HEART DISEASE PREDICTION PROGRAM

HOOTY-HOO SOFTWARE SOLUTIONS

Joshua Jarabek

WESTERN GOVERNORS UNIVERSITY | CAPSTONE PROJECT

HEART DISEASE PREDICTION PROGRAM

TABLE OF CONTENTS

HEART DISEASE PREDICTION PROGRAM

HEART DISEASE PREDICTION PROGRAM

HEART DISEASE PREDICTION PROGRAM

LETTER OF TRANSMITTAL

To Sr. Management at Sage Owl MC,

Sage Owl Medical Center and the City of Hootsville have been seeing an uptick in heart disease. There have been a lot of cases that have flown under the radar these past few years, which have mostly been due to patients not experiencing any symptoms. Sage Owl MC has struggled to catch the silent cases of heart disease, and too many have been taken by it. Due to Covid-19, a lot of our local community have been living more sedimentary lifestyles and they are not as active as they once were. Food delivery apps, such as Uber Eats and Door Dash, have become mainstream, and fast food is only a few taps away whenever you want.

Sage Owl Medical Center needs a solution that will allow you to catch heart disease better for all your patients that come to Sage Owl Medical Center, not just the ones who have symptoms. Sage Owl Medical Center needs to implement a quick and easy solution that all your doctors can utilize for all their patients. Whether that's your general physicians or your specialized physicians, every patient that comes to Sage Medical Center should be screened for heart disease at no extra cost to them. When patients check-in, they use your tablets to answer questions about themselves and their medical history. I can create a solution that can integrate seamlessly with your already implemented digital solution. This solution is a data product that uses bulk data that is specific to heart disease. This software will use machine learning and train itself using the data that's been collected from many people. With enough data, I believe this solution will be able to predict whether a patient has heart disease, even if they don't show any symptoms.

This proposed solution benefits this organization by helping you do what you came here to do every day, which is saving lives. Not only will this help you catch heart disease better, but it will also free up time for a lot of your physicians. A lot of people with heart disease tend to get sick more often and they can show a wide range of symptoms. This could potentially lead to misdiagnosing people when it's heart disease. This will not only help people with heart disease get better, but you will be able to see more patients due to the freed-up time. This means patients will be able to get in to be seen quicker, and the quicker they're seen, the quicker they can also get better.

This project should also be incredibly cost-effective. I believe we can get this project done in less than a month, with a budget of only $10,000. There are plenty of free and open-source datasets in the medical field, with organizations such as the CDC releasing new and up-to-date data sets frequently.

At Sage Owl University, I spent countless hours working with machine learning and A.I.

I have built products for other organizations, such as chatbots and software for autonomous robots. Software like this will be incredibly easy and simple to build and deploy at your medical center. If this is of interest to your organization, my contact information is below.

Josh Jarabek (Software Engineer at Hooty-Hoo Hardware)

Jjarab4@student.wgu.edu | 469-767-6723

HEART DISEASE PREDICTION PROGRAM

PROJECT RECOMMENDATION

To Sr. Management at Sage Owl MC,

Thank you for reaching out about this opportunity, I believe it will make a big difference for your organization. Below, I have provided more details on how this project will meet your organization's needs.

PROBLEM SUMMARY

THE PROJECT (RECAP)

To recap, this project will be data-based. I will use a large dataset that involves health data that is specific to heart disease. It will include many different features, such as age, body mass index, health problems, if they smoke, and many other things. It will also feature whether that person has heart disease. With a large enough dataset, I can create a machine learning software that will use that data to be able to predict whether somebody has heart disease.

WHY THIS PROJECT IS NEEDED (RECAP)

This project will be used through a computer's command-line interface, which can easily be applied to your medical center's current application for when a patient first signs in and inputs their medical history. It is a cost-effective product and will add very little time to a patient's medical history submission. This project is much needed, especially at Sage Owl Medical Center and the City of Hootsville, as there has been a steady increase in heart disease here. This will allow you to detect heart disease more effectively and efficiently and will free up time for your physicians to focus on other things.

HOW IT MEETS YOUR NEEDS (RECAP)

This project will meet your medical center's needs by creating a cheap and effective product that will barely scratch your annual budget. Once you own it, you own it. There won't be any additional monthly or yearly fees to continue using it. I believe this product has the potential to save countless lives of those whose heart disease has gone undetected or will go undetected.

DELIVERY AND ACHIEVEMENT (RECAP)

The software will be delivered as a user-friendly command-line interface program, where all the users must do is follow the directions and information on the screen. They will open the program, hit run, and then be guided through the entire process. I understand that this is a medical center and there's a chance that most end-users might not be as tech-savvy, and that's why I will make sure to create a program that is simple, easy to use, and most importantly, user-friendly. This will allow your medical center to achieve quick and accurate prediction results of whether a patient is at risk of heart disease.

HEART DISEASE PREDICTION PROGRAM

APPLICATION BENEFITS

HOW IT MEETS YOUR NEEDS

This application meets your needs by helping you solve the problem of heart disease flying under the radar. On top of that, this application is extremely secure, as it stores no sensitive patient health data. Your medical center can simply plug in the information about a patient and receive an answer regarding heart disease.

HOW YOU BENEFIT

Sage Owl Medical Center will benefit from this application by giving you the ability to predict and treat heart disease across the entire organization. If a patient who shows no symptoms visits their general physician for their yearly checkup, their medical information will go through the application and verify whether they have heart disease. This will help prevent silent cases of heart disease from going untreated. This will help free up time across the board for all physicians by preventing a misdiagnosis where the patient might end up coming back frequently, taking up appointment spots from others. Less time waiting for appointments means patients can be seen faster.

APPLICATION DESCRIPTION

TECHNICAL SOLUTION

The Heart Disease Prediction Program will be written using the Python programming language with the use of a few external libraries as well. More specifically, we are using Python 3.10 along with Pandas, NumPy, Scikit-Learn, Pickle, Joblib, Matplotlib, and Seaborn. The machine learning models will be tested throughout Jupyter Notebooks and will be used to determine the final solution. Once that solution is found, the program will be written in DataSpell IDE to develop the CLI program. The Python file can be run from any IDE that supports Python, but DataSpell is convenient as it supports both Jupyter Notebook, as well as regular Python files. This will help speed up the development process. Python is a great language for data science, and therefore I've chosen it as the programming language for this program.

DATA DESCRIPTION

THE ORIGIN

The origin of the raw data comes from a site known as Kaggle. It's an open-source data science website that allows data scientists to work together and post datasets that they've put together. The data set I've chosen comes from someone in the online community that took the dataset straight from the CDC in 2020 and then cleaned it. Data cleaning is a very tedious and lengthy part of the machine learning process, so having it precleaned will dramatically reduce the time and cost it takes to develop this program.

THE TYPE

This data is a mix of multiple different data types. Some are quantitative, such as age or body mass index, while others are nominal, such as sex or race.

HEART DISEASE PREDICTION PROGRAM

X & Y

The dependent variable (Y) for this program is heart disease. The independent variables (X) are quite large in number. They include BMI, Smoking, Alcohol, History of Stroke, Injury/Illness (Past 30 Days), Mental Health (Past 30 Days), Difficulty Walking, Sex, Age Category, Race, Diabetes, Physical Activity, General Health, Average Sleep, Asthma, History of Kidney Disease, and History of Skin Cancer.

ANOMALIES AND LIMITATIONS

This data set, since already cleaned, is free of any anomalies. However, this does not mean that each independent variable in the data set is of significance. That is why I will test the relationships between each independent variable with the dependent variable before training the models. If an independent variable is not of significance, it can end up creating noise, so I will drop those variables from the dataset before moving forward with the model training.

Since the data set is of multiple different data types, this creates some limitations. I will encode and standardize any data that isn't quantitative.

OBJECTIVES AND HYPOTHESIS

DESIRED OUTCOME

The desired outcome of this project will be to create a program that is better than the base case of predicting whether somebody has heart disease.

HYPOTHESIS

Machine learning can be more accurate at predicting if a person has heart disease than the base case, which has a 50% accuracy rate.

ACCURACY

The desired prediction accuracy is to predict correctly whether somebody has heart disease at least 70% of the time. The program will lean more towards giving a false positive to minimize false negatives. It's better to tell someone they have heart disease when they don't than to tell someone they don't have heart disease when they do.

HEART DISEASE PREDICTION PROGRAM

METHODOLOGY

DEVELOPMENT AND IMPLEMENTATION

The methodology I will use for the development and implementation of this program will be the KDD method.

WHY THIS METHODOLOGY?

I've chosen to use the KDD method because it's a classic data science life cycle, and it's able to purge the noise while establishing a phased approach to derive patterns and trends that add important knowledge. Since this data set will have many independent variables, we want to make sure we can get rid of many insignificant variables as possible.

PHASES

KDD is a 5-phase process:

1. Selection
    a. This involves selecting the right dataset, target, and variables for the project.
2. Pre-Processing
    a. This step is all about improving the data by cleaning it and fixing any faulty, missing, or mismatched data in the data set.
3. Transformation
    a. Concentration on converting the pre-processed data to the fully utilizable kind. This is where the data is encoded and standardized.
4. Data Mining
    a. This is the most known aspect of the process. It is where we harness the transformed data to seek out the patterns of interest. This is where we create the machine learning models. In this case, we are creating models from Naïve Bayes, Logistic Regression, and Random Forest algorithms.
5. Interpretation/Evaluation:
    a. This final phase is where we are evaluating the 3 different models that have been created, to determine which one best fits the application.

HEART DISEASE PREDICTION PROGRAM

FUNDING REQUIREMENTS

WHAT'S REQUIRED?

The required funding for this project will be $10,000. This program will be able to run on any computer that has Python installed. The only funding required for this project is labor, as there are no monthly or yearly fees. However, the costs are low because this program cannot be distributed or sold to other organizations. It's for strict use by Sage Owl Medical Center only.

DATA PRECAUTIONS

SENSITIVE AND/OR PROTECTED DATA

Since this application will not be storing any data, there are no risks of sensitive or protected data breaches. The data set that is being used is publicly accessible to anybody, and the names of the patients in the data set are anonymized. Sites like Kaggle are public, and therefore, have no restrictions or protection data.

GENERAL GUIDELINES

Some general guidelines when working with this data would be to make sure that only the designated personnel have access to the program and most importantly, they don't leave their computer while a patient's information input has not been fully completed.

DEVELOPER'S EXPERTISE

WHO AM I?

I have experience with machine learning applications through my academic training and Computer Science degree from Western Governors University. I've created numerous projects that involved both Python and machine learning. These projects include things like creating software for autonomous robots, as well as creating chatbots using artificial intelligence.

WHY I'M NEEDED

My expertise in this subset is needed for this project as it involves the use of several different types of machine learning algorithms. Just because a program uses machine learning, does not mean it is the best algorithm for the program, as each project is different, and these algorithms are not created equal. My skillset in machine learning, as well as Python, makes me qualified to complete this project.

HEART DISEASE PREDICTION PROGRAM

PROJECT PROPOSAL

PROBLEM STATEMENT

WHAT IS THE PROBLEM AT HAND?

The problem at hand is that Sage Owl Medical Center is struggling to detect heart disease at the desired level. Too many cases are going on undetected, as plenty of patients do not show the signs or symptoms of heart disease. Physicians do not check for heart disease unless they show signs or symptoms, as this would be cumbersome to not only the physicians but the patients as well. It would become an added expense and consume unnecessary time for both the physicians and the patients. Sage Owl Medical Center needs a simple, quick, and effective solution to its problem.

CLIENT SUMMARY

WHO ARE YOU?

Sage Owl Medical Center is the largest medical center in Hootsville. Area residents prefer coming to Sage Owl Medical Center for all their medical needs, as they can view all their medical information and documents in one place. If a patient sees their general physician for their yearly checkup, they can be referred to a specialist within the same building if something abnormal is found, and they can usually be seen by them immediately after their yearly checkup.

WHY THIS APPLICATION?

Sage Owl Medical Center needs a solution that can be easily integrated into its already existing medical portal/platform. This machine learning application helps detect heart disease in patients who may or may not show signs or symptoms of heart disease. This solution adds no extra cost to the patients and the only extra cost to the medical center is the upfront cost of building the application. It will save money, and time, and most importantly, it will help save lives.

EXISTING SYSTEM ANALYSIS

THE TOOLS YOU HAVE

The tools you have are computers compatible with running Python applications. This is the only hardware requirement for this application.

THE TOOLS YOU NEED

If the computers do not have Python installed, then they will need to be installed to run the application.

HEART DISEASE PREDICTION PROGRAM

DATA

THE DATA SET

The data set used will be a data set of more than 30,000 anonymized public health records. It will feature many different categories of a person's health records, which include things like their body mass index, age, sex, history of skin cancer, history of kidney disease, average sleep per night, and, whether they have heart disease or not, as well as many others.

COLLECTION, PROCESSING, AND MANAGEMENT

The data will be collected from the online data science website known as Kaggle. This data will already be cleaned by other data scientists, which will greatly shorten the time it takes to develop this application.

HANDLING ANOMALIES

Since this data will already be clean of any incomplete data, the anomalies will be minimal. However, during the development process, I will use Jupyter Notebooks to determine which features in the machine learning model can be dropped due to being insignificant to the dependent variable. Including insignificant features in the model can increase the noise of the model, causing the model to become less accurate in its predictions. Therefore, it's important to make sure that each feature, or independent variable, in the model is significant in helping predict whether somebody has heart disease.

HEART DISEASE PREDICTION PROGRAM

PROJECT METHODOLOGY

THE INDUSTRY STANDARD

The industry-standard methodology that will be used to develop this application is the KDD methodology.

DEVELOPMENT PHASES

KDD is a 5-phase process:

1. Selection
    a. This involves selecting the right dataset, target, and variables for the project.
    b. I will determine which independent variables are significant in the prediction model.
    c. If an independent variable is not significant, I will throw it out of the data set to minimize noise and have a more accurate prediction model.
2. Pre-Processing
    a. This step is all about improving the data by cleaning it and fixing any faulty, missing, or mismatched data in the data set.
    b. Since the data set is already pre-cleaned, this step should be quick and easy.
3. Transformation
    a. Concentration on converting the pre-processed data to the fully utilizable kind. This is where the data is encoded and standardized.
    b. Some of the data in the data set might not be a 0 or a 1. Features such as a history of heart disease, kidney disease, or if they smoke are features that will be a 0 or a 1, but features such as race, sex, or age will not be a 0 or 1. Every feature must be of the exact data type and scaled to the same scale, or else the prediction model will be inaccurate.
    c. For features such as race, each race will be split up into its feature, and therefore, will be able to be a 0 or 1.
4. Data Mining
    a. This is the most known aspect of the process. It is where we harness the transformed data to seek out the patterns of interest. This is where we create the machine learning models. In this case, we are creating models from Naïve Bayes, Logistic Regression, and Random Forest algorithms.
5. Interpretation/Evaluation:
    a. This final phase is where we are evaluating the 3 different models that have been created, to determine which one best fits the application.
    b. We will be choosing the model that has the best balance between predicting both yes and no, as well as overall accuracy. Some models might be very good at predicting a, yes, but fall incredibly short when predicting a no. Some models might be equal in predicting both yes and no, but the overall accuracy is low. We are looking for a model that has a high accuracy rate of predicting both yes and no.
    c. Once the best model is found, I will transfer the code over to a Python model and create the CLI application.

HEART DISEASE PREDICTION PROGRAM

PROJECT OUTCOMES

FINISHED APPLICATION

The finished application will be a CLI application that is user-friendly and accurate at predicting whether a patient has heart disease at least 70% of the time. This is better than the base case of predicting heart disease, which has a 50% accuracy. It will be a Python application that can be run on any computer that has Python installed. The user will be able to run the application by hitting the run button, and they will be guided through the entire process once the program begins.

USER GUIDE

The application will also come with a user guide that offers an installation guide for Python, as well as a guide to running and using the application itself.

IMPLEMENTATION PLAN

GENERAL STRATEGY

A general strategy for implementing the application will be to increment the rollout of the application amongst various physician offices within Sage Owl Medical Center. The application will be heavily tested for any bugs, but this will allow the bugs, if any, to be caught before rolling out the application to the entire medical center at once. The best strategy would be to roll out the application from the least-busy office to the most-busy office. Trying to implement this application in an extremely busy office could cause problems if there's a bug, so it's best to test this application in an office that can afford to experience a bug first.

ROLLOUT PHASES

This application should be able to be fully deployed in less than a month, so the rollout will be done in four separate phases over 4 weeks.

1. Week 1:
    a. Rollout to the bottom 25% of physician offices, in terms of office traffic.
2. Week 2:
    a. Rollout to the next 25%, in terms of office traffic, which will total 50% of the entire organization.
3. Week 3:
    a. Rollout to the next 25%, in terms of office traffic, which will total 50% of the entire organization.
4. Week 4:
    a. Rollout to the remaining 25%, which will complete the organization-wide integration of the application.

HEART DISEASE PREDICTION PROGRAM

DEPENDENCIES

Dependencies would be that the offices in which the application is being rolled out have computers with Python installed.

TESTING AND DISTRIBUTION

With each rollout, if a bug is found, the employee should email me what happened, and how I can recreate the bug that was found. After the bug is fixed, we can continue the rollout after swapping the previous application with the updated version on the computers that it's already been rolled out to.

EVALUATION PLAN

VERIFICATION

To evaluate the application during the rollout, there are two steps, which are verification and validation. In terms of verification, we will provide surveys from the end-users of the application daily. Since this application works with sensitive data of your patients, it's important that those answering the surveys strictly answer the fields from a general standpoint, without giving any specifics, which might include sensitive data.

VALIDATION

We are looking for an acceptance rate of over 90% from the end-users of the application. Once that is reached, we can consider the application validated. Otherwise, there may or may not be tweaks done on my end to improve the application and increase the acceptance rate to reach validation.

RESOURCES AND COSTS

HARDWARE AND SOFTWARE

- Hardware:
  - Computers in each physician's office that can run Python 3.10
- Software:
  - Python 3.10 installed on each computer

LABOR TIME AND COSTS

- The only labor cost would be the flat rate of $10,000 for me to create this application.
- This application should take less than a week to develop and begin rolling out.

ENVIRONMENT COSTS

- If a computer cannot run Python 3.10 and needs to be updated, this would be an environmental cost. Other than this, there are no environmental costs to this program, as everything is free and open source.

HEART DISEASE PREDICTION PROGRAM

TIMELINE AND MILESTONES

PROJECTED TIMELINE

Event	Start Date	End Date	Developer Hours
Project Start	August 1st, 2022	August 1st, 2022	0
KDD: Selection	August 1st, 2022	August 6th, 2022	40
KDD: Pre-Processing	August 6th, 2022	August 12th, 2022	40
KDD: Transformation	August 12th, 2022	August 17th, 2022	12
KDD: Data Mining	August 17th, 2022	August 22nd, 2022	12
KDD: Evaluation	August 22nd, 2022	August 27th, 2022	12
Program Development	August 27th, 2022	August 31st, 2022	12
Rollout: Phase 1	September 1st, 2022	September 7th, 2022	0 (Unless bugs are found)
Rollout: Phase 2	September 7th, 2022	September 14th, 2022	0 (Unless bugs are found)
Rollout: Phase 3	September 14th, 2022	September 21st, 2022	0 (Unless bugs are found)
Rollout: Phase 4	September 21st, 2022	September 28th, 2022	0 (Unless bugs are found)
Project End	September 28th, 2022	September 28th, 2022	0

HEART DISEASE PREDICTION PROGRAM APPLICATION

SUBMITTED FILES

The application files being submitted are as followed:

- **Directory: Heart Disease Prediction Application**
  - *Application Folder:*
    - Application_Code.py
      - The code was created to have a working CLI application, which will be run from the Main.py file.
    - Train.py
      - The code was taken from the Jupyter Notebook to create a prediction model using a Random Forest algorithm. This model is then transferred to Application_Code.py using Pickle.
  - *Data Folder:*
    - heart_disease.csv
      - This file contains the data set used to train the models in the Train.py file
  - *Models Folder:*
    - model_rf.pkl
      - This file contains the trained random forest model that was pickled in Train.py and will be used in the Application_Code.py file.
    - scaler.pkl
      - This file contains the scaler for the random forest model and was also pickled in Train.py and will be used in the Application_Code.py file.
  - *Notebook Folder:*
    - Heart Disease.ipynb
      - The Jupyter Notebook contains visualizations, descriptive methods, non-descriptive methods, etc.
      - This notebook was used to test and evaluate the different algorithms to choose the best one for the final program.
  - *Main.py File*
    - This file sits at the top of the directory and is used to run the program.
    - With this file open, the end-user can hit run Main.py, and the application will initialize and run.

HEART DISEASE PREDICTION PROGRAM
POST-IMPLEMENTATION REPORT

VISION

THE PROBLEM OF THE PAST

The problem that Sage Owl Medical Center faced was that they were struggling to detect heart disease in their patients. Plenty of patients in the past had shown no sign of heart disease but eventually succumbed to the disease. They needed a way to detect heart disease on a massive scale, without dramatically increasing their or their patients' expenses or the time it takes to be seen and treated.

HOW IT WAS SOLVED

This problem was solved by creating an application using machine learning to predict heart disease in every patient that is seen at Sage Owl Medical Center. The solution was quick, easy, and effective. It adds a minuscule amount of time and is implemented in the sign-in process that was already in place at Sage Owl Medical Center. It can predict heart disease in any of its patients over 75% of the time. This application's costs were a fraction of their yearly budget.

HOW IT WAS USED

When a patient signs into their physician's office, they input a wide variety of their medical history and information. When that data is submitted, the person at the front desk can read the information they have submitted and can input the corresponding data into the program. Once they are given a result from the application, the employee can mark the patient's file on whether they have heart disease. Their physician can then see this mark when it's time to see that patient. If they were marked for heart disease, the physician can then refer them to a cardiologist within Sage Owl Medical Center. Sage Owl Medical Center has seen a dramatic rise in its ability to catch heart disease in a patient since implementing this program. Many of these patients showed no signs or symptoms of having heart disease.

HEART DISEASE PREDICTION PROGRAM

DATASET

WHAT DATA WAS USED

The data set used contained over 30,000 anonymized health records with a variety of variables. The dependent variable used was the heart disease column, which indicated whether this person had been diagnosed with heart disease. The independent variables include things like age, sex, body mass index, whether they were a smoker if they had a history of skin cancer or kidney disease, their race, sleep habits, and many more features.

EXAMPLE

	HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	S
0	No	16.60	Yes	No	No	3.0	30.0	No	
1	No	20.34	No	No	Yes	0.0	0.0	No	
2	No	26.58	Yes	No	No	20.0	30.0	No	
3	No	24.21	No	No	No	0.0	0.0	No	
4	No	23.71	No	No	No	28.0	0.0	Yes	

ACCESS TO THE DATASET

The data set used is included in the application's directory. It is found within the "Data" folder under the name "heart_disease.csv".

SECURITY

The data's security suits the needs of my project because it is open-source health data that has been anonymized in terms of a patient's identity. This also helps with the application development, as my client was a medical center and works with sensitive data. Being able to create an application that predicts heart disease in their patients without having to worry about storing their patients' data in the application was optimal. Not having to store their patients' data in the application means there's no risk of the application having a data breach.

HEART DISEASE PREDICTION PROGRAM

DATA PRODUCT CODE

FUNCTIONALITY REVIEW

Since the data had been cleaned already, there was minimal to do in terms of processing raw data. However, I did still require some analysis when it came to finding out which features were significant in determining whether a person has heart disease. Comparing features to the heart disease variable using bar plots, box plots, and KDE plots was crucial in determining which independent features helped predict the dependent variable.

NON-DESCRIPTIVE REVIEW

The non-descriptive methods used were Logistic Regression, Random Forest, and Naïve Bayes. These methods were appropriate for the project at hand because they were a mix between regression and classification. I did not know which type would perform better, so I tested each of the three and analyzed their performance using a confusion matrix, ROC curve and AUC, and a precision-recall curve. They were developed using the data set after it had been improved, encoded, and standardized. They were then trained and tested using an 80/20 ratio.

CHOICES AND IMPROVEMENT

The analysis methods used were confusion matrix, ROC curve and AUC, and precision-recall curve. The confusion matrix was chosen because it allows for the visualization of an algorithm's performance. It shows the number of real positive cases in the data, as well as the number of real negative cases in the data, paired with the test data from the various machine learning algorithms I used it on. The ROC curve and AUC allow me to see the algorithm's performance through a curve, rather than a matrix. It also helps determine the overall performance paired with individual performance to see if an algorithm is more accurate at determining a positive rather than a negative, or vice-versa. Finally, the precision-recall curve allows me to see another curve visual that shows me the hit rate of an algorithm. This model also shows me the overall, as well as individual performance of an algorithm, which shows if an algorithm is heavily weighted towards correctly predicting a true positive while lacking in predicting a true negative, or vice versa.

These analysis methods helped me decide which machine learning algorithm worked best with the data. Naïve Bayes came in last place, with it being incredibly inaccurate at predicting the data. Logistic Regression and Random Forest were incredibly close in terms of each model's accuracy, but I went with Random Forest, as it provided more false positives, at the expense of having more false negatives. I found this to be more ideal, as it's better to tell a patient they have heart disease when they don't, as opposed to telling a patient they don't have heart disease and they do.

HEART DISEASE PREDICTION PROGRAM

SOURCES

The data set file is in the application directory, however, the link for the data set is below:

Personal Key Indicators of Heart Disease

OBJECTIVE VERIFICATION

WHAT WAS THE OBJECTIVE?

The objective of this project was to create a machine learning model that can predict heart disease at a rate better than the base case, which is 50%.

WHY IT WAS MET

The objective for this project was met because the machine learning model can predict heart disease at a rate of 74%, which is a near 50% increase from the base case.

HEART DISEASE PREDICTION PROGRAM

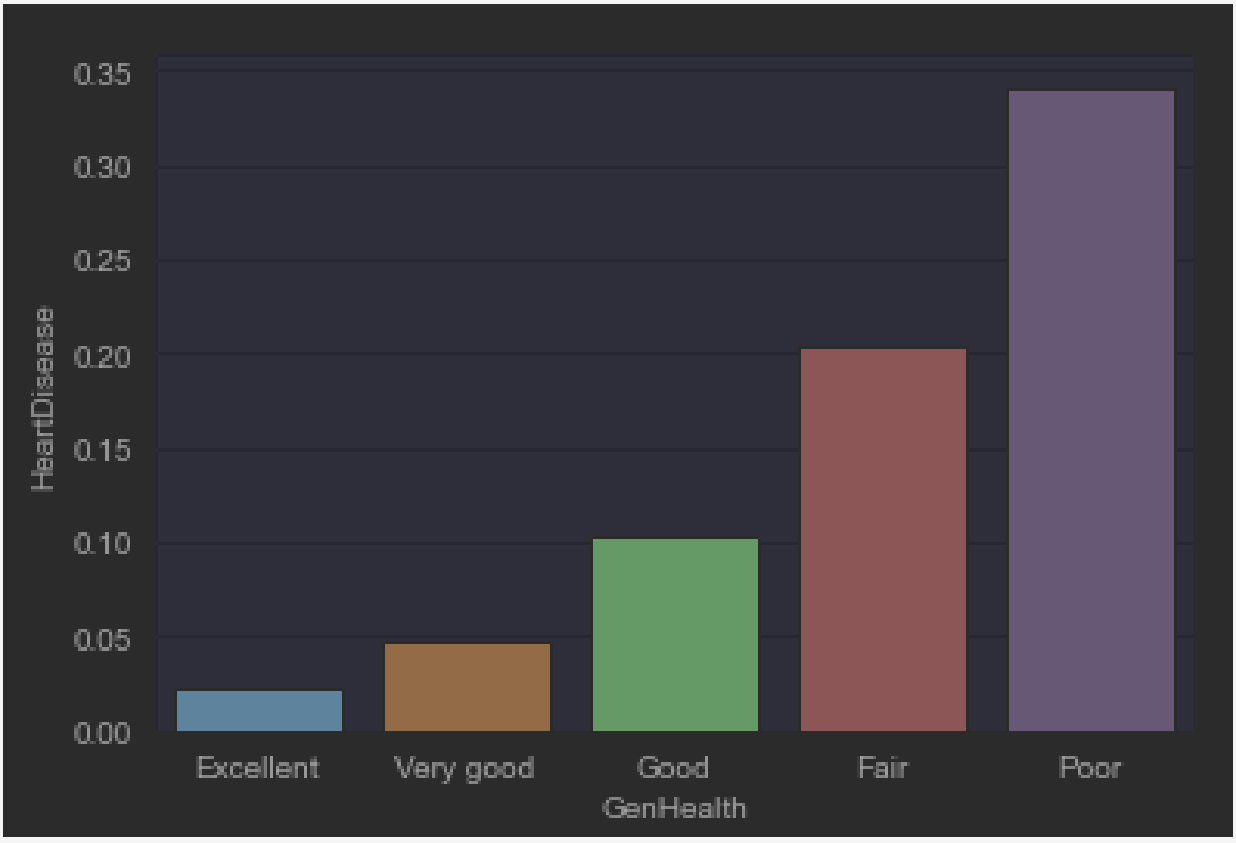
EFFECTIVE VISUALIZATION AND REPORTING

DESCRIPTIVE METHODS AND VISUALIZATIONS

The descriptive methods and visualizations supported my non-descriptive methods development process by helping me explore, analyze, and summarize the data set. This helped me determine which features to keep and which features to drop. Fewer features mean less noise and less code, allowing me to create a more accurate prediction model, and write the program even quicker. My client was a hospital and the program being developed was used to save lives, so the more accurate the model and the faster the development, the better.

VISUALIZATION #1

With this visualization, I was able to see that the worse a person's general health is, the more likely they are to have heart disease. This is helpful when encoding and standardizing the data because each category of health data is weighted differently toward a person having heart disease.



HEART DISEASE PREDICTION PROGRAM

VISUALIZATION #2

This visualization is important, as I was able to drop this feature from the final data set. I found that heavy use of alcohol represented a very small portion of the data set, and on top of that, the difference in whether it causes heart disease was also very small. So, knowing both of those, I decided to drop this feature from the data set.

Alcohol

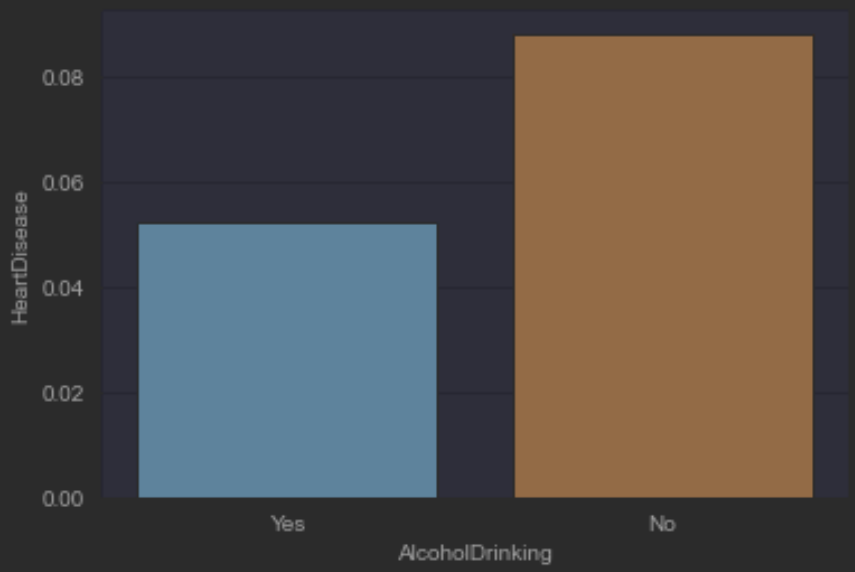
```
df['AlcoholDrinking'].value_counts()/l*100
```

```
No     93.190325
Yes     6.809675
Name: AlcoholDrinking, dtype: float64
```

```
t5=dummy_df[["HeartDisease","AlcoholDrinking"]].grou
```

```
sns.barplot(data=t5.sort_values(by='HeartDisease'),x
```

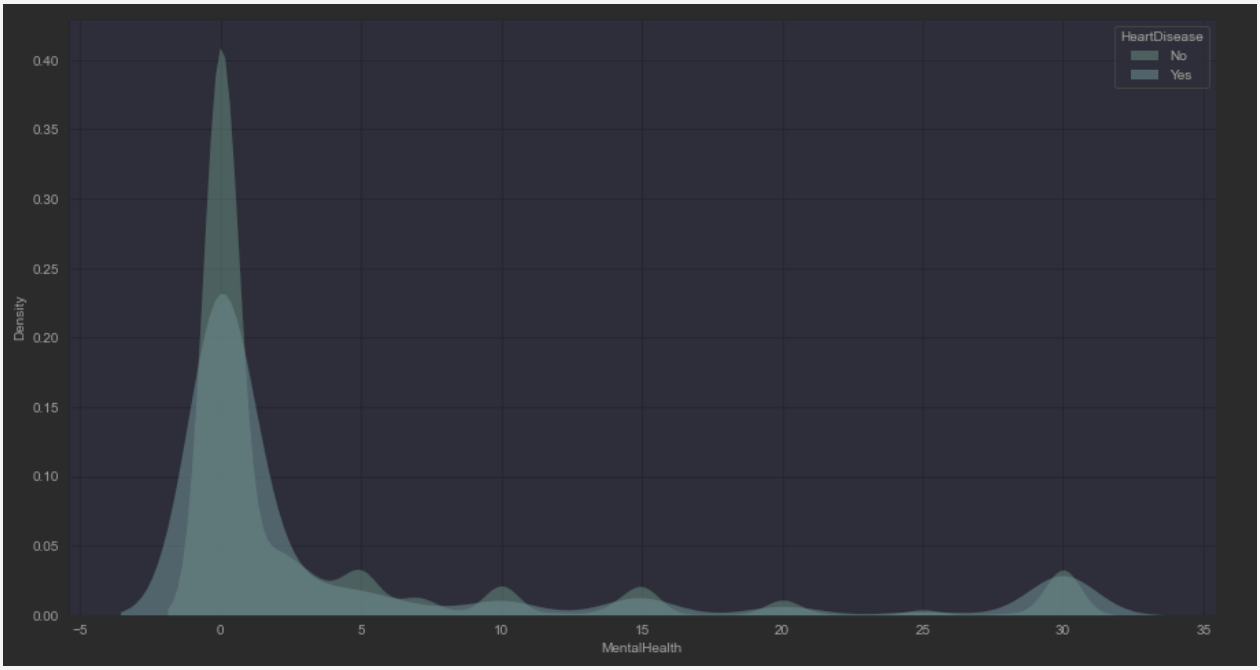
```
<AxesSubplot:xlabel='AlcoholDrinking', ylabel='Hea
```



HEART DISEASE PREDICTION PROGRAM

VISUALIZATION #3

This visualization was important in determining whether to drop the Mental Health feature of the data set. Using this plot, I was able to find that bad mental health is not useful in determining whether someone has heart disease. In fact, in some cases, people with better mental health had a higher rate of heart disease than people with worse mental health. Knowing this, keeping the mental health feature of the data set would have created unnecessary noise in the prediction model, leading to a less accurate model.



HEART DISEASE PREDICTION PROGRAM

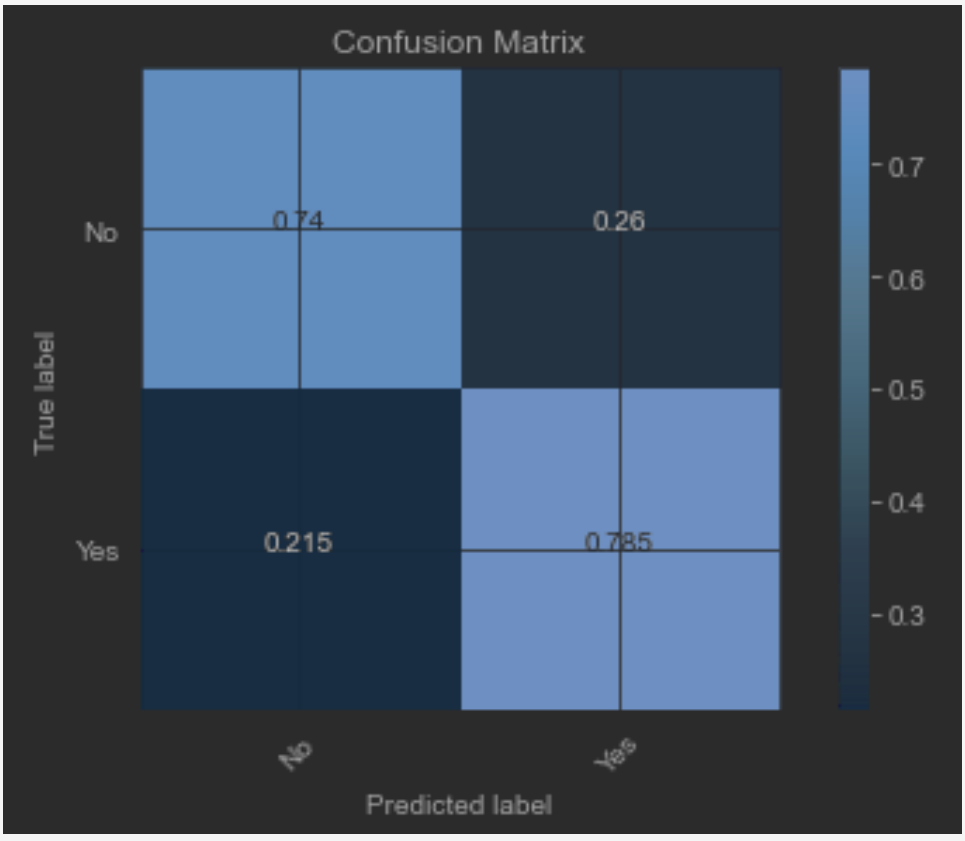
ACCURACY ANALYSIS

DESCRIPTION

Using the 3 accuracy analysis methods, the confusion matrix, the ROC curve, and the precision-recall curve, I was able to determine and assess the accuracy of my non-descriptive methods. These allowed me to see the accuracy rate for both a positive case of heart disease, as well as a negative case of heart disease. It also allowed me to see the rate of false positives and false negatives.

EXAMPLE

Below is an example of the report for the Random Forest algorithm using the confusion matrix. The y-axis is the True label, or in other words, the real heart disease value for the test set. The x-axis is the Predicted label, or in other words, what the random forest algorithm had predicted the result to be. You can see the algorithm was right at determining a correct positive result over 78% of the time and a correct negative result 74% of the time.



HEART DISEASE PREDICTION PROGRAM

APPLICATION TESTING

DESCRIPTION

The application was tested by running through the application more than 50 times using different data values for each run. I also tested it multiple times using data that should give me a positive result for heart disease, as well as tests that should give me a negative result for heart disease.

EXPLANATION

Testing helped improve the application, as there were times when I forgot to put a line break in certain areas of a prompt question. I wanted everything to be uniform so that nothing will confuse the end-user.

HEART DISEASE PREDICTION PROGRAM

APPLICATION FILES

HIERARCHY

- **Directory: Heart Disease Prediction Application**
  - *Application Folder:*
    - Application_Code.py
      - The code was created to have a working CLI application, which will be run from the Main.py file.
    - Train.py
      - The code was taken from the Jupyter Notebook to create a prediction model using a Random Forest algorithm. This model is then transferred to Application_Code.py using Pickle.
  - *Data Folder:*
    - heart_disease.csv
      - This file contains the data set used to train the models in the Train.py file
  - *Models Folder:*
    - model_rf.pkl
      - This file contains the trained random forest model that was pickled in Train.py and will be used in the Application_Code.py file.
    - scaler.pkl
      - This file contains the scaler for the random forest model and was also pickled in Train.py and will be used in the Application_Code.py file.
  - *Notebook Folder:*
    - Heart Disease.ipynb
      - The Jupyter Notebook contains visualizations, descriptive methods, non-descriptive methods, etc.
      - This notebook was used to test and evaluate the different algorithms to choose the best one for the final program.
  - *Main.py File*
    - This file sits at the top of the directory and is used to run the program.
    - With this file open, the end-user can hit run Main.py, and the application will initialize and run.

DESCRIPTION

I used user-friendly names for the folders and files to help them understand and navigate where to go. If the main directory, Heart Disease Prediction Application, is open, all the user must do is run the Main.py file to start the application. Otherwise, the data is stored in the Data folder, the models are in the Models folder, the Jupyter Notebook is stored in the Notebook folder, and the application's code is stored in the Application folder.

HEART DISEASE PREDICTION PROGRAM

USER GUIDE

THE STEPS

1. Unzip the project files to the directory of your choice
2. Make sure you have Python 3.10 installed on the computer you will be running the application on.
    a. If you are unsure whether you have Python 3.10 installed, you can open your command prompt or terminal and enter "python -version".
    b. If you do not have Python installed, it can be downloaded at: https://www.python.org/downloads/
        i. Choose the Python download by the computer you are using.
        ii. If you do not have an IDE that supports Python 3.10, be sure to click install IDLE during the Python installation process.
3. Load the project folder into an IDE that supports Python 3.10
    a. Free IDE's for download online include:
        i. PyCharm Community Edition: https://www.jetbrains.com/pycharm/
        ii. VS Code: https://code.visualstudio.com
        iii. IDLE: Should come installed with your Python download
4. Navigate to the python file called 'Main.py' and run this file.
    a. Be sure to only run 'Main.py' and not the entire directory.
5. Once the application starts, you will be greeted with a command-line interface that will walk you through the entire application.
6. Once the application is completed, the command-line interface will output a result, predicting whether that person has heart disease.

HEART DISEASE PREDICTION PROGRAM

SUMMATION OF THE LEARNING EXPERIENCE

HOW MY BACKGROUND READIED ME

My previous experience readied me for this project in numerous ways. The first one is, learning how to code in Python and how to work with the algorithms and data structures that were involved in the project. The second one is, understanding the fundamentals of machine learning and artificial intelligence through my Intro to A.I. course. This course also helped me write this Capstone Write-Up.

ADDITIONAL RESOURCES

Throughout my coursework at WGU, I've found that different resources help with different courses. My main resource throughout this program was YouTube Premium, where I was able to get a student discount. YouTube Premium allowed me to watch anything I needed ad-free on YouTube. Sometimes I don't need to watch an entire 64-hour long Udemy course, because all I need is one tiny piece of it, which YouTube is great for. Another helpful resource is Udemy, which I, unfortunately, did not find out about until my second to the last course. It would have proved helpful for my C++ course, as I felt hopeless in that course. Educative.io has helped me with not only thing project immensely, but also with the coding interviews that I've been taking these past few weeks. There are plenty of free courses on that site once you verify that you're a student, but there are still a few that cost money. Primarily being their bread-and-butter course, otherwise known as "Grokking the Coding Interview". Of course, finally, Stack Overflow, for all my bug-fixing questions.

HOW THIS EXPERIENCE CONTRIBUTED TO MY LIFE

This experience contributed to my concept of lifelong learning as it made me realize how vast the world of computer science is. I came into my capstone thinking that I would have a good foundation to knock it out quickly due to my past coursework. I was very wrong, but that is not necessarily a bad thing. The world of computer science and software engineering is large, and it's not feasible for anybody to know everything there is about computer science. It's a field that is always advancing and it will always be a game of chase. Computer science is larger today than it was yesterday, and tomorrow it will be larger than it was today. For some, that would be terrifying, but for others like me, it's exciting. It's a field that will force you to learn and grow every day, or else you will be left behind.