**Objectifying Disease**

**Heart Disease Cleveland UCI**

Hayden Horn, [HHorn@bellarmine.edu](mailto:HHorn@bellarmine.edu)

Jacob Schelonka, [JSchelonka@bellarmine.edu](mailto:JSchelonka@bellarmine.edu)

**We chose to create multiple experiments using logistical regression, in order to determine whether or not somebody will develop heart disease or not. Our dataset is one based in reality, from the Cleveland ICU. We will be using variables such as age, chest pain type, cholesterol levels, and others from the dataset “Heart Disease Cleveland ICU” on Kaggle.com. In order to create an accurate predicter, we first looked over the dataset and made sure there were no null values. We then split, tested, and trained the data for each experiment. For all 6 experiments we used logistical regression. We did so because we used a non-continuous dataset. The possible answer is 0 (No Disease) or 1 (Disease). We removed “oldpeak”,” thalach”, “chol”, and” trestbps” as they are all continuous data.**

1. **INTRODUCTION**

**We are working on a dataset from Kaggle.com called “Heart Disease Cleveland ICU” that uses real data from a Cleveland ICU. The dataset is mostly noncontinuous independent variables, and we will be attempting to create a predicter that accurately predicts whether or not a patient will have heart disease. We will be using logistical regression on python to create and test these models.**

1. **BACKGROUND**
   1. *Data Set Description*

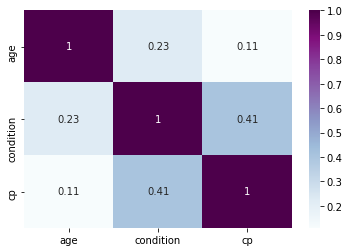
**Our dataset can be found on Kaggle as “Heart Disease Cleveland ICU”, it is made up of 300 entries, and 14 variables. Majority of this dataset is non continuous data, meaning there is only about 2-3 possibilities per category. The category we will be predicting is “condition”, that of which the answer will be either 0, or 1. 0 means there is no heart disease present, and 1 meaning they do have heart disease. Majority of the data correlates, and makes sense, as it is real data from a Cleveland ICU. Therefore, there should be little issues in prepping the dataset. The ICU collected this data in order to objectify which patients will have heart disease, and which of them will not. Obviously, this would be a very useful tool in hospitals.**

* 1. *Machine Learning Model*

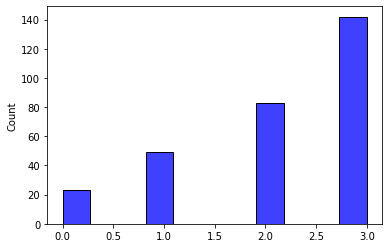
**For this project, we are using logistic regression. This is because we are using categorical data (Mostly 0 or 1 type data). The type of data we are attempting to predict is also non-continuous, which makes this model the easiest to use. Logistic regression uses machine learning to determine our binary outcome. In our case, this outcome is whether or not an individual will develop heart disease, or not. Logistic regression works by measuring the relationship between what we want to predict and our independent variables. Basically, the model is evaluating how much the independent variables effect the chances of getting heart disease, and then it uses all the variables to predict the outcome.**

1. **EXPLORATORY ANALYSIS**

**This data set contains 300 entries, and 14 columns. Mostly consisting of int64 datatypes. This is due to most of the data being non continuous, with only one datatype truly being continuous. Overall, the data is presented well. There are no null values, and very little was done as far as data preparation is concerned.**

**

**Figure 1: correlation of ‘cp’, ‘condition’, and ‘age.’ Figure 2: correlation of ‘fbs, ‘condition’, and ‘rest.’**

**

**Figure 3: ‘cp’ histogram (showing what majority of our data looks like)**

***Most data in this set are similar to this graph, besides the data being excluded.***

Table 1: Data Types

|  |  |
| --- | --- |
| ***Variable Name*** | ***Data Type*** |
| **age** | **Int64** |
| **sex** | **Int64** |
| **cp** | **Int64** |
| **trestbps** | **Int64** |
| **chol** | **Int64** |
| **fbs** | **Int64** |
| **restecg** | **Int64** |
| **thalach** | **Int64** |
| **exang** | **Int64** |
| **slope** | **Int64** |
| **ca** | **Int64** |
| **thal** | **Int64** |

1. **METHODS**

**We used logistic regression for all 6 experiments. However, we changed the perimeters around a bit in order to get a more accurate model. In some cases, we experimented with using different categories that correlated better, we changed the splits, and we also made changes mid-experiment to better the models.**

* 1. *Data Preparation*

**We prepared the data for the model by removing the variables “oldpeak”,” thalach”, “chol”, and” trestbps” as they are all continuous data. We did not need to normalize any data as they are all already in 0 or 1 format.**

* 1. *Experimental Design*

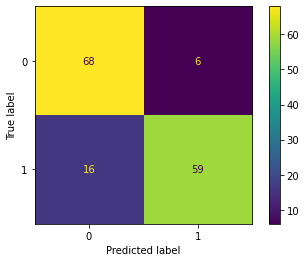
Table X: Experiment Parameters

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | 5 columns: ‘thal’, ’ca’, ’exang’, ’sex’, ’oldpeak’. Split= 0.5. |
| 2 | 4 columns: ‘age, ’fbs’, ’exang’, ’slope’. Split= 0.42. |
| 3 | 3 columns: ‘thal, ’age’, ’restecg’, ’slope’. Split= 0.35. |
| 4 | 5 columns: ‘thal’, ’exang’, ’age’, ’slope’, ‘restecg’. Split= 0.35. |
| 5 | 5 columns: ‘thal’, ’ca’, ’slope’, ’oldpeak’, ‘fbs’. Split= 0.6. |
| 6 | All raw categories. Split = 0.35. |
| 7 | 6 columns: ‘sex’, ’exang’, ’slope’, ’thalach’, ‘age’, ‘cp’. Split= 0.3. |

* 1. *Tools Used*

**The following tools were used for this analysis: Python v3.5.2 running the Anaconda 4.3.22 environment for a mac and Microsoft laptop. Both were used for all analysis and implementation. In addition to base Python, the following libraries were also used: Pandas 0.18.1, NumPy 1.11.3, Matplotlib 1.5.3, Seaborn 0.7.1, SKLearn 0.18.1. We used Python to be able to graph using Seaborn, in order to graph and view correlations in the dataset. We use pandas to view our dataset and get a better analysis of all the data. SKLearn was important in splitting, testing, and training the dataset. It also allowed for us to test our predicters.**

1. **RESULTS**
   1. *Classification Measures*

**

*Experiment 1:*

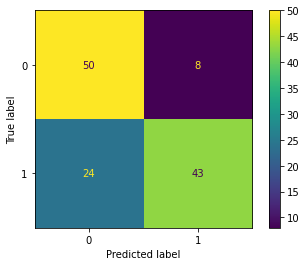
*No heart disease predicted correctly: 59*

*heart disease predicted correctly: 68*

*No heart disease, but predicted to have heart disease: 6*

*Heart Disease, but was predicted not to: 16*

*Accuracy:85%*

**

*Experiment 2:*

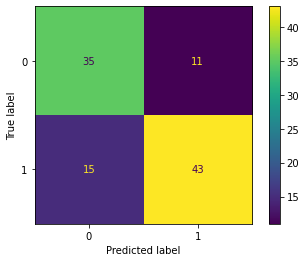
*No heart disease predicted correctly: 43*

*heart disease predicted correctly: 50*

*No heart disease, but predicted to have heart disease: 8*

*Heart Disease, but was predicted not to: 24*

*Accuracy:74.4%*

**

*Experiment 3:*

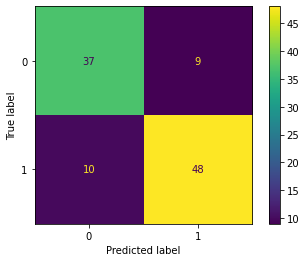
*No heart disease predicted correctly: 43*

*heart disease predicted correctly: 35*

*No heart disease, but predicted to have heart disease: 11*

*Heart Disease, but was predicted not to: 15*

*Accuracy:75%*

**

*Experiment 4:*

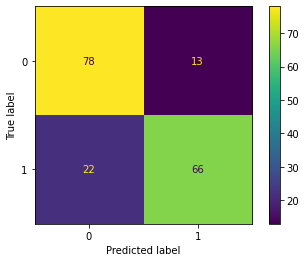
*No heart disease predicted correctly: 48*

*heart disease predicted correctly: 37*

*No heart disease, but predicted to have heart disease: 9*

*Heart Disease, but was predicted not to: 10*

*Accuracy:82%*

**

*Experiment 5:*

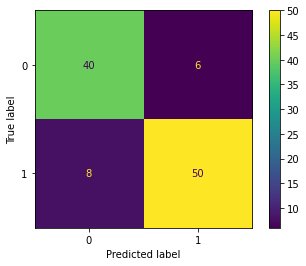
*No heart disease predicted correctly: 66*

*heart disease predicted correctly: 78*

*No heart disease, but predicted to have heart disease: 13*

*Heart Disease, but was predicted not to: 22*

*Accuracy:87%*

**

*Experiment 6:*

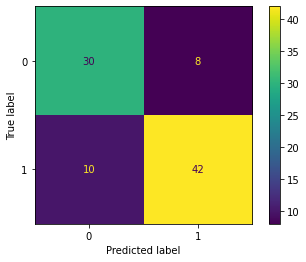
*No heart disease predicted correctly: 50*

*heart disease predicted correctly: 40*

*No heart disease, but predicted to have heart disease: 6*

*Heart Disease, but was predicted not to: 8*

*Accuracy: 83%*

**

*Experiment 7:*

*No heart disease predicted correctly: 42*

*heart disease predicted correctly: 30*

*No heart disease, but predicted to have heart disease: 6*

*Heart Disease, but was predicted not to: 8*

*Accuracy:80%*

* 1. *Discussion of Results*

**The best experiments were experiment 5, and experiment 1 were the best. Experiment 5 was 87% accurate, and experiment 1 was 85% accurate. Experiment 2 was the worst, due to lack of correlating variables. Experiment 5 gave 66** **no heart disease predicted correctly, and 78** **heart disease predicted correctly. The second-best experiment predicted 59 no heart disease predicted correctly, and 68 heart disease predicted correctly. The worst only predicted 73 correct for both categories combined. All the tests were pretty accurate however, almost all scoring above 80% accuracy.**

* 1. *Problems Encountered*

**Some of the data would have made the predicter less accurate or wouldn’t have worked. So, we had to remove a few columns. We also had issues with using the right models and choosing a good data set. Some of the data correlated heavily in some cases but didn’t matter at all in other cases. This led to outliers in some categories becoming hard to predict.**

* 1. *Limitations of Implementation*

**The model is good; however, it isn’t possible for any model to predict whether somebody will develop heart disease in a perfectly accurate and objective way. That is because this is real data, and the human body doesn’t run on numbers, obviously chance and other independent variables not shown impact the odds of developing heart disease. There could be better models, but the limitations derive from how many factors play a role, and how much of a role these factors play is different in almost every case.**

* 1. *Improvements/Future Work*

**Having more time would have led to a more accurate predicter, but since we used a continuous dataset with non-logistical regression, we had to start from scratch a day before it’s due. Also, we’d be interested in working on a continuous dataset, to be able to compare the accuracy of all models. Although it’s impossible to truly create a perfectly accurate predicter for every individual’s case, it is an interesting topic. In the future, we would have used a larger dataset, with more variables in order to create an even more accurate model.**

1. **CONCLUSION**

**The confusion matrix had an accuracy score of 0.87 for our best experiment. The dataset could use some work, but over 4/5 of the time the results were predicted correctly, however if something of this nature was to be implemented into a hospital it would need a much higher success rate in predicted heart disease. However, this could be useful as an “at home heart disease” type of model and will predict correctly 87% without having to pay a lot of money. Overall, this model works will for our goal, even though it may fall short in more professional medical settings.**

**REFERENCES**

<https://www.kaggle.com/datasets/cherngs/heart-disease-cleveland-uci>