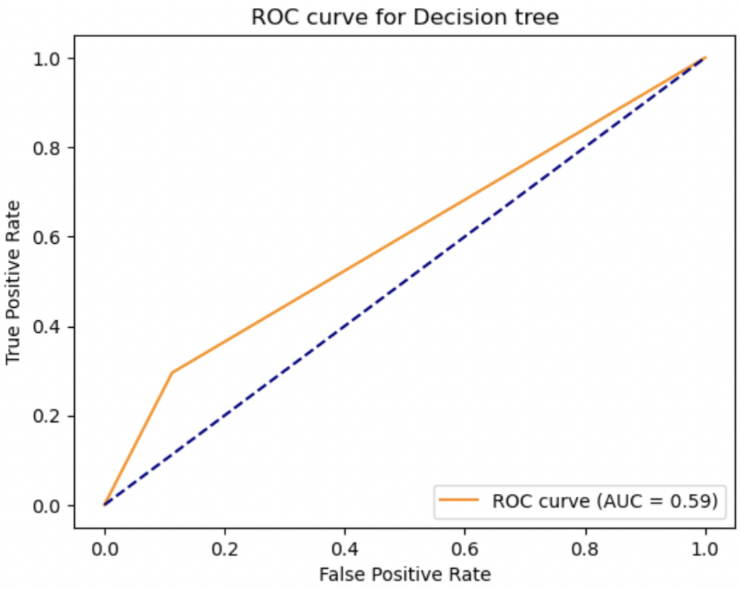
QUESTION 3: Use a single, individual decision tree. Doing so: What is the best predictor of diabetes and what is the AUC of this model?

1. What was done to answer this question?

In the provided code, the goal was to build a decision tree model to predict diabetes and identify the best predictor of diabetes. The data was first split into training and testing sets. The decision tree classifier was trained using the training set and the class_weight parameter was set to 'balanced' to address the class imbalance issue. The model was used to predict the labels for the test data, and the accuracy of the model was calculated. The AUC-ROC score was also computed to evaluate the performance of the model. Finally, the feature with the highest feature_importance_ was identified as the best predictor of diabetes, and its name was printed along with the AUC score.

2. Why this was done?

The aim of this analysis was to identify the best predictor of diabetes using a single decision tree classifier and to determine the AUC of the model. A single decision tree was chosen as it is a simple and interpretable model that can provide insight into the most important features in predicting diabetes. The decision tree model was trained using the balanced class weight to account for the imbalance in the dataset. The model was evaluated using the AUC-ROC score, which is a suitable metric for evaluating binary classification models like decision trees. The feature with the highest feature importance was identified as the best predictor of diabetes, and the AUC of the model was calculated to determine the performance of the model. By answering this question using a single decision tree, we were able to gain insights into the most important predictors of diabetes and evaluate the performance of the model in a simple and interpretable way.



3. What was found?

The best predictor was found to be **HighBP**. The decision tree model was trained on the diabetes dataset and evaluated using a single decision tree. The accuracy of the model on the test data was found to be 0.8648, meaning that the model correctly predicted the presence or absence of diabetes in 86.48% of cases. The best predictor of diabetes was found to be HighBP, which had the highest feature importance score among all the features in the dataset. The AUC of the decision tree model was found to be 0.5914, indicating that the model's ability to distinguish between positive and negative classes is not very good. This suggests that the decision tree model is not very effective in predicting the presence or absence of diabetes in patients. The ROC curve also suggests the same bad performance of the model. Indeed, the curve hugs the diagonal line, indicating that the model is not much better than random at predicting the target variable.

4. What do you think these findings mean?

The decision tree model achieved an accuracy of 0.865 and an AUC of 0.591, which is lower than the performance of the logistic regression and SVM models. These findings make sense considering the nature of the algorithm as decision trees tend to overfit the training data, which means that the model may have learned the noise in the data rather than the underlying patterns. This can lead to poor generalization to new data and lower model performance. To mitigate this, we can use ensemble methods such as bagging, random forests, or boosting, which combine multiple decision trees to reduce overfitting. Indeed, the following questions will show a better performance for Random Forest and AdaBoost. Additionally, decision trees are limited in their ability to capture complex non-linear relationships between the features and target variable. Other models such as support vector machines (SVM) might be better suited for such tasks which explain the better performance in the previous questions.

In terms of the ROC curve, a decision tree with an AUC of 0.591 suggests that the model is not able to effectively distinguish between positive and negative cases, and therefore the ROC curve would not be very steep.

QUESTION 4: Build a random forest model. Doing so: What is the best predictor of diabetes and what is the AUC of this model?

1. What was done to answer this question?

For this question, I started by splitting the data into training and testing sets. I then used the RandomForestClassifier from sklearn.ensemble to train a model on the training set with a set of hyperparameters including class_weight='balanced', n_estimators=200, max_samples=0.2, max_features=0.01, bootstrap=True, and criterion='gini'. I then used the trained model to predict labels for the test data and computed the accuracy of the model by dividing the number of correct predictions by the total number of predictions. Next, I computed the predicted probabilities of the positive class for each test sample using predict_proba method and computed the AUC-ROC score using roc_auc_score from sklearn.metrics. Finally, I found the feature with the highest feature importance using the feature_importances_ attribute of the model, and printed out the name of the best predictor along with the AUC of the model.

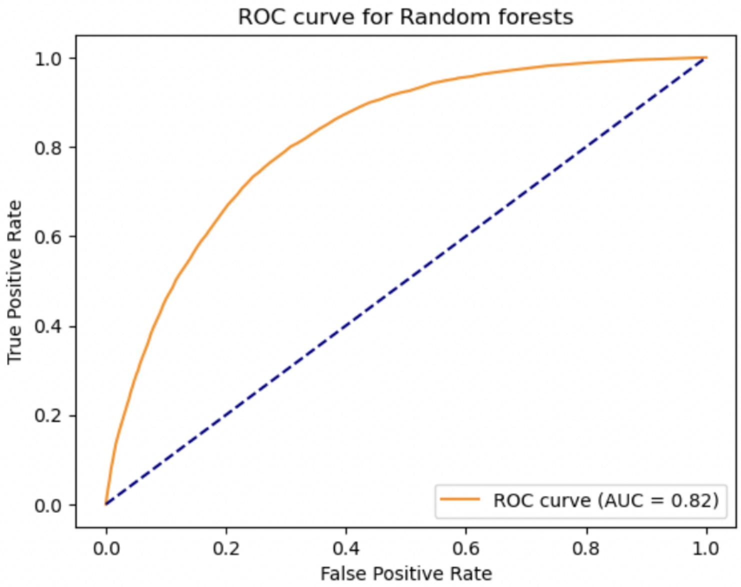
2. Why this was done?

The question asked for the best predictor of diabetes and the AUC of a random forest model. To answer this, I trained a random forest classifier on the diabetes dataset, with 200 decision trees and a maximum of 20% of samples per tree, to prevent overfitting. The maximum features per tree were set to 1% to reduce the correlation between decision trees. The criterion used for splitting nodes was the Gini impurity index, and the class_weight was set to 'balanced' to account for the imbalanced nature of the dataset. I then used the trained model to predict labels for the test data and calculated the accuracy of the predictions. Next, I computed the AUC-ROC score using the predicted probabilities for the test data. Finally, I identified the best predictor of diabetes by finding the index of the feature with the highest feature_importance_ and getting the name of that feature. This approach was chosen because random forest is a popular machine-learning algorithm that can provide a good balance between accuracy and overfitting, and it can handle both continuous and categorical variables which is perfect considering the nature of our algorithm. Furthermore, the feature_importance_ attribute of the random forest model allows us to identify the most important predictor variables.

3. What was found?

Based on the results of the random forest model, it was found that the best predictor of diabetes is **BMI**. The model achieved an accuracy of 0.8640807316304006, indicating that it correctly classified 86.41% of the samples. The AUC of the model was 0.8192154680395389, which is a measure of how well the model is able to distinguish between positive and negative cases. A value of 0.5 represents random guessing, while a value of 1 represents perfect classification. The AUC of 0.8192154680395389 suggests that the model is able to effectively differentiate between positive and negative cases. Overall, these results suggest that the random forest model using BMI as the best predictor may be a useful tool for predicting the likelihood of diabetes in patients.

Looking at the ROC curve, it curves upward toward the top left corner of the plot, indicating that as the true positive rate increases, the false positive rate also increases.



4. What do you think these findings mean?

The ROC curve suggests an increase in the true positive rate greater than the increase in the false positive rate, indicating that the model is performing well. We can see that the ROC curve is well above the random line (diagonal line). A perfect classifier would have an AUC of 1, so an AUC of 0.8192 indicates that the model is reasonably good at distinguishing between positive and negative cases. Contrasting with other models, the Random forest as expected is definitely performing better than the individual; decision tree.

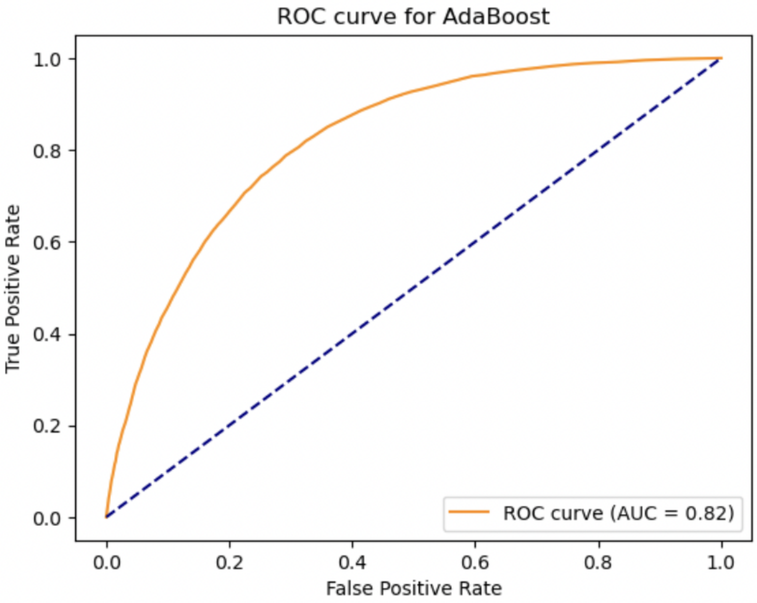
QUESTION 5: Build a model using adaBoost. Doing so: What is the best predictor of diabetes and what is the AUC of this model?

1. What was done to answer this question?

First, the data was split into training and testing sets using the train_test_split function. Then, an AdaBoosted decision tree was created and fit to the training data using AdaBoostClassifier. I tried multiple parameter combinations but ended up using the initial parameter settings used in the lab. After fitting the model, the predict function was used to make predictions on the testing data, and the accuracy was computed by comparing the predicted values to the true values. Next, predict_proba was used to obtain predicted probabilities of class membership for the testing data, and the AUC-ROC score was computed using roc_auc_score by passing the true values and predicted probabilities as arguments. Finally, the best predictor of diabetes was found by getting the index of the feature with the highest feature importance, which was obtained using bdt.feature_importances_.argmax(), and then getting the name of the best predictor using X.columns[best_predictor_idx].

2. Why this was done?

The approach used in this question is suitable as it uses a widely used ensemble method, adaBoost, which is effective in improving the performance of weak classifiers by combining them. Additionally, as mentioned for previous questions, the AUC-ROC score is a commonly used evaluation metric for binary classification models and provides a good measure of the model's ability to distinguish between positive and negative samples. Finally, the feature_importances_ attribute of the adaBoost model provides information about the importance of each feature in the model, which is useful for identifying the best predictor of diabetes.



3. What was found?

The analysis performed using AdaBoost revealed that the best predictor of diabetes is **HighBP**, and the AUC of the model is 0.8299014484567945. This indicates that the model has good predictive power for distinguishing between patients with and without diabetes. The ROC curve looks similar to a concave curve, with a steep rise in the true positive rate and a gradual increase in the false positive rate.

4. What do you think these findings mean?

Overall, these results suggest that AdaBoost is a suitable approach for predicting diabetes based on the available data, and that **HighBP** is the most informative predictor for this task. Indeed, the performance of AdaBoost with its highest AUC score of 0.8299014484567945 has been the best so far compared to the other models we trained in previous question.

EXTRA CREDIT:

A) Which of these 5 models is the best to predict diabetes in this dataset?

According to the last questions, here are the AUC scores previously obtained:

Logistic regression: 0.6886751071423851

SVM: 0.8059388380964961

Individual Decision Tree: 0.5914

Random Forest: 0.8192154680395389

AdaBoost: 0.8299014484567945

Based on the AUC scores above, it appears that the AdaBoost model has the highest AUC score of 0.8299, followed closely by the Random Forest model with an AUC score of 0.8192. The SVM model also has a relatively high AUC score of 0.8059, indicating that it may be a good model for this dataset as well. The Logistic Regression model has a relatively lower AUC score of 0.6887, suggesting that it may not be the best model for this dataset. The individual Decision Tree model has the lowest AUC score of 0.5914, indicating that it may not be a good model for this dataset.

It is important to note that AUC scores alone do not provide a complete evaluation of a model's performance, and other metrics such as accuracy, precision, and recall could also be considered for a more holistic view of the peformance.

Indeed, in our case of predicting diabetes, it is important to correctly identify as many positive cases (patients with diabetes) as possible, while also minimizing the number of false positives (non-diabetic patients incorrectly classified as diabetic). However, in an unbalanced dataset as ours where there are more negative cases (non-diabetic patients), a model that simply classifies all cases as negative can achieve a high accuracy but still be useless for practical purposes.

AUC, on the other hand, takes into account both the true positive rate (sensitivity) and the false positive rate (1-specificity) across a range of classification thresholds. This makes it a good metric for evaluating how well a classifier is able to distinguish between the positive and negative cases, regardless of the imbalance in the dataset.

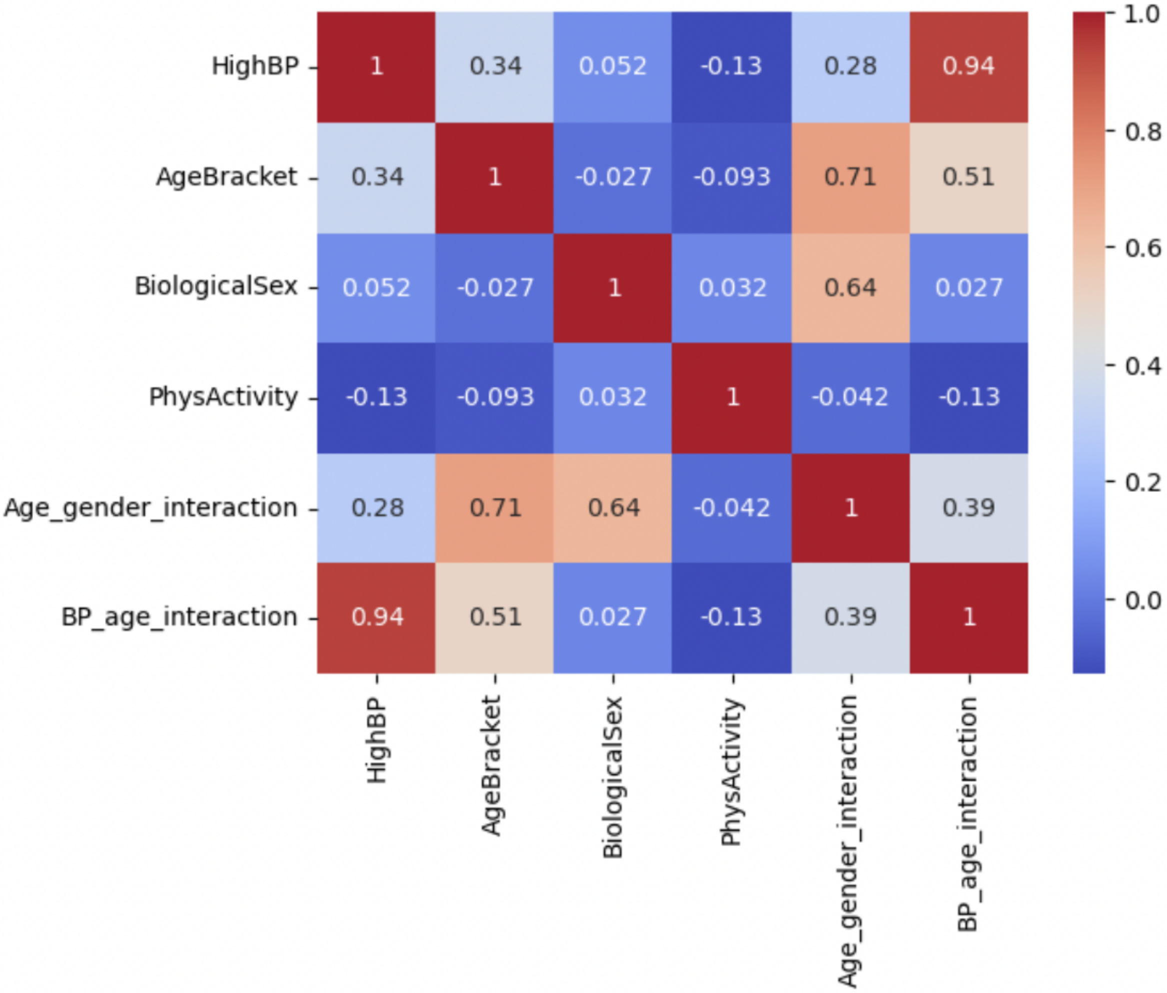
In sum, AUC provides a more comprehensive evaluation of a classifier's performance in unbalanced datasets, and is therefore a good metric to use when evaluating the performance of classifiers in predicting diabetes.

Lastly, the choice of model should also be based on the specific problem and dataset, indeed, having looking for answering a specific question beyond presence or absence of diabetes, as well as a different dataset may lead to very different results and conclusions. Additionally, different models may perform differently depending on the hyperparameters chosen.

B) Tell us something interesting about this dataset that is not already covered by the questions above and that is not obvious.

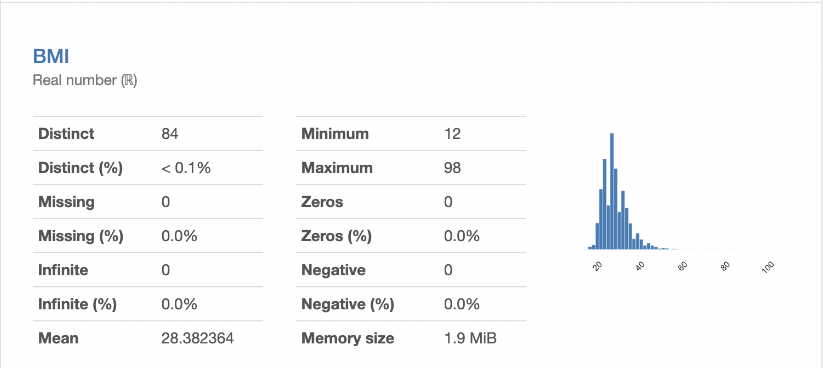
For this question, I decided to explore the potential interactions between the variables in the dataset.

Based on the logistic regression results shown in the code for extra credit in the Jupyter notebook submitted, it appears that HighBP, AgeBracket, BiologicalSex, PhysActivity, Age_gender_interaction, and BP_age_interaction are all significant predictors of diabetes. Specifically, a one-unit increase in HighBP is associated with an increase in the log-odds of having diabetes by 1.9551, a one-unit increase in AgeBracket is associated with a decrease in the log-odds of having diabetes by 0.2337, a one-unit increase in BiologicalSex (where 0=female, 1=male) is associated with a decrease in the log-odds of having diabetes by 2.2893, a one-unit increase in PhysActivity is associated with a decrease in the log-odds of having diabetes by 0.6147, a one-unit increase in Age_gender_interaction is associated with an increase in the log-odds of having diabetes by 0.2549, and a one-unit increase in BP_age_interaction is associated with a decrease in the log-odds of having diabetes by 0.0669. This suggests that these variables play an important role in predicting the presence or absence of diabetes in the dataset. Interestingly, the Age_gender_interaction variable appears to be a significant predictor of diabetes, which suggests that the relationship between age and diabetes may differ depending on gender. Additionally, the BP_age_interaction variable is also a significant predictor, indicating that the relationship between blood pressure and diabetes may differ depending on age. Overall, these findings suggest that there may be complex interactions between variables in this dataset that could be further explored to gain a better understanding of the factors contributing to diabetes.



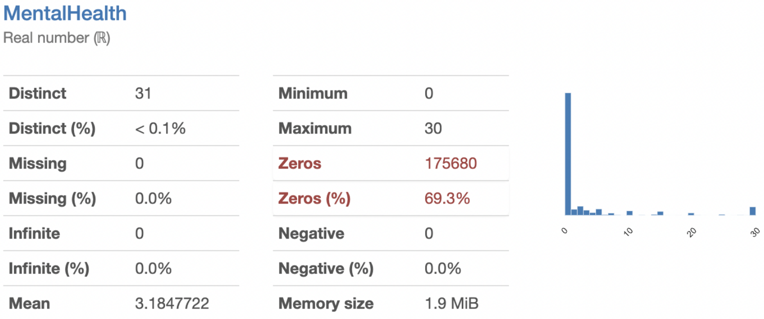
Indeed, after seeing theses results and being intrigued by the importance of gender in this dataset, I decided to evaluate the logistic regression model accuracy for each gender separately. The difference found between the two accuracies is small with Accuracy for female: 85.31% and Accuracy for male: 87.03%. However, after giving it some thought, I realized how it is not really beneficial to do such analysis on gender for such a dataset as we cannot really say whether the model is biased or the data is biased (inferring how nature is biased which is not really reasonable). In fact, models cannot really be biased if it picks up on reality. It was interesting for me to think about such questions while working with the given data.

Another interesting aspect I analyzed is the distribution of the data, or more specifically the distribution of the predictors. At the start of my work on the assignment, to get an overview of the data, I run the Profile Report on the extracted data, and was able to see interesting patterns of the distribution of some variables.


BMI
Real number (ℝ)

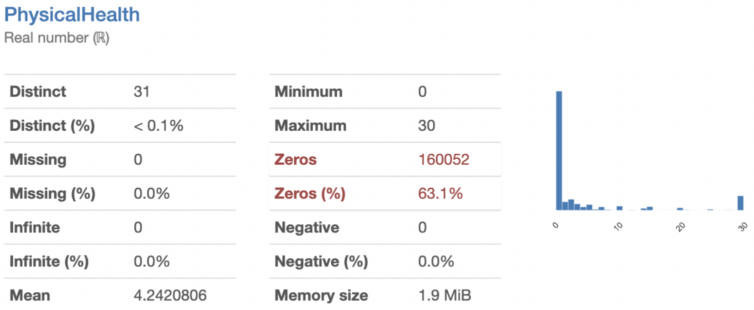
Distinct	84	Minimum	12
Distinct (%)	< 0.1%	Maximum	98
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	28.382364	Memory size	1.9 MiB

For instance, we can see that BMI has a more normal distribution, however, looking at some other predictors as Mental Health the distribution is not normal. Indeed, as a lot of models assume the data to be normal, it would be interesting to investigate such patterns in the data and how we can deal with them.



MentalHealth
Real number (ℝ)

Distinct	31	Minimum	0
Distinct (%)	< 0.1%	Maximum	30
Missing	0	Zeros	175680
Missing (%)	0.0%	Zeros (%)	69.3%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	3.1847722	Memory size	1.9 MiB



PhysicalHealth
Real number (ℝ)

Distinct	31	Minimum	0
Distinct (%)	< 0.1%	Maximum	30
Missing	0	Zeros	160052
Missing (%)	0.0%	Zeros (%)	63.1%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	4.2420806	Memory size	1.9 MiB

In most cases, machine learning models assume that the predictor variables are normally distributed. This assumption is made because many statistical methods, such as linear regression, rely on the normal distribution for their underlying mathematical calculations. However, if a predictor variable does not have a normal distribution, it can still be used in a model, but the model's performance may be affected.

If a predictor variable is not normally distributed, it may be necessary to transform the data to achieve a normal distribution. There are various transformations that can be used, such as logarithmic or square root transformations. Alternatively, non-parametric methods can be used that do not require the normal distribution assumption, such as decision trees or support vector machines.

The impact of non-normality on model performance will depend on the severity of the non-normality and the specific machine learning algorithm being used. In general, if the non-normality is mild, it may not have a significant impact on the model performance. However, if the non-normality is severe, it may lead to biased estimates and reduced prediction accuracy. Therefore, it is important to assess the distribution of predictor variables and consider appropriate transformations or alternative modeling approaches if non-normality is present.