**COVID-19 Recovery**

**Complete Blood Count Database**

Hannah Garrett, [hgarrett@bellarmine.edu](mailto:hgarrett@bellarmine.edu)

**ABSTRACT**

This project focuses on predicting the recovery of COVID-19 patients using logistic regression models. The dataset includes various features such as the patient’s gender, blood cell counts, age, treatment provided, and multiple other column variables. The target variable for the regression model is the outcome. The objective of this model is to predict the recovery of COVID-19 patients based on each of their different variables, separating them into two categories: recovered and not recovered.

1. **INTRODUCTION**

This data set contains 13 different variables – admission date, discharge date, outcome, patient age, gender, treatment, ventilated (Y/N), and 7 other variables pertaining to the patient’s blood count. These variables contain no null values and have 103 entries. I chose this dataset because it contained a clear variable to predict (outcome) and it was large enough to build a reliable prediction model on. I used binary classification because this data set gave two classes to separate the data into (recovered or not recovered). This project is built with python.

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

The “COVID-19 Complete Blood Count (CBC)” database was collected by Tawsifur Rahman and 2 other collaborators who wanted to allow researchers to use this database to produce useful and impactful scholarly work on COVID-19, which can help in tackling the pandemic issue. The objective of this study was to analyze health factors and predict patient outcomes based on these factors. The dataset is small, yet its complexity arises due to the fact it has strong multicollinearity. A complete listing of all the variable is shown in Table 1.

* 1. *Machine Learning Model*

Logistic regression is a method used to analyze and model the relationship between a binary dependent variable and either one or more independent variables. It attempts to estimate the probability of the target variable’s value using the values of the independent variables. Logistic Regression is mostly used for classification but can also be used for regression. It is used in fields such as psychology, data science, economics, finance, and many more.

1. **EXPLORATORY ANALYSIS**

This dataset contains 309 samples with 14 columns of various data types.

**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| V1 Admission\_DATE | Object |
| V2 Discharge\_DATE | object |
| V3 Outcome | Object |
| V4 Patient Age | Int 64 |
| V5 Gender | Object |
| V6 Sample Collection Date | object |
| V7 Treatment Provided | Object |
| V8 Ventilated (Y/N) | Object |
| V9 Monocytes (%) | Float64 |
| V10 RBC distribution width | Float64 |
| V11 WBC count | Float64 |
| V12 Platelet count | Float64 |
| V13 Lymphocyte count | Float64 |
| V14 Neutrophils count | Float64 |

1. **METHODS**
   1. *Data Preparation*

To begin the preprocessing, I first started with renaming columns to make them easier to work with and then dropping unnecessary columns. Since I was analyzing patient outcomes based on health factors, I dropped 4 columns pertaining to the patients’ dates admitted/discharged, their collection date, and the treatment provided, because every treatment contained the same value. These factors are important to the dataset, but unnecessary to logistic regression. There were no null values, so I did not have to drop null values. I then had to encode categorical variables like M/F and ventilated (Y/N) to be 0 and 1 to make the data easier to work with. After this, I could then split it up into dependent (outcome variable) and independent variables (all other variables).

* 1. *Experimental Design*

All experiments ran with logistic regression

Table X: Experiment Parameters

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | 80/20 train test split with 50/50 validation split  Results:  Logistic Regression accuracy: 93.54% |
| 2 | 80/20 train test split with 60/40 validation split  Results:  Logistic Regression accuracy: 94.59% |
| 3 | 70/30 train test split with 50/50 validation split  Results:  Logistic regression accuracy: 97.82% |
| 4 | 70/30 train test split with 60/40 validation split  Results:  Logistic regression accuracy: 98.18% |

* 1. *Tools Used*

The following tools were used for this analysis: Python v3.9.12 running the Anaconda 4.3.22 environment for Apple Macintosh computer was used for all analysis and implementation. In addition to base Python, the following libraries were also used: Pandas 1.4.2, Numpy 1.21.5, SKLearn 0.18.1. I chose these python packages because this is what we use in class.

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

Table

Description automatically generated Square

Description automatically generated

* 1. *Discussion of Results*

Overall, the model did very well in detecting whether a patient has recovered or not from COVID based on their health variables. From the confusion matrix, it looks like the model only predicted one value incorrect, which was a false negative. In this case scenario, that is better than getting a false positive because the incorrect prediction showed a patient did not recover when they actually did. The f1 score of the not recovered side was 98% with a precision of 95%. The f1 score of the recovered side is 99% and precision of 100%. This was a very good model because the accuracy was close to perfect although the sample sizes were very small.

* 1. *Problems Encountered*

The first problem I encountered was finding a suitable dataset to perform the logistic regression on. It was hard to find a dataset on Kaggle that was suited for a logistics regression analysis and classification. I used a different dataset but later found out it did not work for prediction and confusion matrix. Another problem was finding good numbers for the train/test split in order to find a good score with the model.

* 1. *Limitations of Implementation*

The model performed very well, but the sample sizes ended up being so small, so it is difficult to see the true accuracy of the model. The accuracy score was close to perfect, but that was only reflecting a small sample size. This is probably because the dataset only contained around 300 samples.

* 1. *Improvements/Future Work*

In the future, I would probably use a different dataset that contained a larger sample size so that I could see the accuracy of the prediction on a wider scale. I think there were enough variables, but the data could have definitely used more data points to help the model.

1. **CONCLUSION**

For this project, I utilized a logistic regression model to attempt to classify whether a patient has recovered or not from COVID-19 based on 10 different variables. Overall, the model achieved an accuracy score of 99%. The model was very accurate as a whole and only had one wrong prediction based on the confusion matrix.

**REFERENCES**

[*https://www.kaggle.com/datasets/tawsifurrahman/covid19-complete-blood-count-clinical-database*](https://www.kaggle.com/datasets/tawsifurrahman/covid19-complete-blood-count-clinical-database)