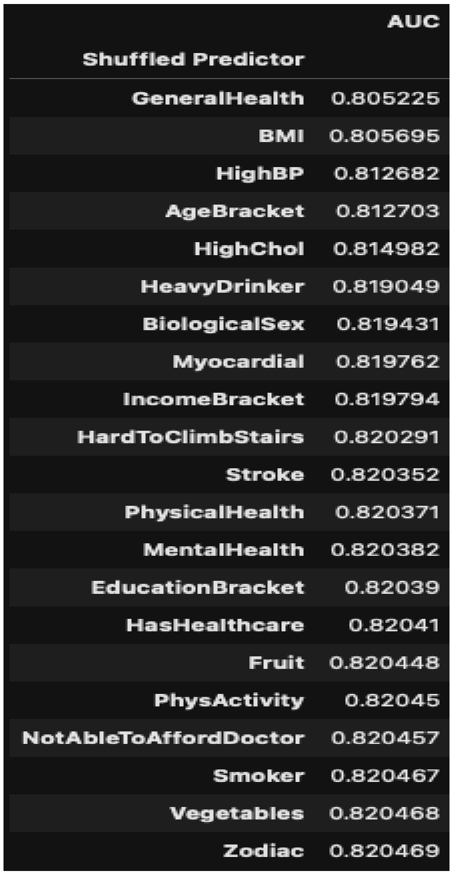
Preliminary: this will serve to reduce repeating information.

For all models in this assignment, no extra data processing is done (except for splitting X and y) because there is no missing data, and each model uses all the predictors. To compute the AUC score, I use the mean of cross_val_score(X, y, scoring = 'roc_auc') because the average of 5 tests is always better than 1.

To find the best predictor of a model, I used the hint provided in the spec sheet: shuffle each predictor's labels and record the auc of the model trained on the new data, and see which predictor dropped the AUC the most (Permutation feature importance). I did this because when a predictor's label is shuffled, it becomes a useless feature because it is just a list of random numbers. When we turn the best predictor into a list of random numbers, the AUC of the model will drop the most.

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

   Because sklearn gives me "not converging" warning, I set max_iter to 800 because the model needs more iterations to converge, which resolves such a problem. To find the best predictor, I used the hint, shuffling each predictor and recording the AUC of the model trained on the new data. The result is on the right. As we can see, the AUC is the lowest when we shuffle GeneralHealth, which means that GeneralHealth has the most effect on the model, suggesting that it is the best predictor. I also examined the betas of each predictor, and HighBP had the highest beta. The reason why HighBP is not the best predictor may be that GeneralHealth has a larger range of value, which HighBP is 0 or 1. GeneralHealth being the most important predictor indicates that we can predict diabetes from a person's overall health better than any other feature in this dataset.

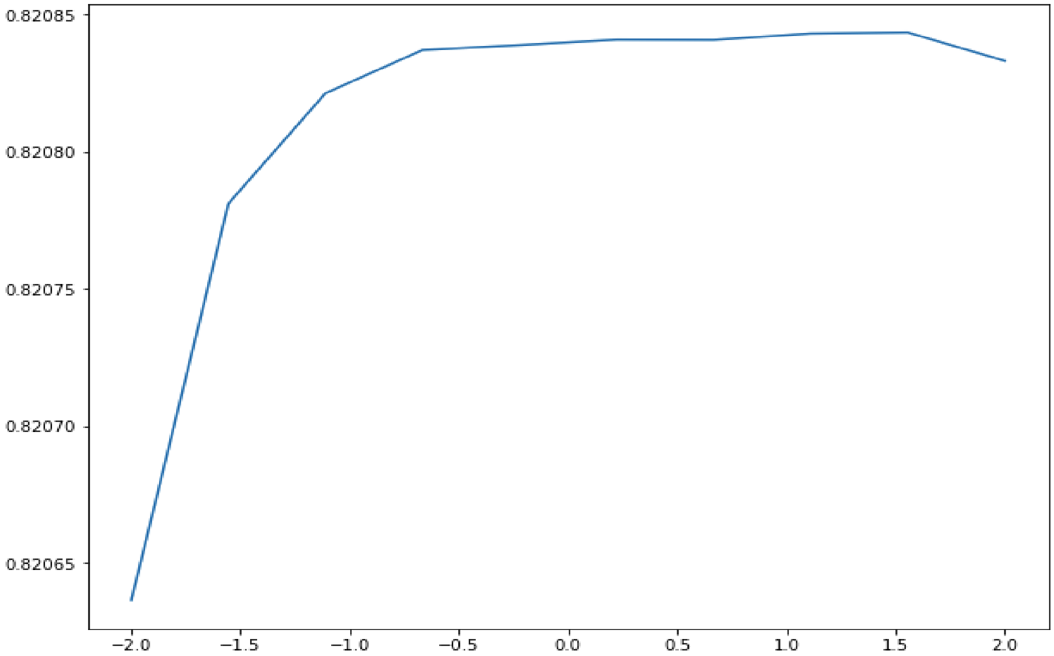


Shuffled Predictor	AUC
GeneralHealth	0.805225
BMI	0.805695
HighBP	0.812682
AgeBracket	0.812703
HighChol	0.814982
HeavyDrinker	0.819049
BiologicalSex	0.819431
Myocardial	0.819762
IncomeBracket	0.819794
HardToClimbStairs	0.820291
Stroke	0.820352
PhysicalHealth	0.820371
MentalHealth	0.820382
EducationBracket	0.82039
HasHealthcare	0.82041
Fruit	0.820448
PhysActivity	0.82045
NotAbleToAffordDoctor	0.820457
Smoker	0.820467
Vegetables	0.820468
Zodiac	0.820469

   The AUC of the full model (average of 5 fold cv) is 0.8205. The AUC is quite high, meaning that the logistic regression model is a good fit.

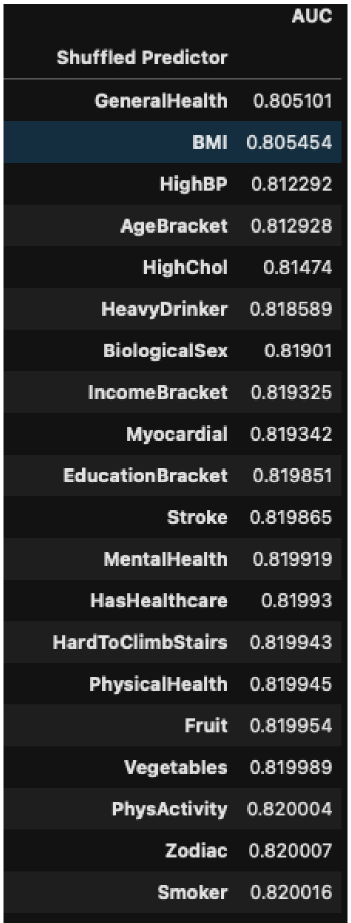
2. Build a SVM. Doing so: What is the best predictor of diabetes and what is the AUC of this model?

The default SVM model has an AUC of 0.8209. I did a hyperparameter tuning to see if the model could be better. The C values I tested were np.logspace(-2, 2, 10). I used this list of C values because through some playing-around with C, I found AUC of the model to be good within that range. The result is below:



The AUC basically stops increasing after C = log(0) = 1, so I used that value for C because higher C could induce overfitting and increases training time, and the AUC of the model is 0.8209, suggesting that this model is a good fit.

To find the best predictor, I shuffled each predictor's labels and recorded the AUCs. Again, from this method, GeneralHealth is the best predictor. GeneralHealth being the most important predictor indicates that we can predict diabetes from a person's overall health better than any other feature in this dataset.

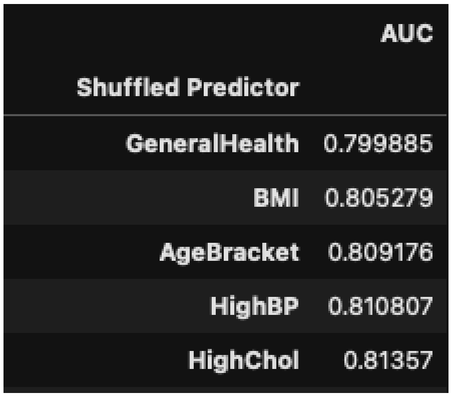


	AUC
Shuffled Predictor	
GeneralHealth	0.805101
BMI	0.805454
HighBP	0.812292
AgeBracket	0.812928
HighChol	0.81474
HeavyDrinker	0.818589
BiologicalSex	0.81901
IncomeBracket	0.819325
Myocardial	0.819342
EducationBracket	0.819851
Stroke	0.819865
MentalHealth	0.819919
HasHealthcare	0.81993
HardToClimbStairs	0.819943
PhysicalHealth	0.819945
Fruit	0.819954
Vegetables	0.819989
PhysActivity	0.820004
Zodiac	0.820007
Smoker	0.820016

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

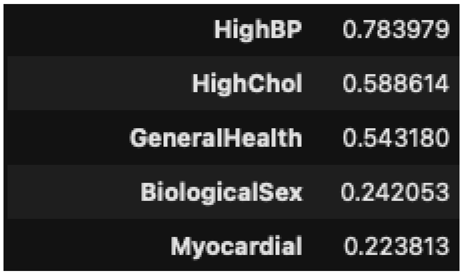
The default performance of the decision tree is terrible: AUC of 0.5963. Overfitting is very likely to be the problem. To make it better, I played around with the hyperparameters, and found that changing min_samples_leaf had the most effect. Through some trials, I found that min_samples_leaf = 0.002 had the best performance, yielding an AUC of 0.8181. I also tried GridSearchCV with a few other parameters because I wanted to see if the performance of the model could be increased further with by adjusting other hyperparameters, which GridSearchCV can do. The hyperparameter candidates were: {'min_samples_leaf': 1/np.logspace(0.302,3, 6, base = 10), 'min_samples_split': [2,3,4,5]}. One thing to note is that 1/np.logspace(0.302,3, 6, base = 10) ranges from 0.5 to close to 0, the range that decision tree takes for min_samples_leaf. I chose these candidates because when I played around with the hyperparameters individually, I found that these had a large effect on the model performance, and the values I chose were giving good results. The best parameter found was min_samples_leaf: 0.0034641764103524896, min_samples_split: 2. However, the AUC of the model with these parameters were 0.8172, not as good as just setting min_samples_leaf = 0.002. Maybe that parameter was not covered in the hyperparameter set. I could not add as many values as I like due to the exhaustive nature GridsearchCV. Therefore, for the final decision tree model, I used min_samples_leaf = 0.002. The AUC of the model is 0.8181, suggesting that this model is a good fit but not as good as previous models in this assignment..

Using permutation feature importance, I find the best predictor to be, again, GeneralHealth. GeneralHealth being the most important predictor indicates that we can predict diabetes from a person’s overall health better than any other feature in this dataset.



	AUC
Shuffled Predictor	
GeneralHealth	0.799885
BMI	0.805279
AgeBracket	0.809176
HighBP	0.810807
HighChol	0.81357

However, the feature_importance_ attribute of the decision tree says HighBP is the most important. The difference may be due to the difference in methods for calculating feature importance. Through my brief research, the feature_importance_ attribute is calculated based on impurity, according to my brief research. For the purpose of this assignment, I will stick to the permutation feature importance. So, GeneralHealth is the best predictor.

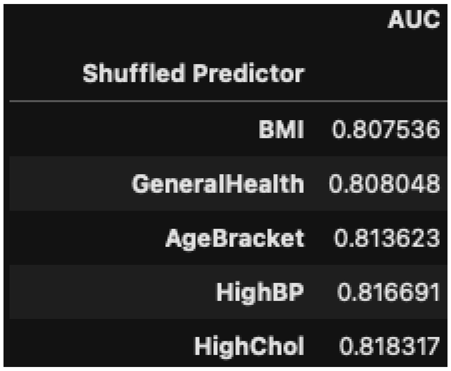


HighBP	0.783979
HighChol	0.588614
GeneralHealth	0.543180
BiologicalSex	0.242053
Myocardial	0.223813

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

   The random forest model with default parameters has an AUC of 0.8058, which is already good. Because I wanted to further improve the model, I did a gridsearch because this function allows me to find the best hyperparameters in a given set. The parameter candidates was {'n_estimators': [500, 1000, 1500, 2000],'max_samples': [0.1,0.2,0.3,0.4], 'max_features': [0.1,0.2,0.3,0.4], ‘bootstrap’: [True, False]}. I chose these candidates because when I played around with the hyperparameters individually, I found that these had a large effect on the model performance, and the values I chose were giving good results. From GridsearchCV, I found that the best parameters were n_estimators = 2000, max_features = 0.2, max_samples = 0.1, bootstrap = True. Using these hyperparameters, the result AUC is 0.8237, better than the default random forest, suggesting it is a better fit.

   To find the most predictive feature, I used permutation feature importance. The best predictor was BMI.BMI being the most important predictor indicates that we can predict diabetes from a person’s BMI better than any other feature in this dataset. Although it is different from the previous models, intuitively, this also makes sense because people suffering from diabetes are likely to be overweight, which is reflected in BMI. The top 5 best predictors are shown below:



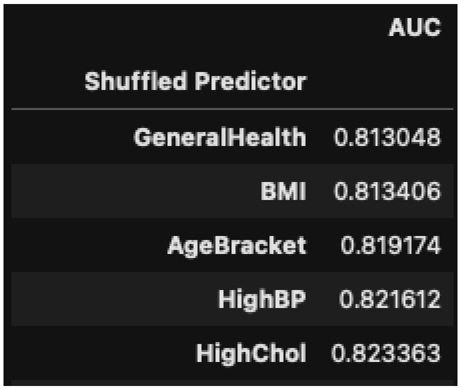
	AUC
Shuffled Predictor	
BMI	0.807536
GeneralHealth	0.808048
AgeBracket	0.813623
HighBP	0.816691
HighChol	0.818317

The following finding was very surprising to me: from the feature_importance_ attribute of random forest, Zodiac comes as the second most important feature. This is surprising because rationally speaking, the time at which a person is born should not have any effect on the health of that person. I am not sure how to interpret this finding.

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

The default adaBoost has an AUC of 0.825. Again, I used gridsearch to perform hyperparameter tuning because I want to find the best hyperparameters, which gridsearch can do. The candidates are: {'base_estimator__max_depth':[1,2,3,4], 'n_estimators':[100,500,1000], 'learning_rate':[0.001, 0.01,0.1, 1]}. I chose these candidates because when I played around with the hyperparameters individually, I found that these had a large effect on the model performance, and the values I chose were giving good results . Because GridsearchCV is exhaustive, I could not add many values. The best parameters I found were: DecisionTreeClassifier(max_depth = 3), n_estimators = 1000, learning_rate = 0.01. The AUC of adaBoost with these hyperparameters is 0.8283, the best performance out of all the models in this assignment.

Using permutation feature importance, I find that the best predictor is GeneralHealth, once again. GeneralHealth being the most important predictor indicates that we can predict diabetes from a person’s overall health better than any other feature in this dataset.

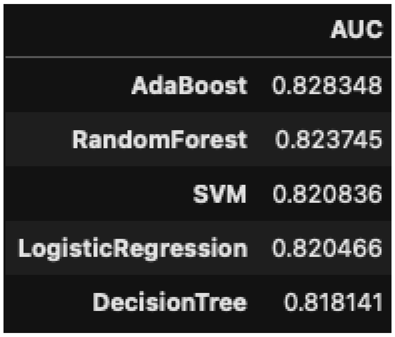


Shuffled Predictor	AUC
GeneralHealth	0.813048
BMI	0.813406
AgeBracket	0.819174
HighBP	0.821612
HighChol	0.823363

To add a comment, using permutation feature importance, the best predictors for all the models so far are consistent: the top 5 are the same.

6. Which of these 5 models is the best to predict diabetes in this dataset?

   AUC indicates the level of performance of a model, so the model with the highest AUC is the best model to predict diabetes. AdaBoost best predicts diabetes in this dataset. The AUC of each model is below:



	AUC
AdaBoost	0.828348
RandomForest	0.823745
SVM	0.820836
LogisticRegression	0.820466
DecisionTree	0.818141

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

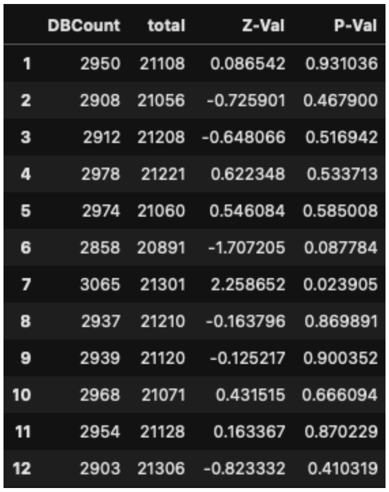
   Because in all the tree models, Zodiac has high importance score, I decided to do hypothesis tests to see if Zodiac really has an effect on diabetes. To do so, I did one-proportion Z-test on each Zodiac, with theoretical proportion = 1/12. I also assume that all samples are drawn randomly.

   Null hypotheses: the proportion of people with Zodiac x having diabetes is 1/12, x in {1,...,12}.

   Alternate hypothesis: the proportion of people with Zodiac x having diabetes is **not** 1/12, x in {1,...,12}.

   Alpha = 0.05 (standard convention)

   To calculate the p-values, I used statsmodel.stats.proportion.proportions_ztest. The result is below:



	DBCount	total	Z-Val	P-Val
1	2950	21108	0.086542	0.931036
2	2908	21056	-0.725901	0.467900
3	2912	21208	-0.648066	0.516942
4	2978	21221	0.622348	0.533713
5	2974	21060	0.546084	0.585008
6	2858	20891	-1.707205	0.087784
7	3065	21301	2.258652	0.023905
8	2937	21210	-0.163796	0.869891
9	2939	21120	-0.125217	0.900352
10	2968	21071	0.431515	0.666094
11	2954	21128	0.163367	0.870229
12	2903	21306	-0.823332	0.410319

Except for Zodiac 7, all other zodiacs have p-value < alpha. Now the questions is, what is with people of Zodiac 7 that leads them to having a higher probability of suffering from diabetes?