**Predicting Patient Survival With a Decision**

**Tree Model**

Kerry Effinger, keffinger2@bellarmine.edu

Katie Zopff, kzopff@bellarmine.edu

**ABSTRACT**

Machine learning models can be used to predict binary outcomes for datasets in various fields of study. In this report, a decision tree model was used with a large dataset on Patient Survival after one year of treatment at a hospital in Greenland. Certain variables describe each patient, such as whether or not the patient was a smoker, whether the patient was from a rural or urban area, and which drug the patient was treated with. Additionally, the main characteristics we used in the model were the patient’s age, the patient’s body mass index, and the number of previous conditions displayed in the patient (ranging from 1 to 5). These variables were then used with a decision tree machine learning model to predict if the patient survived one year or not.

**I.** **INTRODUCTION**

For our project, we studied a data set about patient survival at a hospital in Greenland after a year of treatment. The data comes from patients over many years, though it is unclear how many years exactly. The set has multiple variables containing information about the patients’ health, and then the binary value of 1 if the patient survived after one year and 0 if they did not. To model this data, we decided to use a Decision Tree due to the data’s many categorical and binary variables. The main variables we worked with in the data were the patient’s age, body mass index, and the number of previous conditions the patient had or was treated for. These were some of the main numerical variables, with the rest of the data primarily binary variables. Using the SciKitLearn library in python, we prepared the data with decimal scaling, split the data into train, validate, and test groups, then ran the model. We ran four different experiments in an attempt to get the best performance from the model.

**II.** **BACKGROUND**

*A.* *Data Set Description*

The dataset we used was found on Kaggle. It is from a study of patients at a hospital in Greenland. The patients have one or more of unidentified health conditions (condition A, B, C, D, E, F and Z) and the data has a binary column stating whether or not the patient survived past one year at the hospital. It is unclear where the data is originally from, or if the data is real or not. There is not a reference on Kaggle citing the original source; it just states that the data is Public Domain. The URL to the website is: <https://www.kaggle.com/rsrishav/patient-survival-after-one-year-of-treatment>

*B.* *Machine Learning Model*

We used the decision tree machine learning model for our dataset. We originally worked with another dataset about forest fires and decided that K Nearest Neighbor would make sense for that data since we were predicting how several factors might contribute to if the fire was large or small. However, that dataset had some issues (which will be discussed later) and we changed to the dataset on Patient Survival. Since this dataset had many categorical variables relating to a patient’s health and conditions, we decided that the Decision tree model would work well.

A Decision tree model works by splitting the data into different sets based on some variable, and then splitting it again and again into more subsets until it comes to a decision. The different categorical and binary variables act as the nodes on the decision tree and whether or not the patient survived after one year of treatment is the output predicted by the model.

**III.** **EXPLORATORY ANALYSIS**

The Patient Survival data set explored in this analysis contains 23097 samples with 18 columns of various data types. It was found that 13 samples had ages over 100, no data for what drugs they were treated with, unusual numbers for the patient BMI, and were the only patients to have condition Z. So, we removed those 13 samples and the column for condition Z. This left us with 23084 samples and 17 columns of variables. A complete listing of the variables used and their data types is shown in **Table 1** After some initial analysis, we found that some samples had NaN values for the different conditions and the number of previous conditions. It was noted that the most common condition had by the patients was condition A (as shown in **Table 2**), so for each sample with NaN values we filled them with a 1.0 for condition A and 0.0 for the rest, with 1.0 for the number of previous conditions. **Table 3**displays the statistical summary for the variables Patient\_Age, Patient\_Body\_Mass\_Index, and Num\_of\_prev\_cond. This is interesting data as it shows the range of ages and BMIs in the data set, as well as the distribution of the number of conditions, which would logically be significant variables in modeling for rates of survival. The frequency of the categorical and binary data points is shown in **Figure 1.**

**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| ID\_Patient\_Care\_Situation | Ordinal /int64 |
| Diagnosed\_Condition | Nominal/int64 |
| Patient\_ID | Ordinal/int64 |
| Treated\_with\_drugs | Nominal/object |
| Patient\_Age | Ordinal/int64 |
| Patient\_Body\_Mass\_Index | Ordinal/float64 |
| Patient\_Smoker | Nominal/object |
| Patient\_Rural\_Urban | Nominal/object |
| Patient\_mental\_condition | Nominal/object |
| A | Nominal/float64 |
| B | Nominal/float64 |
| C | Nominal/float64 |
| D | Nominal/float64 |
| E | Nominal/float64 |
| F | Nominal/float64 |
| Number\_of\_prev\_cond | Nominal/float64 |
| Survived\_1\_year | Nominal/int64 |

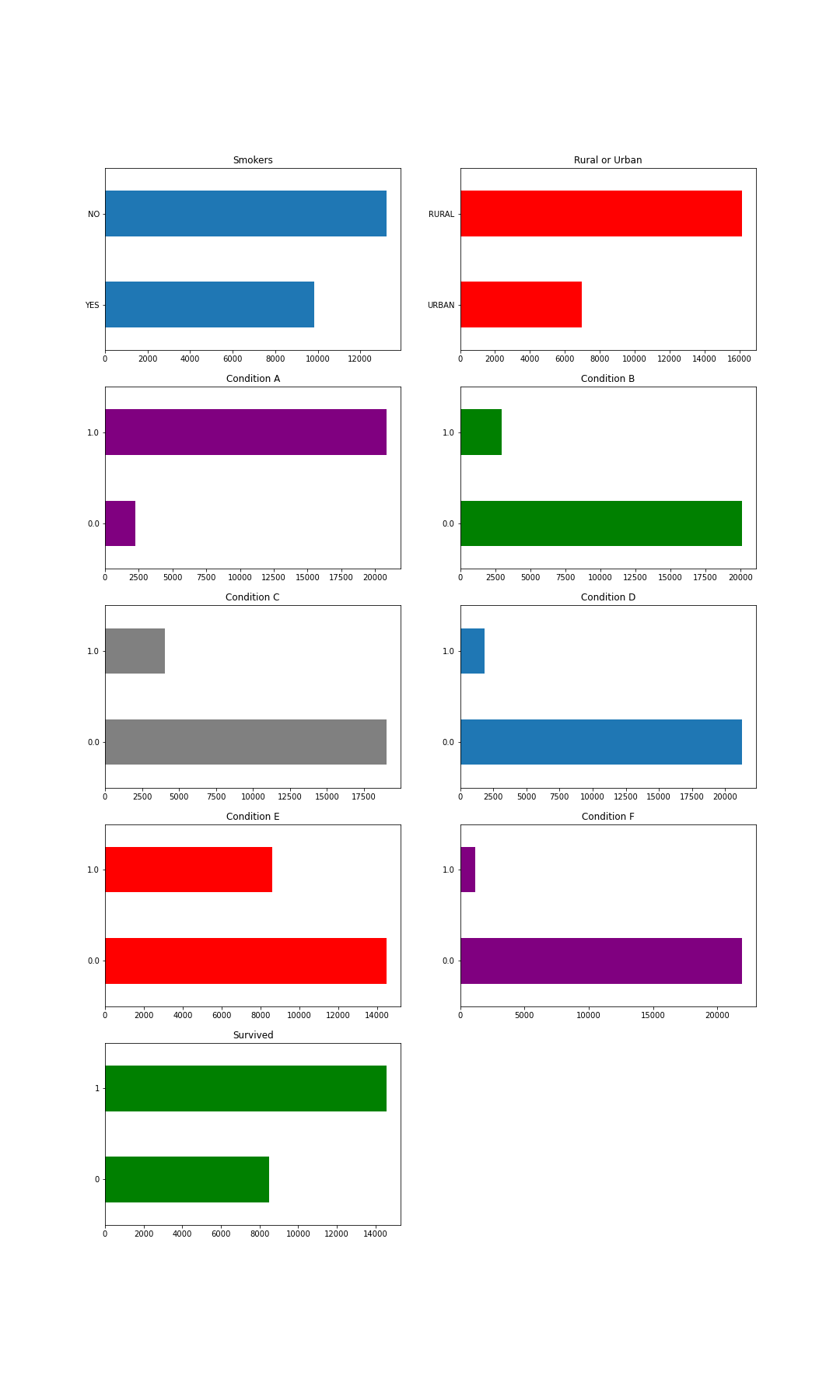
**Table 2: Proportions for A**

|  |  |  |
| --- | --- | --- |
| *Category* | *Frequency* | *Proportion* |
| Has Condition A (1.0) | 20865 | 90.39% |
| Does not have Condition A (0.0) | 2219 | 9.61% |

**Table 3: Summary Statistics for Patient\_Age, Patient\_Body\_Mass\_Index, and Num\_of\_prev\_cond**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *Count* | *Mean* | *Std.* | *Min* | *25%* | *50%* | *75%* | *Max* |
| *Patient Age* | 23084 | 33.16 | 19.43 | 0 | 16 | 33 | 50 | 66 |
| *Patient BMI* | 23084 | 23.47 | 3.77 | 17 | 20.21 | 23.39 | 26.79 | 29.9996 |
| *Num. Of Prev. Conditions* | 23084 | 1.71 | 0.77 | 1 | 1 | 2 | 2 | 5 |

**Figure 1: Collection of horizontal bar charts for various categorical data and binary data in the Patient Survival data set.**



**IV.** **METHODS**

*A.* *Data Preparation*

As mentioned above, we noticed that some rows had NaN values for the various conditions and the number of previous conditions. Noticing that condition A was the most common condition for the patients to have by a significant amount, we filled all the rows with NaN values with 1.0 for condition A, 0.0 for the rest of the conditions, and 1.0 for the number of previous conditions. After this, we also noticed that 13 patients had some abnormalities. These 13 patients had ages over 100 (the highest being in the 140s), BMIs between 1 and 2 (much too low to be possible), no information for the drugs they were treated with, and they were the only patients with a value of 1.0 in the condition Z column. For this reason, we decided to eliminate these samples from our data and delete the Z column since after deleting the rows, every value in that column was 0.0.

For running our model, we normalized the data for two experiments using decimal scaling. In particular, we scaled the values for patient age and BMI by dividing them both by 100 to ensure the values are within a smaller range of each other.

*B.* *Experimental Design*

We decided to test the decision tree model with four different experiments. The parameters for each experiment are as shown in **Table 4**.

**Table 4: Experiment Parameters**

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | All four (4) raw features with 60/20/20 split for train, validate, and test |
| 2 | All four (4) decimal normalized features with 60/20/20 split for train, validate, and test |
| 3 | All four (4) raw features with 80/10/10 split for train, validate, and test |
| 4 | All four (4) decimal normalized features with 80/10/10 split for train, validate, and test |

*C.* *Tools Used*

The following tools were used for this analysis: Python v3.8.6 running the Anaconda Navigator v4.9.2 for Apple Macintosh computer and Windows 10 was 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, and SKLearn 0.18.1. These tools helped us explore the data, analyze it, and run the decision tree model to predict outcomes of patient survival.

**II.** **RESULTS**

*A.* *Classification Measures*

After running each experiment, we found the classification measures for each experiment, which are shown in **Table 5,** to evaluate the effectiveness of the model in each case.

**Table 5: Classification Measures for each experiment**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *Accuracy* | *Precision* | *Sensitivity* | *F Measure* |
| *1: 60/20/20* | 0.61598 | 0.70982 | 0.65770 | 0.68277 |
| *2: Normalized 60/20/20* | 0.61642 | 0.71129 | 0.65564 | 0.68233 |
| *3: 80/10/10* | 0.64428 | 0.73734 | 0.68991 | 0.71284 |
| *4: Normalized 80/10/10* | 0.64731 | 0.73970 | 0.69262 | 0.71538 |

*B.* *Discussion of Results*

All of our results were very close to one another. The accuracy for the two experiments performed using an 80/10/10 split was slightly higher than that of the experiments with a 60/20/20 split. This makes sense, as the training sets in the 80/10/10 split experiments are larger, so the model has more data to learn from. The classification measures are also very slightly higher for the experiments using the normalized data compared to their un-normalized counterparts. This could be because the normalization method of decimal scaling makes the continuous variables to be within a smaller range and thus have less deviation.

*C.* *Problems Encountered*

We came across several issues with our first dataset on forest fires. First of all, the source was relatively small, with only about 500 rows of data. Second, the output that we were trying to predict with a model was not already a binary output, so we had to convert a continuous variable to a binary one, using the mean of the column. In converting the column to binary, we realized several additional issues in the data. Mainly, the area of the fire had over 200 instances where the area was 0.0. We did not understand how that made sense, but we decided to try to make it work by doing a log transform to help correct the skewness that resulted from all the 0s. Finally, the log transform and multiple normalizing techniques did not help the extreme skewness of the data and we ultimately decided to find another dataset. This set us back a few days, but we still had plenty of time to put together a thorough report on the new data while still using some of the code we came up with the first time.

With the new dataset on patient survival rates, we had some unique challenges as well. As mentioned in the exploratory analysis and the data preparation parts earlier, there was some missing and unusual data that had to be accounted for. We did not have nearly as many issues with our new dataset as with the previous one, but some of the data was strange, so we went ahead and removed them from the data. We had plenty of data so we could afford to lose a few rows. Another issue with the dataset is that it is not cited in detail as to where it came from. We found it on Kaggle and there is a very brief and vague description, but no credit to where it is from or how it was collected. This raises some questions about the context of the dataset and also its reliability.

*D.* *Limitations of Implementation*

The classification measures for our experiments shows that our model was not very accurate. The Decision Tree model may be too simple for our data. We may have had better outcomes using a Random Forest model, creating many decision trees rather than just one to predict whether or not a patient would survive. A Naive Bayes model might also have worked for this data set to predict survival based on probabilities from their given medical conditions/history. Additionally, different variables within our dataset may have been a better fit for the decision tree model. We realized that the binary variables in our data may have been an interesting experiment and possibly led to better results from the model. Perhaps we had the right model, but were just using the wrong variables within the dataset.

*E.* *Improvements/Future Work*

In the future, we could try more experiments with different normalization methods for example. However, using a different model altogether might be the best approach, since we got relatively low accuracy for each of the different methods of splitting and normalization that we attempted. We also could have tried using the same dataset, but different variables within it. We only used a select few of the many options, and as mentioned earlier, the binary variables may have been a better fit. Finally, a different dataset could have improved our results, but as we have found both in this project and in other projects, the “perfect” dataset is very difficult to find. All things considered, the Patient Survival data had few issues and did not require much cleaning or additional work. Perhaps the model was just not the most ideal for predicting survival, or we needed to spend more time making more precise and accurate models.

**III.** **CONCLUSION**

We used the Patient Survival data set to predict if a patient would survive after one year of treatment using a Decision Tree model. The model created was not very good and did not have very strong classification measures. We found that using a larger training set and normalizing the data helped a little, but not enough to make it a great model. This error in our model could be because the Decision Tree model was not as good of a fit for the data as some other model. The model may be improved by further work to explore the model’s accuracy based on individual variables. Ultimately, however, it would be worthwhile to look at different models for this dataset since the measures were consistently lower than is ideal.

**REFERENCES**

[*https://www.kaggle.com/rsrishav/patient-survival-after-one-year-of-treatment*](https://www.kaggle.com/rsrishav/patient-survival-after-one-year-of-treatment)

[*https://scikit-learn.org/stable/modules/tree.html*](https://scikit-learn.org/stable/modules/tree.html)