Part 1: Introduction

In part 1 we are dealing with the prediction of whether or not someone has heart disease. But it's not good enough to be able to predict the status of the peoples hearts. We need to do so with the maximum amount of power!! In order to do this we are going to utilize the powers of the k value and the various attributes to determine the correct combination of the two to maximize our systems predicting power! The stronger the power the better use this method would serve for the possibly heart disease ridden people.

Project 4 Presentation
Project 4

Part 1: Methods

Our goal is to find the amount and identities of the attributes that would maximize our predicting power. We used four different methods to determine the best attributes. Then compared the different methods resulting F1 values for different amounts of attributes to determine which one was the most effective.

The feature selection methods we tried were (chart identifier):

- Univariate Selection (univariate):
    - Selects the best features based on univariate statistical tests. For classification tasks, it uses the ANOVA F-value between each feature and the target. Fast and scalable but doesn't account for features interactions.
- Mutual Information (mutual_info):
    - Measures the mutual dependence between two variables. It can capture nonlinear relationships between features and the target, which linear correlation measures might miss.
- Recursive Feature Elimination (rfe):
    - Works by recursively removing attributes and building a model on those attributes that remain. It uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute.

- Random Forest Feature Importance (random_forest):
    - Uses the Random Forest algorithm to compute feature importance. Optimizes based on how much each feature decreases the weighted impurity in a tree.

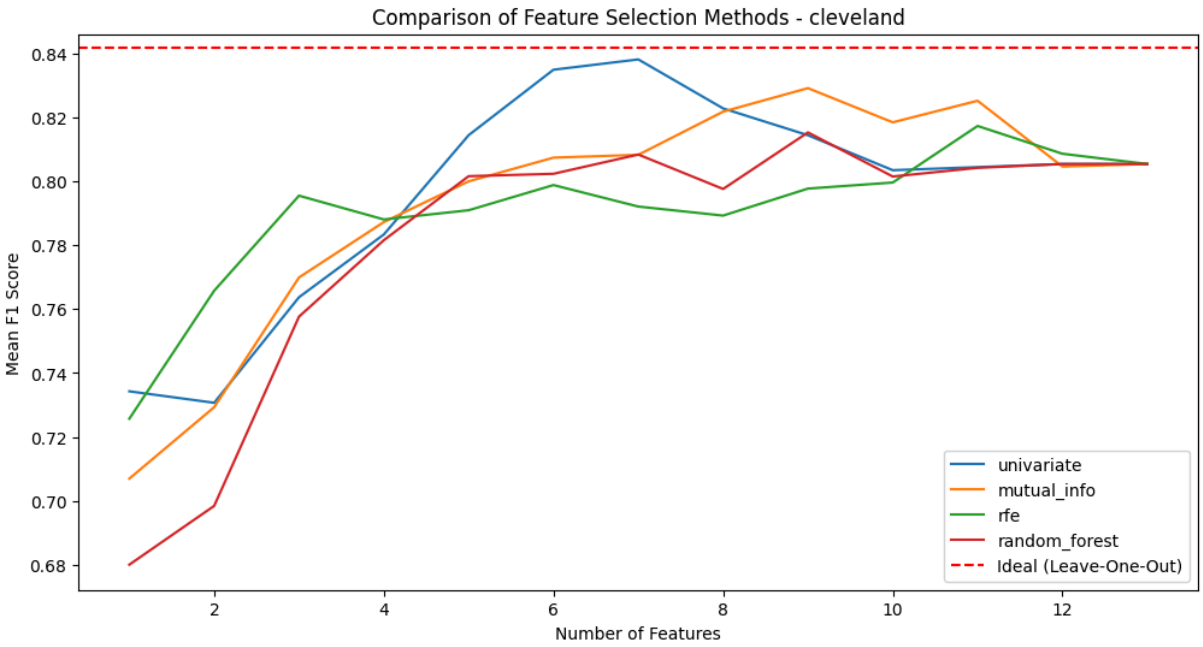
By exploring the different methods, we can better determine and be certain that we are using the best method for the case at hand. Increasing the robustness of our methods, and better utilizing the complementary information in the data sets.

We also will be using the Leave-One-Out method to determine the ideal goal to compare our results too. The closer to this ideal the better.

Part 1: Results

For this results section, we are using the results and numbers obtained from the use of the sample test data. That way it would be repeatable and easily used. So that's what these results show. We will also be using the Leave-One-Out method throughout. The value of which was found to be 0.842.

We started with finding out which attributes and how many should be included in our method. As seen in methods, we used a variety of different attribute picking formulas. And determining which method and how many features gave us the highest mean F1 score to result in the best possible combination. This model's best was considered to be the closest to the ideal found through the Leave-One-Out method. For the sample data it was determined that the best combination was to have 7 features (k) be included, and the method of determining the attributes will be Univariate Selection.

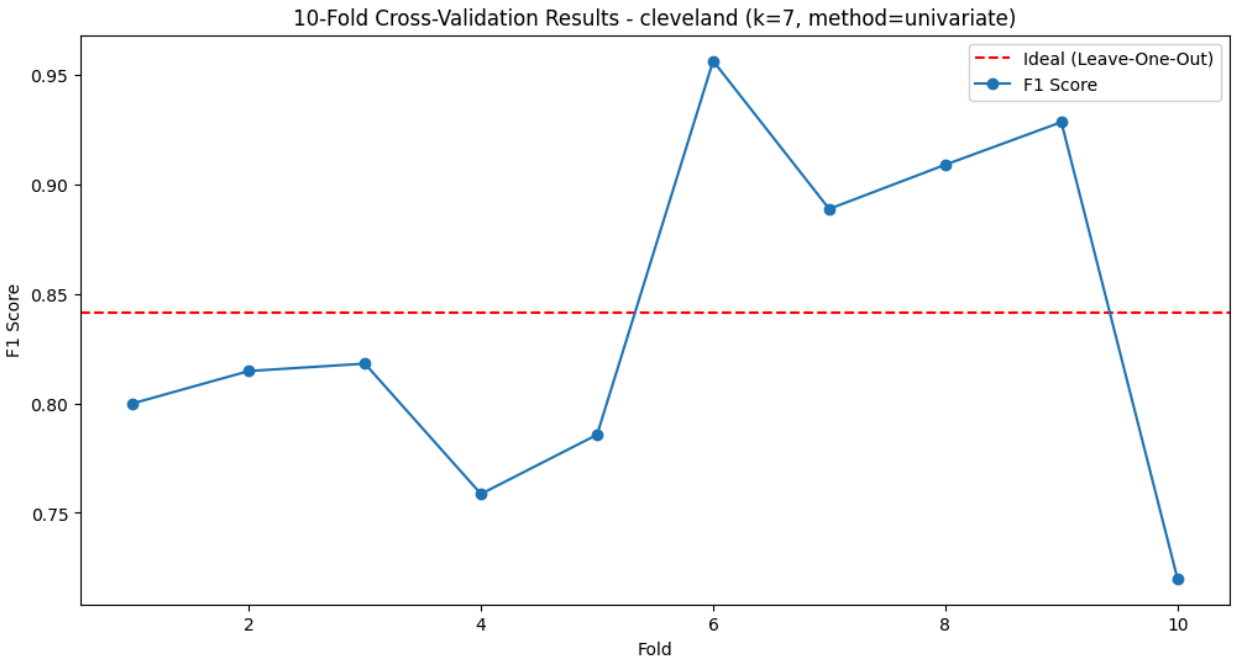


The features that were included are cp, thalach, exang, oldpeak, slope, ca, and thal. And the Mean F1 score was 0.838. The rest of the F1 scores and other details can be found in the project files if the need arises.

Now we can use our findings to better ensure our ability to correctly predict the chance of heart disease of the people in our area! Helping to put the community at ease and enable better maintenance of their health!

We also calculated the amount of folds our cross validation should have to obtain the best results. By reporting the results after each fold and comparing the value to the ideal found using the Leave-One-Out method we determined that the optimal amount of folds was 3.

Our final F1 score for the Test Sample Data was 0.882, with a recall of 0.833, and a precision of 0.938. These are pretty good numbers! Showing that our method did provide us with a pretty high F1 score, and therefore a stronger predicting power!



Part 2: Introduction

In part 2 we decided to look into kaggles diabetes data. If the first part can help better predict the occurrence of heart disease, putting the sufferers at a better chance of fighting off the resulting effects. Then this one could help those with diabetes! Which is a disease that is better caught early when there is a chance of reversing the process. And is extremely important to find before the sufferers lose limbs! We will be using mostly the same pattern as we did in part 1. By finding the best number and type of attributes for our data type we can ensure we have the best possible chance of predicting if someone will have diabetes!

Part 2: Dataset

For the most part not much was done to wrangle our found dataset. We got the data set from Kaggle. Specifically the Diabetes Health Indicators Dataset. Then we did a minimal clean up by finding and replacing all the NaN values with the mean to help standardize those values and prevent them from changing the results. We didn't use the full set, because that would be a lot of information that takes a long time to go through. Enough to take a whole day or so to do just the training portion. So instead we took a

randomly selected set of 1000 entries. That way we could get a decent chunk and hopefully a good picture of the data set.

Our main focus was the diabetes binary. This would be able to show if our predictions were correct or not. And with the rest of the data set covering soo many parts, we should have the best possible chance to correctly predict the diabetes status of the person.

Part 2: Methods

Our goal is to find the amount and identities of the attributes that would maximize our predicting power. We used four different methods to determine the best attributes. Then compared the different methods resulting F1 values for different amounts of attributes to determine which one was the most effective.

The feature selection methods we tried were (chart identifier):

- Univariate Selection (univariate):
    - Selects the best features based on univariate statistical tests. For classification tasks, it uses the ANOVA F-value between each feature and the target. Fast and scalable but doesn't account for features interactions.
- Mutual Information (mutual_info):
    - Measures the mutual dependence between two variables. It can capture nonlinear relationships between features and the target, which linear correlation measures might miss.
- Recursive Feature Elimination (rfe):
    - Works by recursively removing attributes and building a model on those attributes that remain. It uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute.
- Random Forest Feature Importance (random_forest):
    - Uses the Random Forest algorithm to compute feature importance. Optimizes based on how much each feature decreases the weighted impurity in a tree.

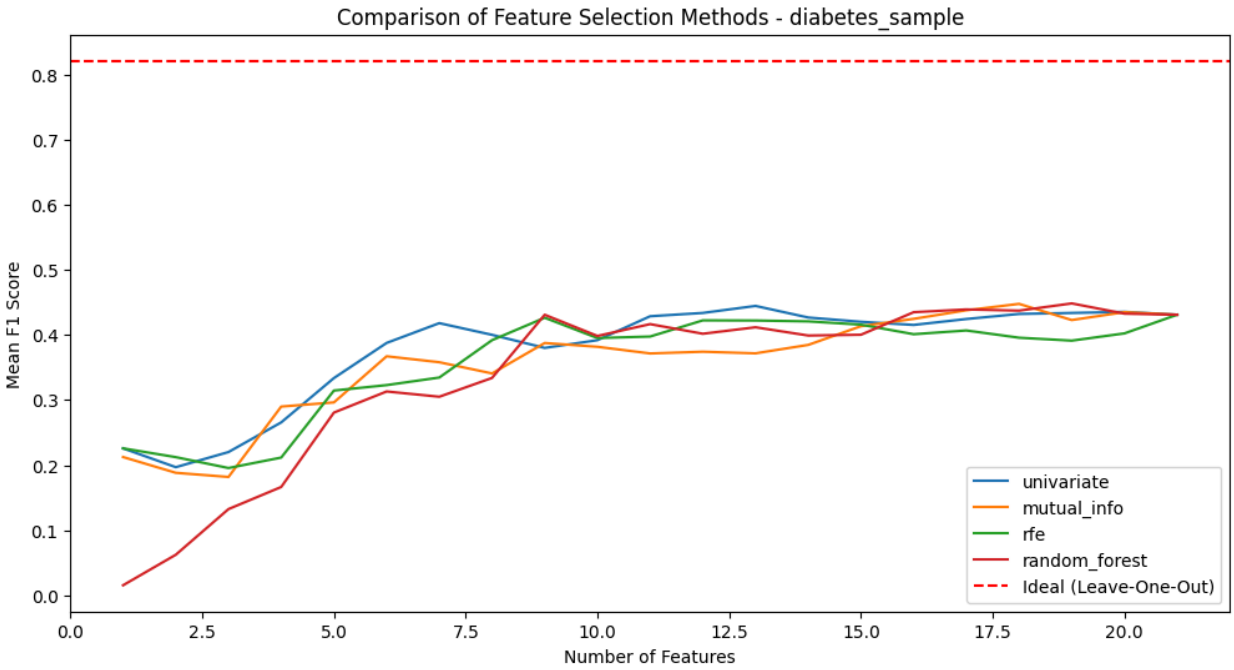
By exploring the different methods, we can better determine and be certain that we are using the best method for the case at hand. Increasing the robustness of our methods, and better utilizing the complementary information in the data sets.

We also will be using the Leave-One-Out method to determine the ideal goal to compare our results too. The closer to this ideal the better.

Part 2: Results

For this part we will also be using the Leave-One-Out method throughout, just like part 1. The value of which was found to be 0.821. And the data, as discussed in the database portion, was found on kaggle. Specifically the diabetes health indicators dataset.

We started with finding out which attributes and how many should be included in our method. As seen in methods, we used a variety of different attribute picking formulas. And determining which method and how many features gave us the highest mean F1 score to result in the best possible combination. This model's best was considered to be the closest to the ideal found through the Leave-One-Out method. For the retrieved dataset it was determined that the best combination was to have 19 features (k) be included, and the method of determining the attributes will be Random Forest.



The Mean F1 score was 0.448. The rest of the F1 scores and other details, including which attributes were used, can be found in the project files if the need arises.

Now we can use our findings to better ensure our ability to correctly predict the chance of diabetes for the people in our area! Helping to put the community at ease and enable better maintenance of their health!

We also calculated the amount of folds our cross validation should have to obtain the best results. By reporting the results after each fold and comparing the value to the ideal found using the Leave-One-Out method we determined that the optimal amount of folds was 9.

Our final F1 score for the Test Sample Data was 0.375, with a recall of 0.444, and a precision of 0.324. This is not very good. Our theories for this are that there are some parts of the data that are likely non-linear. Which would not be addressed well with the models we are using, and be better served with a logistic regressor model instead. And the fact that our model is a bit simple, and not complex enough to fit the data as well. Especially with our limited training set.