Building a Classifier to Predict Cardiovascular Disease

Author: ***Suleyman Qayum***

Business Understanding

Cardiovascular disease is the term for all types of diseases that affect the heart and/or blood vessels, including coronary heart disease (clogged arteries), which can cause heart attacks, stroke, congenital heart defects and peripheral artery disease. According to the *National Heart, Lung, and Blood Insitute (NHLBI)*, more than 800,000 people die of cardiovascular disease every year in the United States alone.

> *As part of an initiative to reduce the number of deaths related to cardiovascular disease in the United States, the U.S. Department of Health and Human Services (HHS) have allocated funding for the development of a machine learning model that can accurately predict whether or not a patient is at risk for cardiovascular disease. The eventual goal is to deploy such a model and integrate it into the major EHR systems as a preliminary screening tool for cardiovascular disease. They also request that the most important features be identified.*

Data Understanding

Overview

> *The dataset describes basic medical information for 69,301 patients. In addition to being factors that contribute to cardiovascular disease,* ***the features of this dataset were chosen for their simplicity and the fact that they can be easily and quickly obtained in a medical setting****. As a result, they fall under 3 basic categories with respect to their origin:*
>
> - *Objective Patient Information:* basic information about the patient that can be verified as truth (*i.e. age, height, weight, and gender*)
> - *Subjective Patient Information:* basic information about the patient that cannot be verified, but is accepted as truth (*i.e. whether or not the patient smokes, drinks alcohol, or is physically active*)
> - *Measurement:* simple blood measurements (*e.g. blood pressure, cholesterol and glucose levels*)

Features

> **The dataset consists of 11 input features and 1 target variable:**

> - **age** *[int | continuous] - patient's age (years)*
> - **height** *[int | continuous] - patient's height (cm)*
> - **weight** *[float | continuous] - patient's weight (kg)*
> - **gender** *[int | categorical] - patient's gender*
>   - *1 = male*
>   - *2 = female*
> - **bp_hi** *[int | continuous] - patient's systolic blood pressure*
> - **bp_lo** *[int | continuous] - patient's diastolic blood pressure*
> - **cholesterol** *[int | categorical] - patient's cholesterol level*
>   - *1 = normal*
>   - *2 = high*
>   - *3 = very high*
> - **glucose** *[int | categorical] - patient's glucose level*
>   - *1 = normal*
>   - *2 = high*
>   - *3 = very high*
> - **smoking** *[bool] - indicates whether or not the patient is a smoker*
>   - *0 = patient is a non-smoker*
>   - *1 = patient is a smoker*
> - **alcohol** *[bool] - indicates whether or not the patient drinks alcohol*
>   - *0 = patient dos not drink alcohol*
>   - *1 = patient drinks alcohol*

> - **active** *[bool] - indicates whether or not the patient is physically active*
>   - *0 = patient is not physically active*
>   - *1 = patient is physically active*

> - **[Target Variable] cardio** *[bool] - indicates the presence or absence of cardiovascular disese in the pateint*
>   - *0 = patient does not suffer from cardiovascular disease*
>   - *1 = patient suffers from cardiovascular disease*

Issues

> *Given the fact that that the patients in the dataset range from 35 to 60 years old, the outliers listed below give rise to alarming discrepencies*:
>
> - *heights under* $120\ cm\ (\approx 4\ ft)$ *and above* $245\ cm\ (\approx\ 8\ ft)$
> - *weights under* $30\ kg\ (\approx 66\ lbs)$
> - *diastolic (* `bp_lo` *) and systolic (* `bp_hi` *) blood pressures:*
>   - *less than* $0\ mmHg$ *(physically impossible)*
>   - *under* $30\ mmHg$ *(extremely unlikely)*
>   - *above* $500\ mmHg$ *(a person would explode - cusing death)*

> *The* `bp_lo` *,* `bp_hi` *,* `height` *, and* `weight` *features are not normally distributed throughout the dataset:*
>
> - `bp_lo` *,* `bp_hi` *are extremely right-skewed*
> - `height` *is left-skewed with a jagged curve at the median*
> - `weight` *is right-skewed with a jagged curve at the median*

Data Preparation

Data Cleaning

Entries containing negative values for `bp_hi` or `bp_lo` were removed immediately due tosuch values being physically impossible. The continuous numerical features (`bp_lo` , `bp_hi` , `height` , and `weight`) contained outliers, many of which did not make sense from a physiological point of view. These outliers were removed using the inter-quartile range method, such that the distributions of the these numerical predictors were no longer skewed or contained absurd values.

Feature Engineering

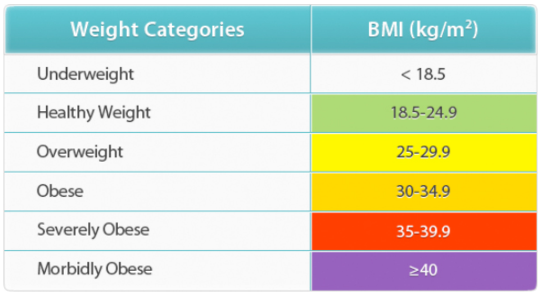
`bmi` & `bmi_category`

Obesity is known to have a causal link with cardiovascular disease. *This condition is categorized by calculating the* ***Body Mass Index (BMI)*** *for each patient in the dataset, and then using this BMI to determine the patient's weight category.*

- *BMI was calculated using the following formula:*

$$BMI = \frac{weight(kg)}{[height(m)]^2}$$

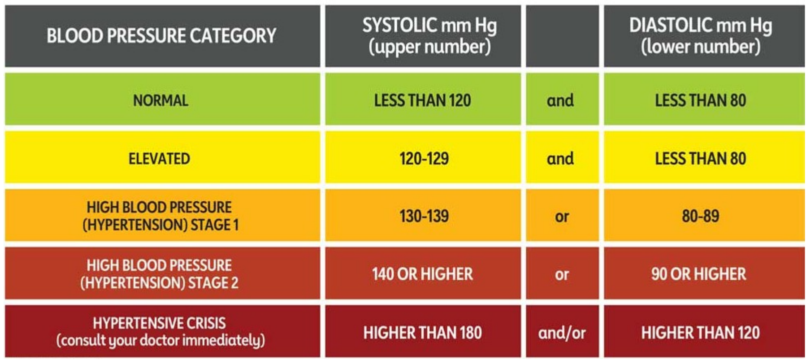
- *Using BMI, a patient's weight class was determined from the following table:*



Weight Categories	BMI (kg/m²)
Underweight	< 18.5
Healthy Weight	18.5-24.9
Overweight	25-29.9
Obese	30-34.9
Severely Obese	35-39.9
Morbidly Obese	≥40

bp_category

Hypertension is also known to have a causal link with cardiovascular disease. *This condition is categorized a patient's blood pressure level, which will be determined from the following table:*



BLOOD PRESSURE CATEGORY	SYSTOLIC mm Hg (upper number)		DIASTOLIC mm Hg (lower number)
NORMAL	LESS THAN 120	and	LESS THAN 80
ELEVATED	120-129	and	LESS THAN 80
HIGH BLOOD PRESSURE (HYPERTENSION) STAGE 1	130-139	or	80-89
HIGH BLOOD PRESSURE (HYPERTENSION) STAGE 2	140 OR HIGHER	or	90 OR HIGHER
HYPERTENSIVE CRISIS (consult your doctor immediately)	HIGHER THAN 180	and/or	HIGHER THAN 120

Encoding Categorical Features

The `glucose` , `cholesterol` , `gender` , `bmi_category` , and `bp_category` features were one-hot encoded via the `OneHotEncoder` transformer.

Scaling the Feature Matrix

The feature matrix was standardized using a `StandardScaler` object. The scaler was fitted to the training data only, and then used to scale the training, validation, and test sets. This was done to ensure no data leakage ocurred.

Modeling

Scoring

A custom F_2 scorer was created and used to measure training loss during validation and evaluate models when searching a parameter grid. This score is meant to emphasize recall without compromising precision. The formula for the $F_2\text{-Score}$ is based on the more general $F_\beta\text{-Score}$:

$$F_\beta = \frac{(1+\beta^2)PR}{\beta^2 P + R}$$

$\beta = 2$
$P = \text{precision}$
$R = \text{recall}$

Classifiers

Random Forest

A `RandomForestClassifier` base estimator was fit to the training dataset. The training loss was minimized by first adjusting the maximum depth of its trees, resulting in the `rfc_bestMaxDepth_1` classifier. Starting from the base estimator again, the minimum number of samples required for a node to become a leaf was adjusted until training loss was again minimized, resulting in the `rfc_bestMinSamplesLeaf_1` classifier. After some feature adjustment (discussed below), another pair of randomized forests, `rfc_bestMaxDepth_2` and `rfc_bestMinSamplesLeaf_2` , were created as updated versions of the previous two.

The random forest classifiers, along with their optimized hyperparameters, are listed below:

```
rfc_bestMaxDepth_1 = RandomForestClassifier(max_depth=9, n_jobs=-1, random_state=0)
rfc_bestMinSamplesLeaf_1 = RandomForestClassifier(min_samples_leaf=35, n_jobs=-1, random_state=0)
rfc_bestMaxDepth_2 = RandomForestClassifier(max_depth=8, n_jobs=-1, random_state=0)
rfc_bestMinSamplesLeaf_2 = RandomForestClassifier(min_samples_leaf=39, n_jobs=-1, random_state=0)
```

The validation metrics for the final two random forest classifiers (`rfc_bestMaxDepth_2` *and* `rfc_bestMinSamplesLeaf_2` *) are shown below:*

Logistic Regression

A `LogisticRegression` estimator was fit to the training set and the regularization (`C`) parameter that minimized training loss was found with a grid search, resulting in the `logreg_best` estimator.

The parameter information for `Logreg_best` is shown below:

> ```
> logreg_best = LogisticRegression(C=0.0138643651488824, max_iter=10000, random_state=0, solver='saga')
> ```

K-Nearest Neighbors (KNN)

A `KNeighborsClassifier` estimator was fit to the training set. The number of neighbors (`n_neighbors` parameter) and distance metric (`p`) that minimized training loss was found through a grid search, resulting in the `knn_best` estimator.

The parameter information for `knn_best` is shown below:

> ```
> knn_best = KNeighborsClassifier(n_jobs=-1, n_neighbors=21, p=3)
> ```

Bagging Classifier [KNN Base Estimator]

A `BaggingClassifier` , with `knn_best` as the base estimator, was fit to the training data and optimized in a grid search in order to reduce the variation of `knn_best` . This classifier was named `bagged_knn_best` .

The parameter information for `bagged_knn_best` is shown below:

> ```
> bagged_knn_best = BaggingClassifier(base_estimator=KNeighborsClassifier(n_jobs=-1, n_neighbors=21, p=3), bootstrap_features=True, max_features=0.
> ```

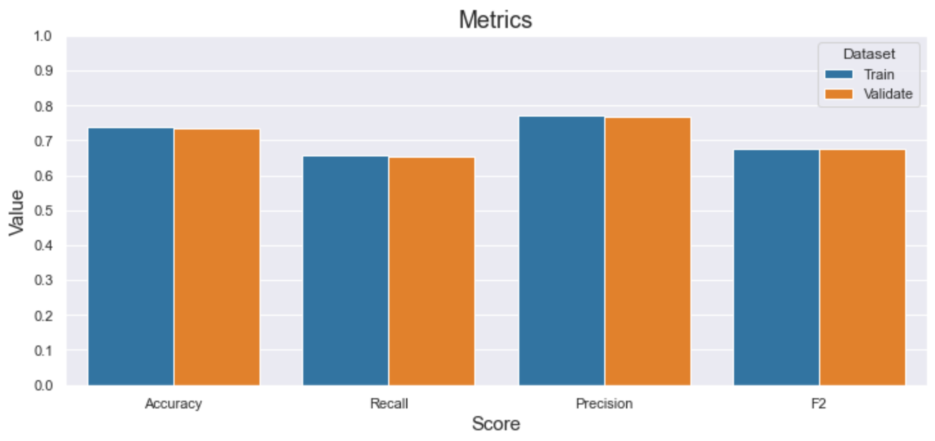
Results

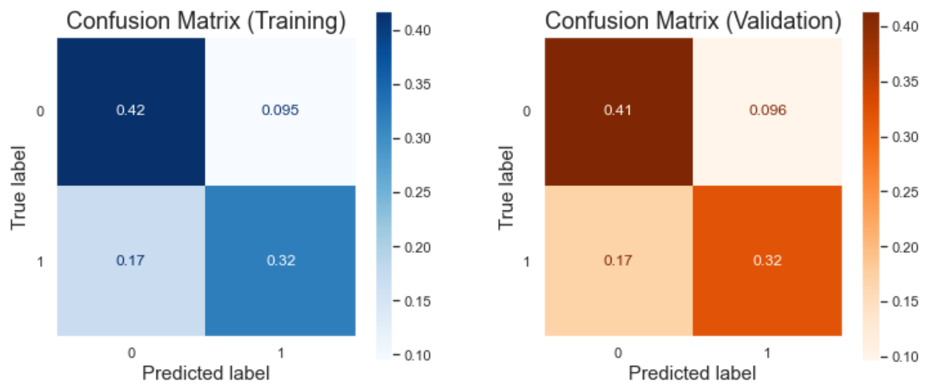
Models

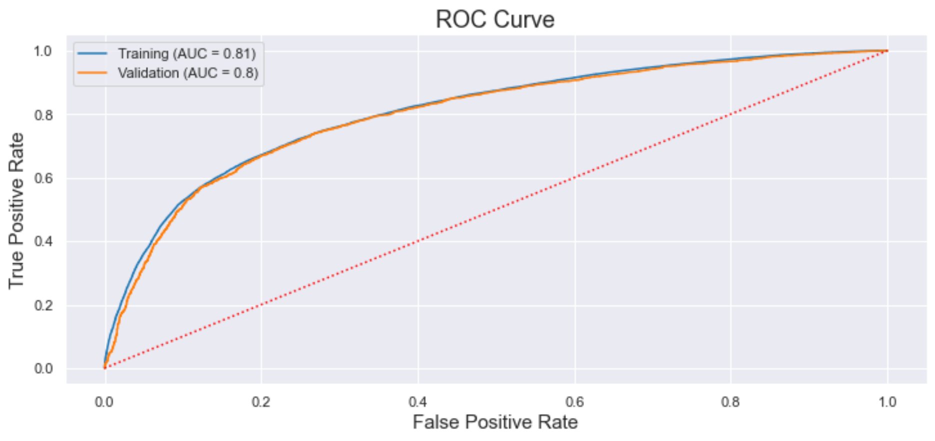
The validation metrics for predictions against the test (hold-out) set for each model is shown below:

`rfc_bestMaxDepth`

> ```
> RandomForestClassifier(max_depth=8, n_jobs=-1, random_state=0)
> ```

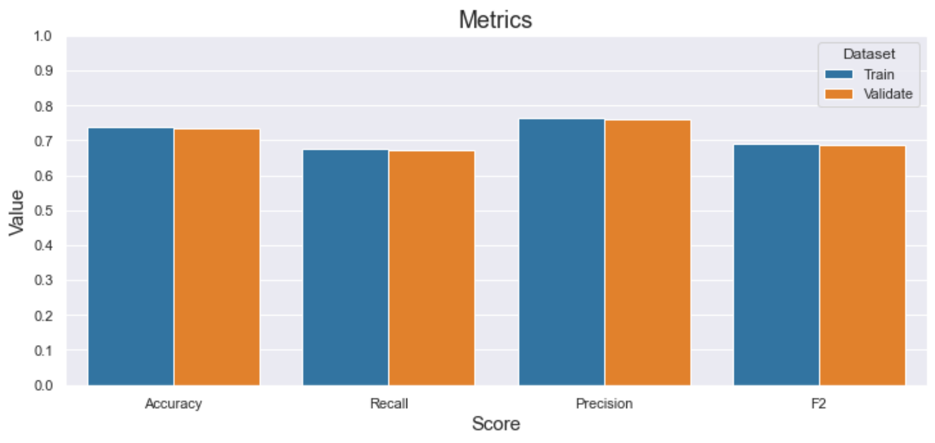


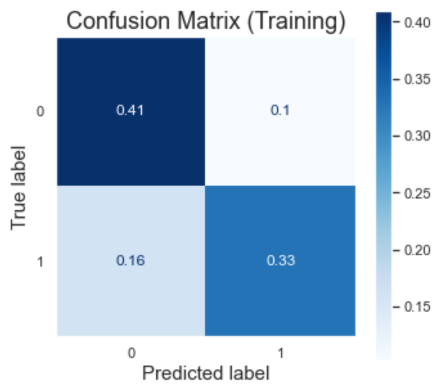


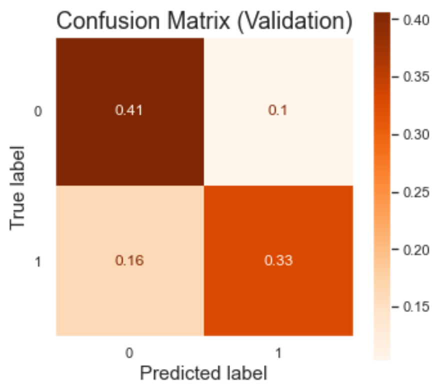


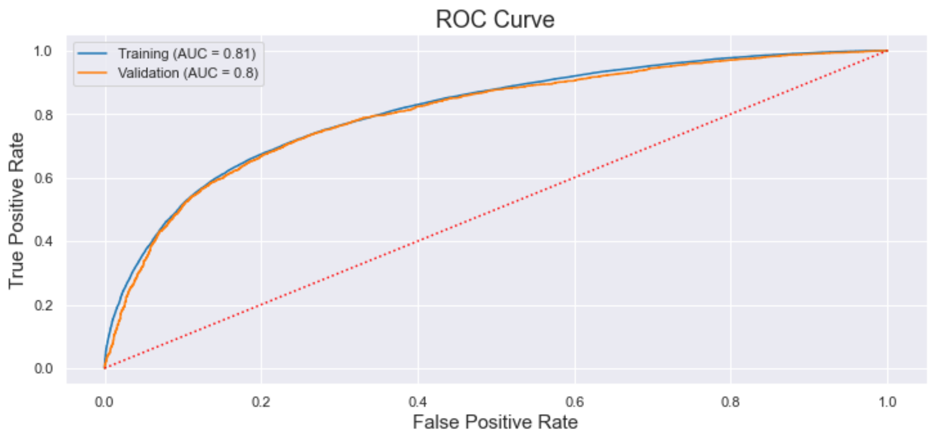
rfc_bestMinSamplesLeaf

```
RandomForestClassifier(min_samples_leaf=39, n_jobs=-1, random_state=0)
```



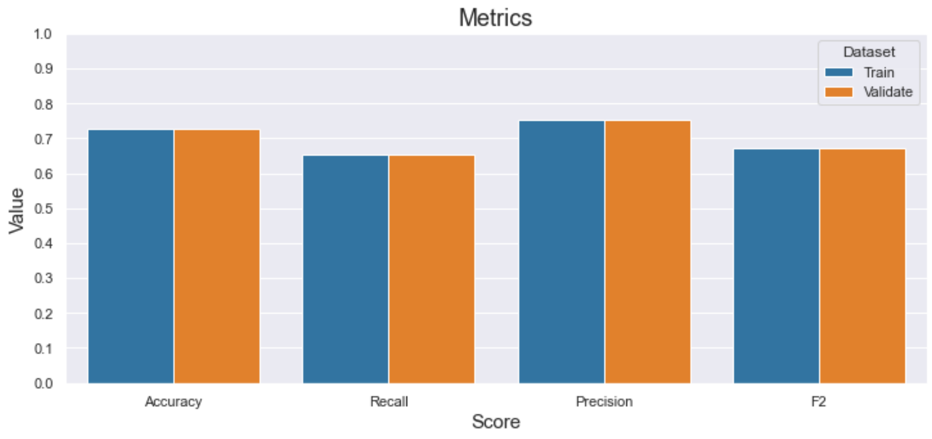


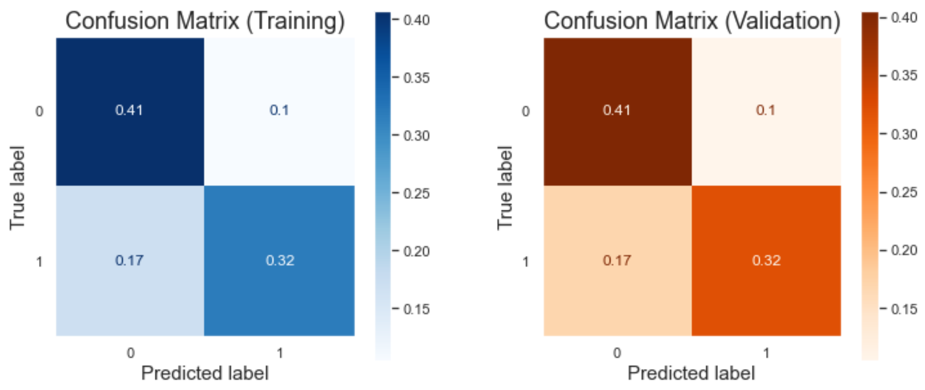


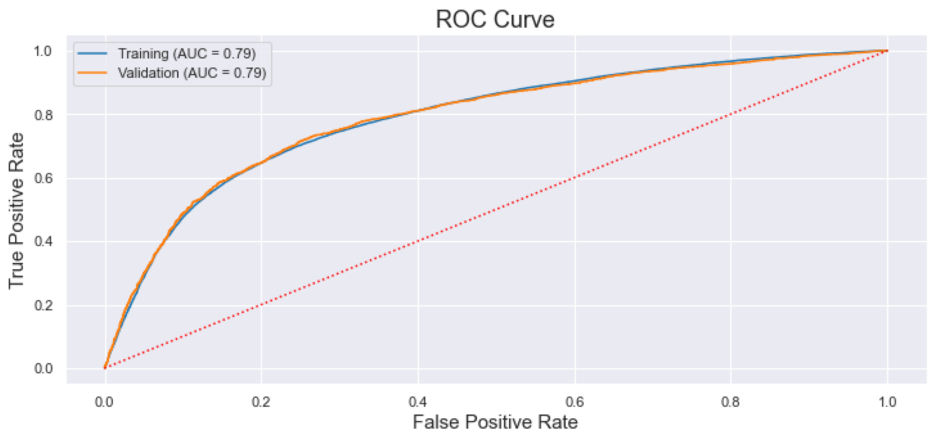


logreg_best

```
LogisticRegression(C=0.0138643651488824, max_iter=10000, random_state=0, solver='saga')
```

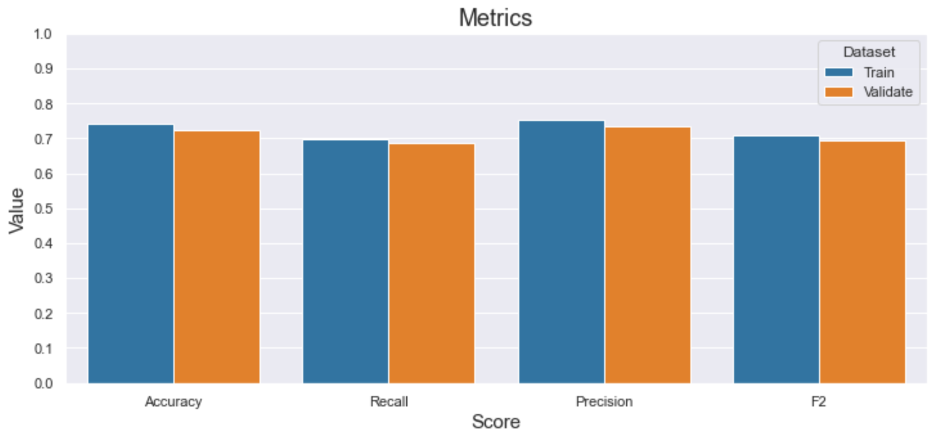


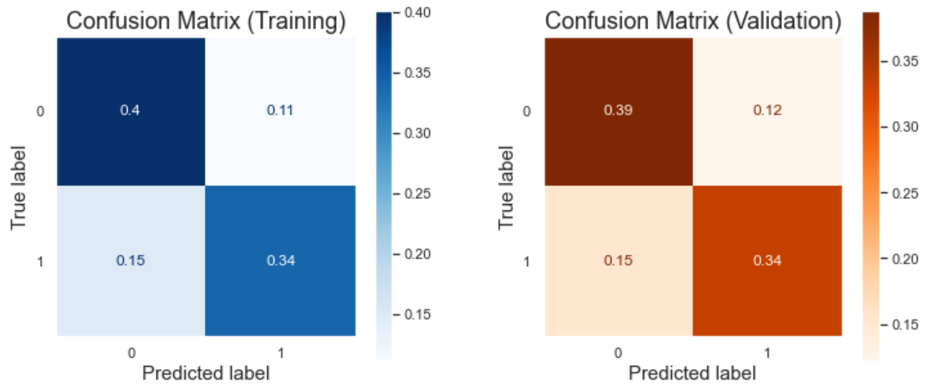


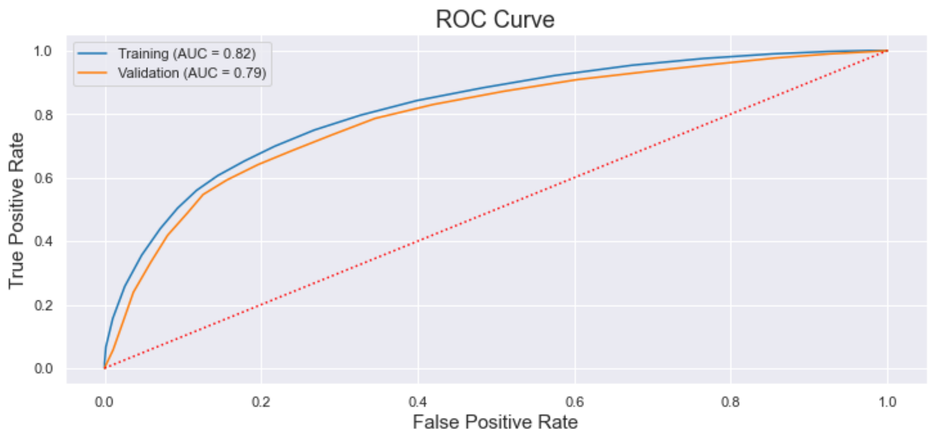


knn_best

```
KNeighborsClassifier(n_jobs=-1, n_neighbors=21, p=3)
```

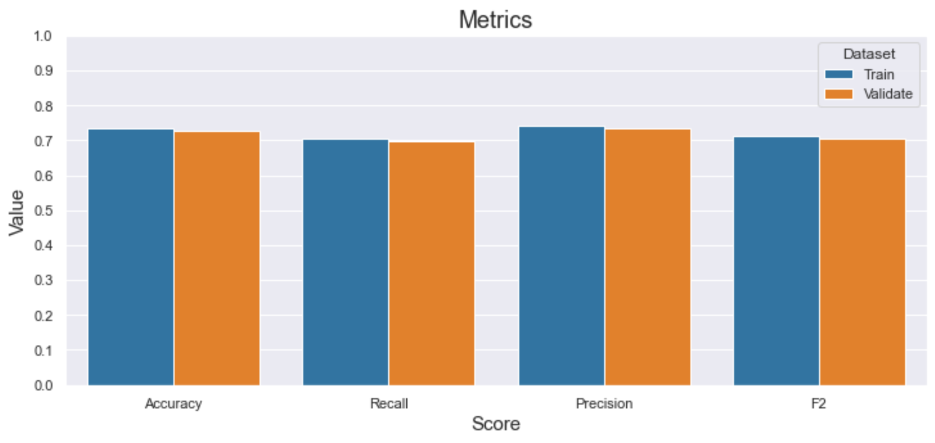


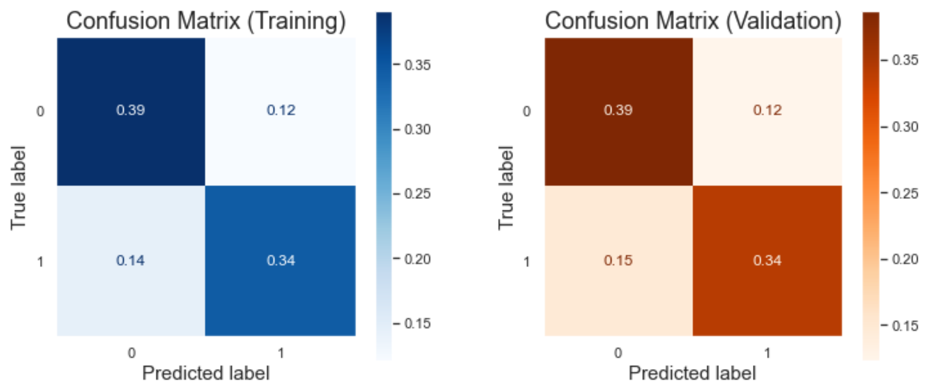


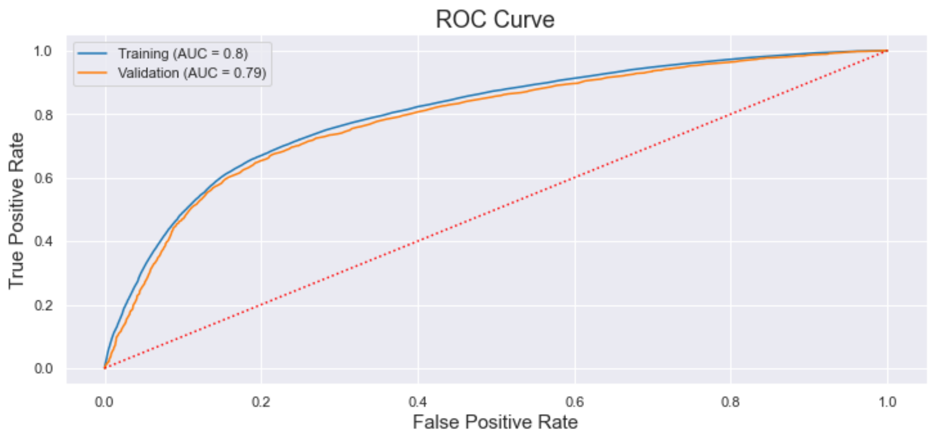


bagged_knn_best

```
BaggingClassifier(base_estimator=KNeighborsClassifier(n_jobs=-1, n_neighbors=21, p=3), bootstrap_features=True, max_features=0.5, max_samples=0.7
```



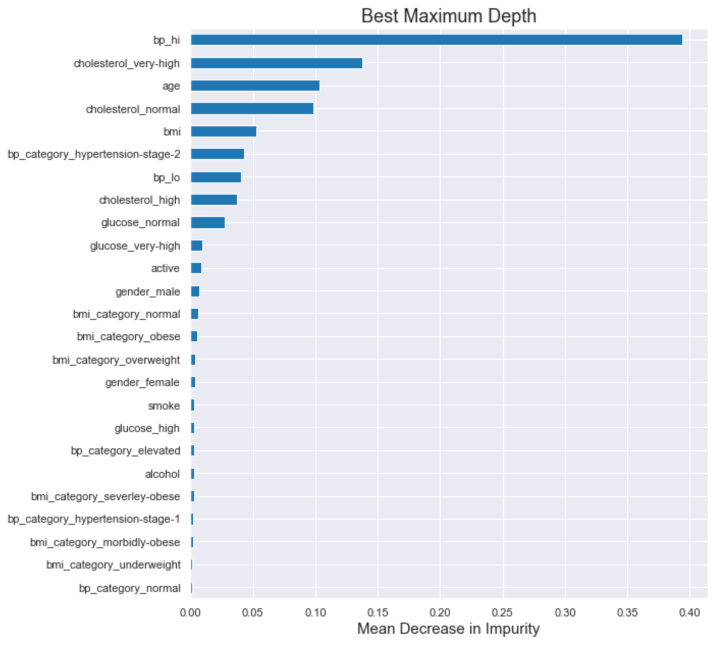


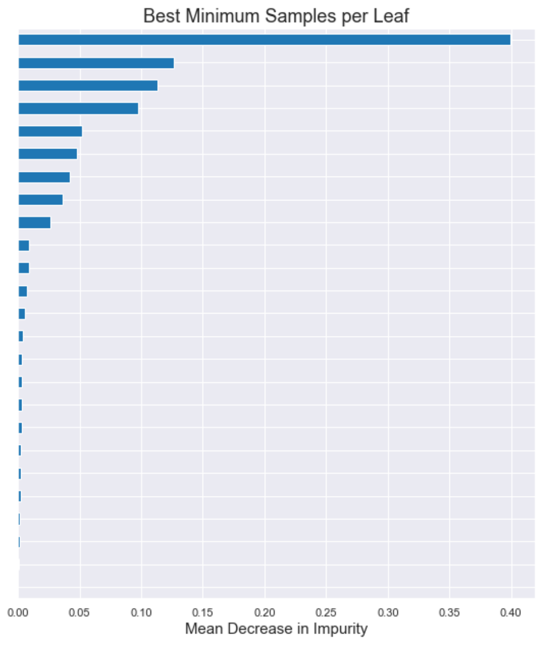


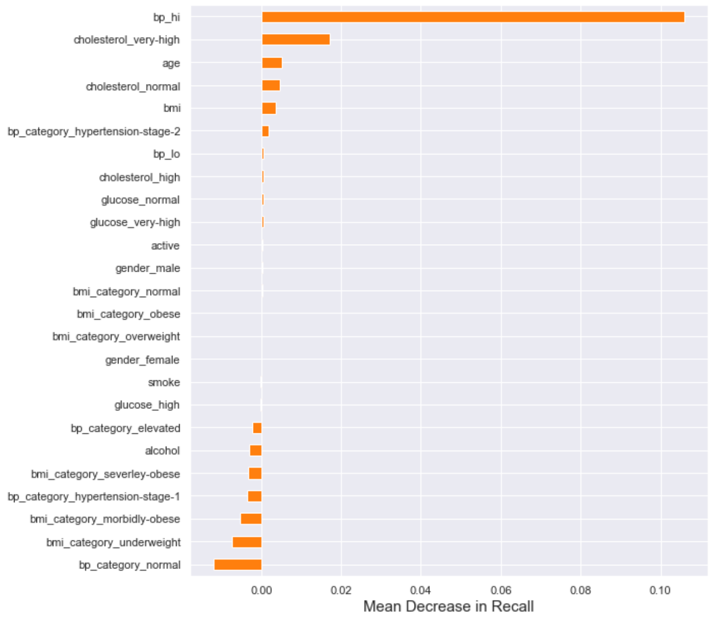
Feature Importances

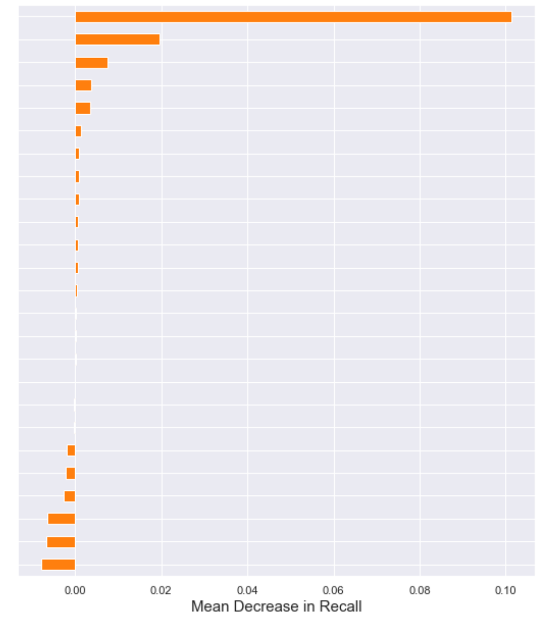
Impurity-based and permutation-based feature importances (with respect to recall and precision) were extracted from `rfc_bestMaxDepth_1` and `rfc_bestMinSamplesLeaf_1` .

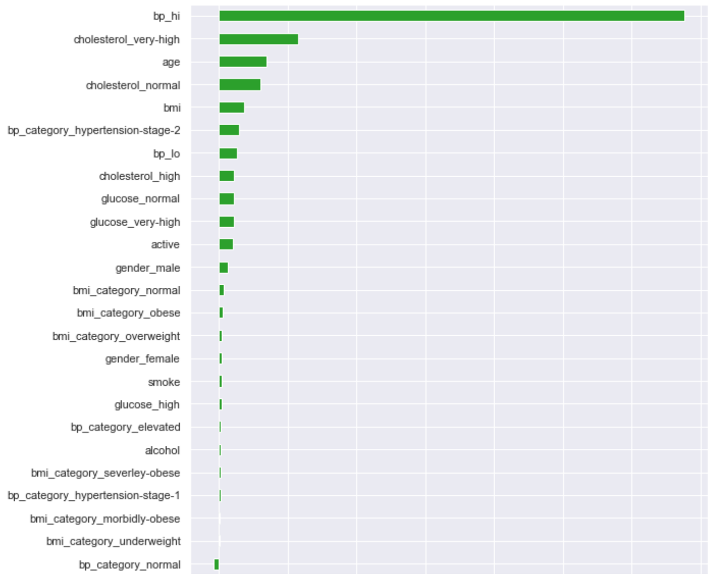
These feature importances are shown below:

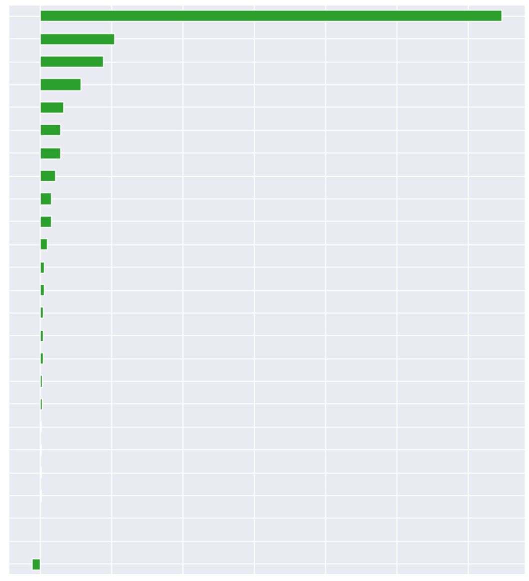

Best Maximum Depth
bp_hi
cholesterol_very-high
age
cholesterol_normal
bmi
bp_category_hypertension-stage-2
bp_lo
cholesterol_high
glucose_normal
glucose_very-high
active
gender_male
bmi_category_normal
bmi_category_obese
bmi_category_overweight
gender_female
smoke
glucose_high
bp_category_elevated
alcohol
bmi_category_severley-obese
bp_category_hypertension-stage-1
bmi_category_morbidly-obese
bmi_category_underweight
bp_category_normal
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Mean Decrease in Impurity

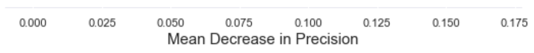

Best Minimum Samples per Leaf
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Mean Decrease in Impurity


bp_hi
cholesterol_very-high
age
cholesterol_normal
bmi
bp_category_hypertension-stage-2
bp_lo
cholesterol_high
glucose_normal
glucose_very-high
active
gender_male
bmi_category_normal
bmi_category_obese
bmi_category_overweight
gender_female
smoke
glucose_high
bp_category_elevated
alcohol
bmi_category_severley-obese
bp_category_hypertension-stage-1
bmi_category_morbidly-obese
bmi_category_underweight
bp_category_normal
0.00
0.02
0.04
0.06
0.08
0.10
Mean Decrease in Recall


0.00
0.02
0.04
0.06
0.08
0.10
Mean Decrease in Recall


bp_hi
cholesterol_very-high
age
cholesterol_normal
bmi
bp_category_hypertension-stage-2
bp_lo
cholesterol_high
glucose_normal
glucose_very-high
active
gender_male
bmi_category_normal
bmi_category_obese
bmi_category_overweight
gender_female
smoke
glucose_high
bp_category_elevated
alcohol
bmi_category_severley-obese
bp_category_hypertension-stage-1
bmi_category_morbidly-obese
bmi_category_underweight
bp_category_normal





It can be concluded from the plots displaying impurity-based importances that the tree-based models struggled to split the one-hot encoded features that were engineered in *Part C* (i.e. the `bmi_category` and `bp_category` columns), possibly because they have too many levels. This, in turn, means that only a small fraction of the data belongs to some of these levels, and the corresponding one-hot encoded columns will mostly contain zeros. It follows that splitting on this column produces a relatively small reduction in impurity, causing the tree-based algorithms to ignore these columns in favor of others.

With respect to the plots displaying permutation-based importances, it can also be seen that the `severley-obese` , `morbidly-obese` , and `underweight` levels of `bmi_category` , along with the `normal` , `elevated` , and `hypertension-stage-1` levels of `bp_category` , were actually detrimental to the recall of both models, since the recall scores for both `rfc_bestMaxDepth` and `rfc_bestMinSamplesLeaf` increased upon permutation of these categories.

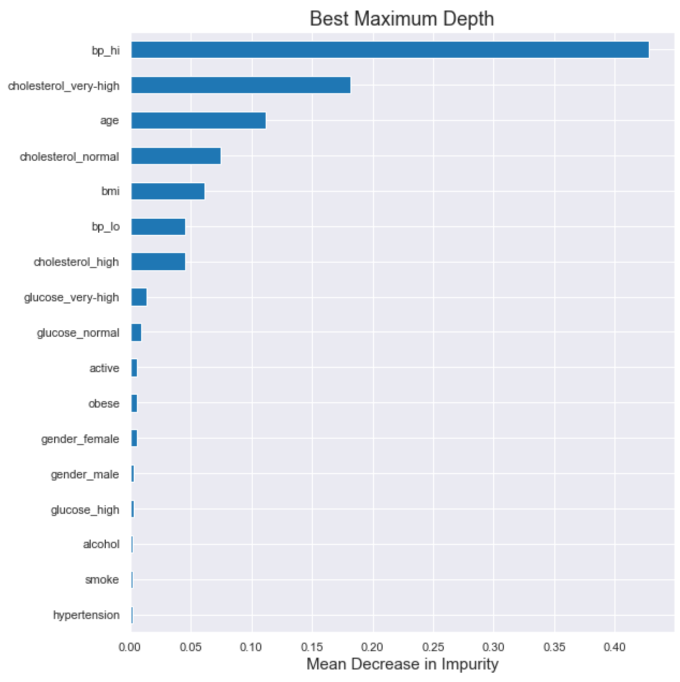
To alleviate these problems, the levels belonging to `bmi_category` and `bp_category` were grouped together.

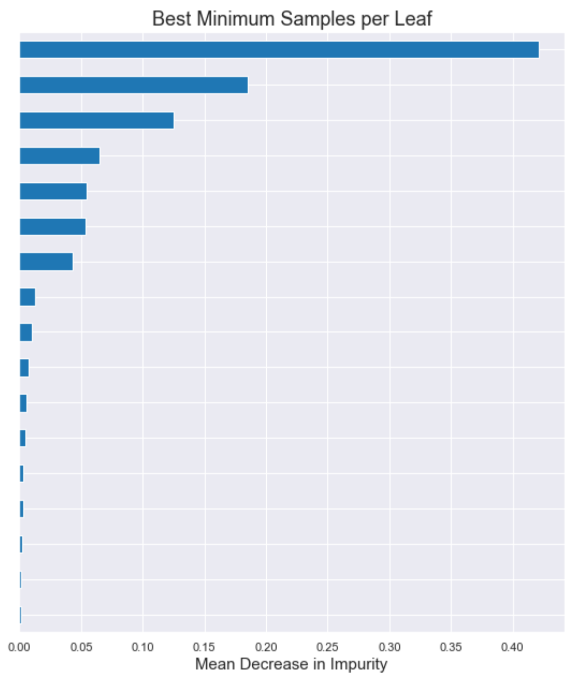
> *The BMI levels were grouped in such a way that the `overweight` , `severley-obese` , `morbidly-obese` , and `underweight` levels were eliminated. As a result, the BMI levels were grouped as:*
>
> - **`not-obese` = { `underweight` , `normal` , `overweight` }**
> - **`obese` = { `obese` , `severley-obese` , `morbidly-obese` }**
>   *The grouped columns were then merged into a single binary column called **`obese`** .*

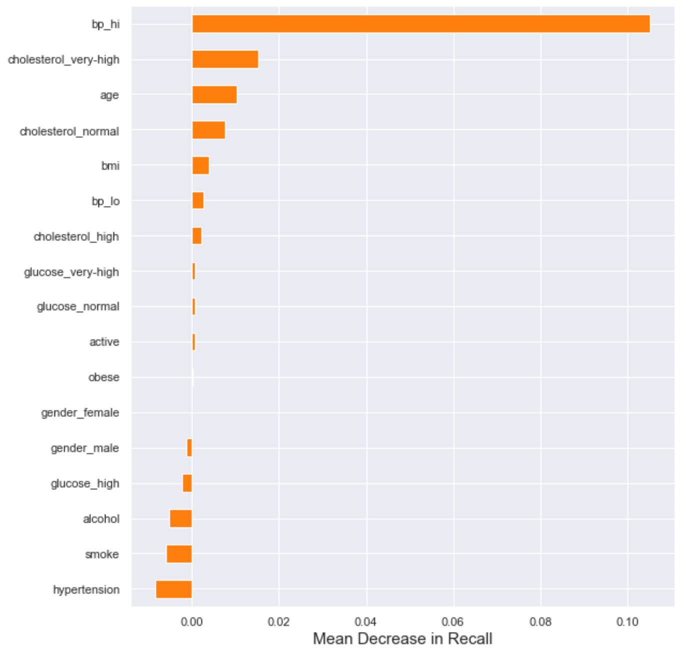
> *Similarly, the BP levels were grouped in such a way that the `normal` , `elevated` , and `hypertension-stage-1` levels were eliminated. As a result, the BP levels were grouped as:*
>
> - **`not-hypertension-stage-2` = { `normal` , `elevated` }**
> - **`hypertensive` = { `hypertension-stage-1` , `hypertension-stage-2` }**
>   *The grouped columns above were then merged into a single binary column called **`hypertension`** .*

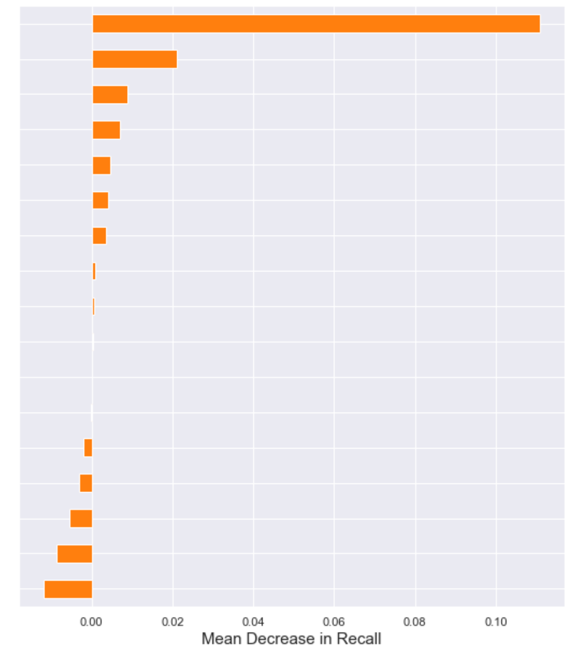
The `rfc_bestMaxDepth_2` and `rfc_bestMinSamplesLeaf_2` were trained on the adjusted dataset and used to extract updated feature importances.

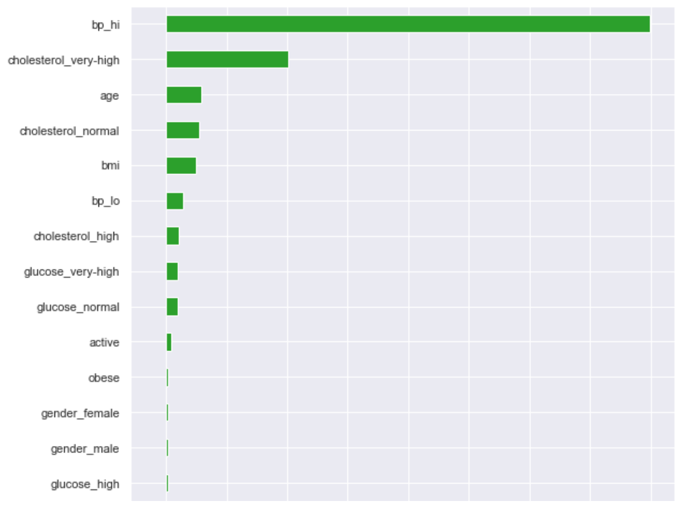
These updated feature importances are shown below:

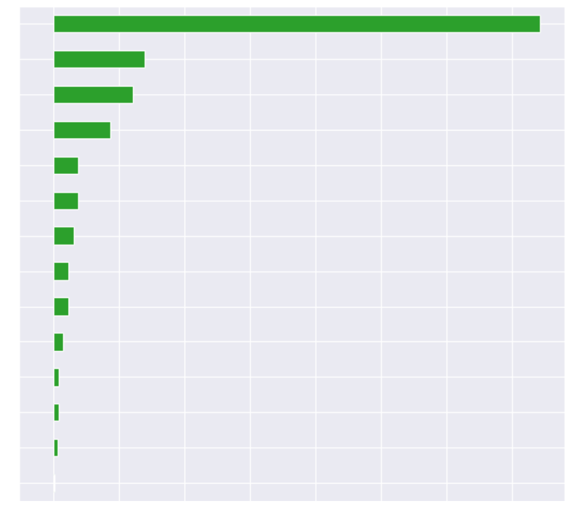

Best Maximum Depth
bp_hi
cholesterol_very-high
age
cholesterol_normal
bmi
bp_lo
cholesterol_high
glucose_very-high
glucose_normal
active
obese
gender_female
gender_male
glucose_high
alcohol
smoke
hypertension
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Mean Decrease in Impurity

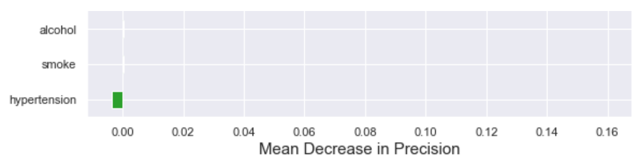

Best Minimum Samples per Leaf
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Mean Decrease in Impurity

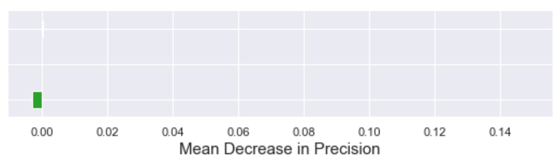

bp_hi
cholesterol_very-high
age
cholesterol_normal
bmi
bp_lo
cholesterol_high
glucose_very-high
glucose_normal
active
obese
gender_female
gender_male
glucose_high
alcohol
smoke
hypertension
0.00
0.02
0.04
0.06
0.08
0.10
Mean Decrease in Recall


0.00
0.02
0.04
0.06
0.08
0.10
Mean Decrease in Recall


bp_hi
cholesterol_very-high
age
cholesterol_normal
bmi
bp_lo
cholesterol_high
glucose_very-high
glucose_normal
active
obese
gender_female
gender_male
glucose_high







The updated feature importances shown above indicate that creating the `obese` and `hypertension` features somewhat decreased the overall loss in recall (as calculated by permutating features). However, the problem of features being detrimental to recall persists. Also, the `hypertension` column itself became detrimental to both recall and accuracy, and possibly should have been removed. This strongly indicates that engineering `bp_categories` was unwise.

Overall the most important features were:

> - Systolic Blood Pressure (`bp_hi`)
> - Cholesterol Level (`cholesterol_normal` , `cholesterol_high` , and `cholesterol_very-high`)
> - Age (`age`)
> - Body Mass Index (`bmi`)
> - Diastolic Blood Pressure (`bp_lo`)
> - Glucose Level (`glucose_normal` , `glucose_very-high`)

Evaluation

> *The models with the best scores overall were the `knn_best` and `bagged_knn_best` , however `bagged_knn_best` generalizes slightly better. Therefore, implementing a KNN Classifier with bagged sampling proved to have a slight edge over its competitors.*

None of the models shown here, however, meet the requirements for use as an integrated medical screening test. The recall score is too low across all models, which is the most important to maximize in this situation. The models all produced similar validation metrics when making predictions against the test set, indicating more data may need to be collected and features added/removed. All the models discussed here are good candidates for further development because they all generalized well tothe training set.

Limitations & Ideas for Further Analysis

As discussed above, none of the models are ready for use as a screening tool for cardiovascular disease because their recall (and accuracy) scores need to be improved. Theses results are promising though, and effort needs to start shifting towards gathering more data and further training these models. It would be prudent to access the API of awidely-used EHR system and start gathering data that is the same as, and similar to, the features used in this analysis.

It is recommended that a consultant with domain-specific (i.e. a cardiologist) be brought in to help make sense of features, point out highly correlated features, recommend new features, and aid with feature selection. This is vital for model improvement.

As seen from the plots showing feature importances, features like `smoke` , `alcohol` , and `active` are too vague to be useful as predictive variables. If a patient is to answer these questions, they should respond with a quantity instead of a yes/no answer. This would benefit the accuracy of the models at the very least.

Finally, combining the different models into a single classifier by feeding them into ensemble methods shuch as a Voting Classifier or Stacking Classifier could have improved results.

Further Information

Review the full analysis in the Jupyter Notebook or the view the Presentation.

For any additional questions, please contact:

> **Suleyman Qayum (sqayum33@gmail.com)**

Repository Structure

```
├── data
    └── cardio_disease_data.csv
├── images
    ├── bagged-knn-best.pmg
    ├── bp_index.png
    ├── feature-importances-1.png
    ├── feature-importances-2.png
    ├── knn-best.png
    ├── logreg-best.png
    ├── obesity_index.png
    ├── rfc-bestMaxDepth.png
    └── rfc-bestMinSamplesLeaf.png
├── Predicting_Cardiovascular_Disease_Presentation.pdf
├── predicting-cardiovascular-disease.ipynb
├── classifier_utils.py
└── README.md
```