De La Salle University - Manila
Term 2, Academic Year 2023 - 2024

In Partial Fulfillment
Of The Course Requirements
In **CSINTSY - S15**

MCO3: Machine Learning

Submitted By:

Chong, Hans Kirzen
Salen, Rommel Kendric
Tuco, Kevin Bryan
Uy, Wesley King

Submitted To:
Mr. Thomas James Tiam-Lee

Submitted On:
April 17, 2023

I. Introduction

Covid-19 has been a major pandemic happening starting from Wuhan, China (Page, Hinshaw, & Mckay, 2021). It has spread around the world. The disease currently infected 600 million people and almost caused 7 million deaths (Worldometer, 2023). The severity of Covid 19 ranges between patients. Some may not develop symptoms, some may have mild cases, and some more severe or critical and need medical attention. According to the CDC, 81% received mild cases of Covid-19 while 14% and 5% received a more severe and critical version of it, respectively.

During the peak of the virus, hospital staff are in a short supply of medical equipment. Patients who are at high risk could not get their medical attention and their condition often got worse. The program is designed to test the patient and determine if they are at a higher risk of death because of Covid-19. It will help hospitals and clinics who are in low supply to accurately ration and effectively allocate their resources to help people who are at a higher risk of death and in need of hospitalization.

II. The Dataset

The dataset was published by the Mexican government during 2020. The dataset consist of an abundant number of entries of patients with related information to their conditions. It consists of 21 features and 1,048,575 entries in the dataset, 971,633 alive cases and 76,942 dead cases.
https://www.kaggle.com/datasets/meirnizri/covid19-dataset

The features of the dataset are sex, age, pneumonia, diabetes, copd (Chronic obstructive pulmonary disease), medical unit, asthma, inmsupr (immunosuppression), hypertension, cardiovascular, renal chronic, obesity, tobacco, usmer(level of patient treated medical units), other disease, dead date and classification final. The values for patient_type will be 1 and 2, 1 is for return home, and 2 will be hospitalization.

III. Methodology

Logistic Regression

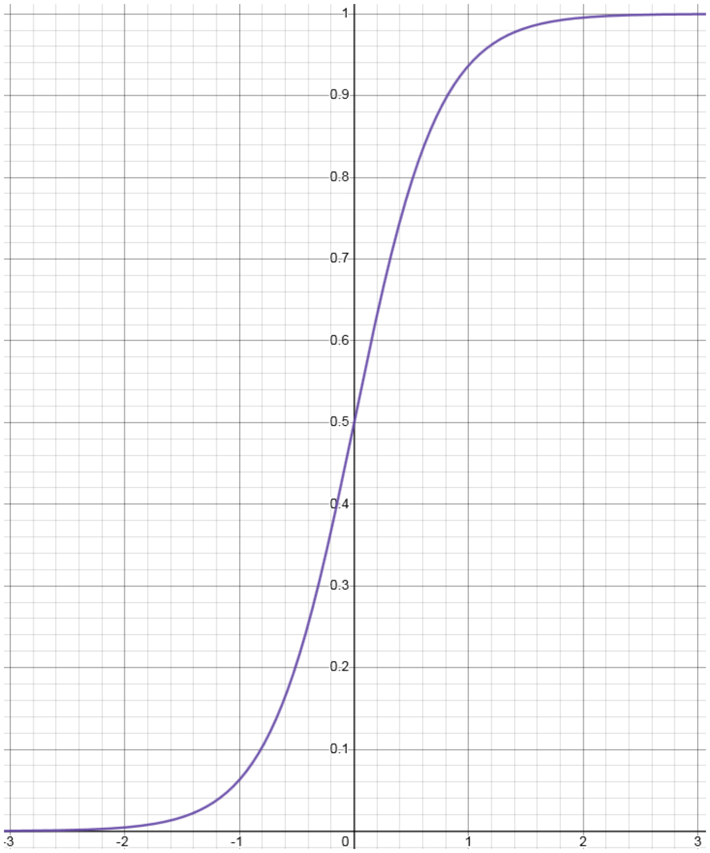
The first of the implemented algorithms is Logistic Regression. Given that the dataset already had its values properly labeled, a supervised learning algorithm was best, and it is one that’s well-known for solving classification problems. It’s used to determine the probability of a binary outcome, but instead of outputting a 0 or 1, it calculates the probability of the event, and is transformed into either 0 or 1 afterwards.

It uses a logistic function or a sigmoid function; in a graph, it is an S-shaped curve that represents the probabilities; in an equation, it can be represented as:

$$s(x) = 1/(1 + e^{-value})$$

Where *‘e’* is Euler’s number and *‘value’* is the input. Note that *value* is preceded by a negative operator.

(Sigmoid/Logistic Function - Desmos Graphing Calculator)



In the Sigmoid Function above, x is the input (the dataset) and y is the output (the probability). A large positive x value means a higher likelihood of outputting a 1 or true, and a large negative x value means a higher likelihood of outputting a 0 or false.

Logistic Regression takes the input from the dataset, combining them linearly using the coefficient value of its respective column - the coefficient is a constant real value, and is determined by the algorithm based on the values of the dataset - and represents it via an equation, which then outputs the probability of what we are asking for. An example of what a Logistic Regression represented as an equation would be:

$$y = e^{(b0 + b1*x)} / (1 + e^{(b0 + b1*x)})$$

Where y is the probability, e is Euler’s number, $b0$ is the bias, $b1$ is the respective coefficient (determined by the program itself based on the dataset) of each column, and x is the input themselves. In this program, the equation was already provided for in Python’s library.

That probability is then transformed into a binary output through a condition check (*e.g.* p(x) >= 0.5; return 0, else if p(x) < 0.5; return 1) and subsequently outputs the binary outcome.

The main reasons why we picked the Logistic Regression is because it’s a solid choice for supervised classification problems due to its efficiency and inexpensive computational cost. It’s a simple algorithm that learns quickly, and does well in accurate predictions.

Decision Tree Algorithm

The second algorithm of the project applied is Decision Tree algorithm. A Decision Tree Algorithm (DTA) is a type of supervised machine learning algorithm that makes use of a tree structure to predict a result from a series of features. The basic idea is to split the input data based on a set of rules or conditions to partition the data into homogeneous subsets with respect to the target variable (variable we are trying to predict/ classify). The tree has multiple types of nodes being:

Root nodes - the beginning of a decision tree and the start from which the population starts dividing according to different features

Decision nodes - nodes we get after splitting the root nodes

Leaf nodes - nodes where you can’t split anymore; it also is the output nodes

The choice of which feature is the root/ decision nodes or when to stop splitting is dependent on factors such as entropy and information gain

Entropy is an estimation on the uncertainty in the dataset or feature to compare data. If the two choices have equal or almost equal values (i.e. yes or no have values of 5, uncertain which to pick), the choice/ node will have high entropy. High entropy means high randomness; which is a bad thing because we want a pure sub-split (either its a yes or a no only). To fix this, we have to split the node once again using a feature. Which feature will lessen the impurity of the node the most? That's when we use information gain. The formula for entropy is the following:

$$E(S) = -p_{(+)}\log(p_{(+)}) - p_{(-)}\log(p_{(-)})$$

Where S is the subset of the training example, $p_{(+)}$ is the probability of the positive class and $p_{(-)}$ is the probability of the negative class

Information gain measures the reduction of uncertainty/ entropy given some feature; it's what helps us to pick which feature or attribute is selected as a root or decision node. Information gain is calculated using this formula:

$$\textit{Information Gain} = E(Y) - E(Y|X)$$

This is just entropy of the full dataset - entropy of the dataset given some feature

Once we calculate this for all features, the root node, or the first split, will be attributed to the feature with the highest information gain. After that, we recursively go to each node, calculate the information gain for the set of features, then pick the one with the highest information gain. We keep doing this until a stopping criterion is met; be it a maximum depth, minimum number of samples per leaf node, max leaf nodes, etc.

There is a chance that the decision tree is overfit; which means that the tree starts to become too fit on the training data. This makes the tree be really good at classifying training data, but really bad at classifying unseen data. A way to fix this is by pruning, or by deleting nodes that aren't important or significant; we can do this while creating the tree or after the tree has been created.

The main reasons why we picked the Decision Tree Algorithm is because they are good for classification tasks as they can predict both continuous and discrete values. They are simple and easy to understand and interpret. They can also handle any type of data whether the data is numeric, categorical or boolean. They are useful in data exploration as they are one of the fastest in relating the features and producing a result variable based on the features.

Comparison of the two algorithms

The two algorithms both sufficiently fulfill the purpose of taking data and outputting a binary output, however both have their own respective advantages and disadvantages, and their performance against each other wildly varies depending on the dataset, and there is no completely superior algorithm to the other.

Logistic Regression provides more numerical output, providing confidence intervals and generally having better predictive accuracy, but could potentially be difficult to interpret for users. Decision Trees are much more clear-cut and easier to interpret and understand to users who are not as versed in reading data.

Logistic Regression also requires the user to select the interaction terms, while Decision Trees decide it automatically. Decision Trees also perform better in larger data sizes, given its nature to continuously separate based on data (but do have the potential of overfitting) while Logistic Regression looks at the predictor's effects simultaneously, making it better for smaller sample sizes.

Generally speaking, Logistic Regression excels better in accurate outcomes, but Decision Tree performs better when making predictions or describing data due to their interpretability, but again, which algorithm performs better tends to be on a case-by-case basis.

Data Preprocessing

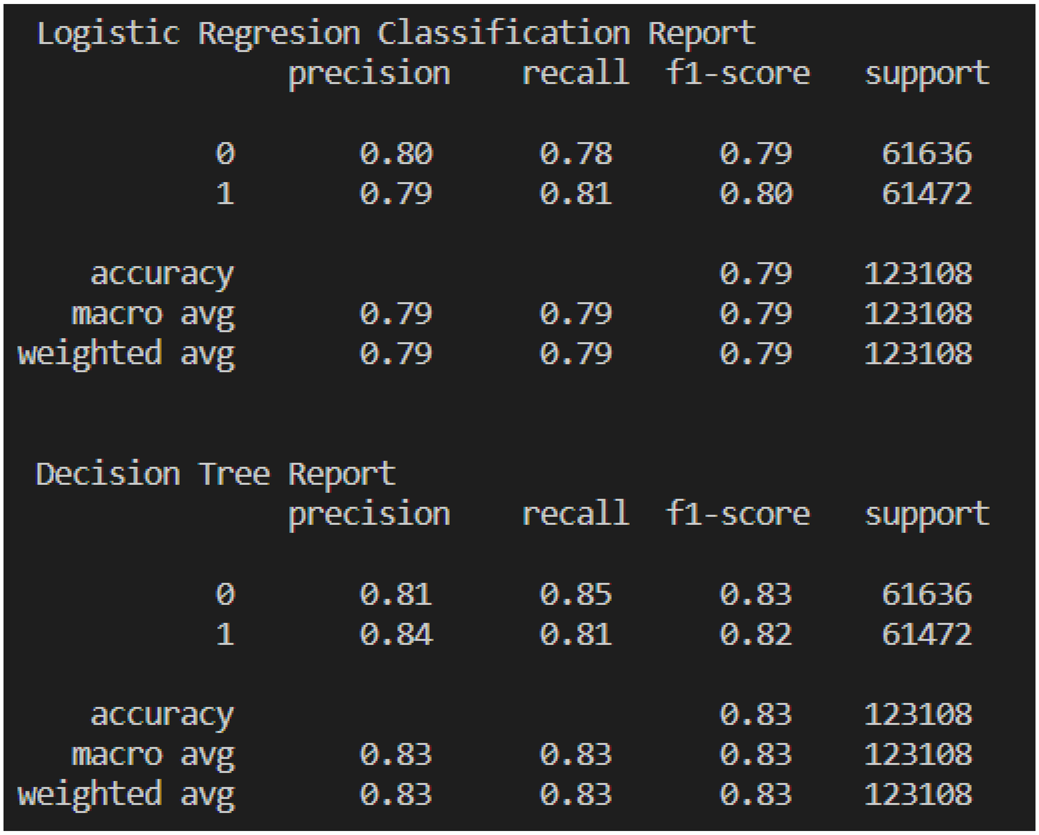
The processing data start off with remodifying data with 'CLASSIFICAL_FINAL' being from numerical values converted to binary, if the value is 3 or less it will be converted to 1, because the patient has a form of Covid19, and if the value is 4+ it will be converted into 0. Next is converting sex, the data 2 to 0 and 1 as 1. The patient_type is also converted to 2 to 1 and 1 to 0; which means, when looking at the dataset source in kaggle, hospitalized is 1 and 0 is returned home. Finally the rest of the data converted 2 to 0 indicated that they are false. The program creates a new column and sets it as 1 if it is a valid date and 0 if it is not then, drops the date_died.

Due to the overwhelming amount of missing data in the ICU, Intube, and pregnant i.e. values are 97 or 99 and the unclear values for the medical unit. They were dropped as a feature from the dataset. The dataset suffers from large class unbalance in terms of the distribution of values on multiple columns like the Dead column. To combat it, before the testing, the program undersamples to prevent bias and prevent the model from skewing toward the majority. This is to ensure the program would try to even out the program and data used in random sampling from the original dataset.

IV. Results and Analysis

The classification problem that the two algorithms are trying to solve is whether a patient is at risk of dying to Covid-19 or not. For this, the target column is the “DEAD” column, to show if the patient died or not given the values from the rest of the columns. In our implementation, the algorithm returns either a 1 or 0; 1 for means that they are in risk of dying and 0 means that they are not in risk of dying.

The program evaluates the performance by running a test set and comparing the prediction and the actual results of the dataset.



```
 Logistic Regresion Classification Report
              precision    recall  f1-score   support

           0       0.80      0.78      0.79     61636
           1       0.79      0.81      0.80     61472

    accuracy                           0.79    123108
   macro avg       0.79      0.79      0.79    123108
weighted avg       0.79      0.79      0.79    123108


 Decision Tree Report
              precision    recall  f1-score   support

           0       0.81      0.85      0.83     61636
           1       0.84      0.81      0.82     61472

    accuracy                           0.83    123108
   macro avg       0.83      0.83      0.83    123108
weighted avg       0.83      0.83      0.83    123108
```

Figure 1. Data Produced the Classification Report

Based on Figure 1, there is almost 80% accuracy on the logistic regression. The learning model correctly identifies 80% of the time when fed an input. The accuracy of the Decision Tree Report is at least 83% and correctly identifies more patients compared to the Logistic Regression. In terms of precision, recall, and f1 test. The Decision Tree tends to score higher than the Logistic Regression, this is part due the similarity the feature that it can connect with one another to produce a more accurate prediction.

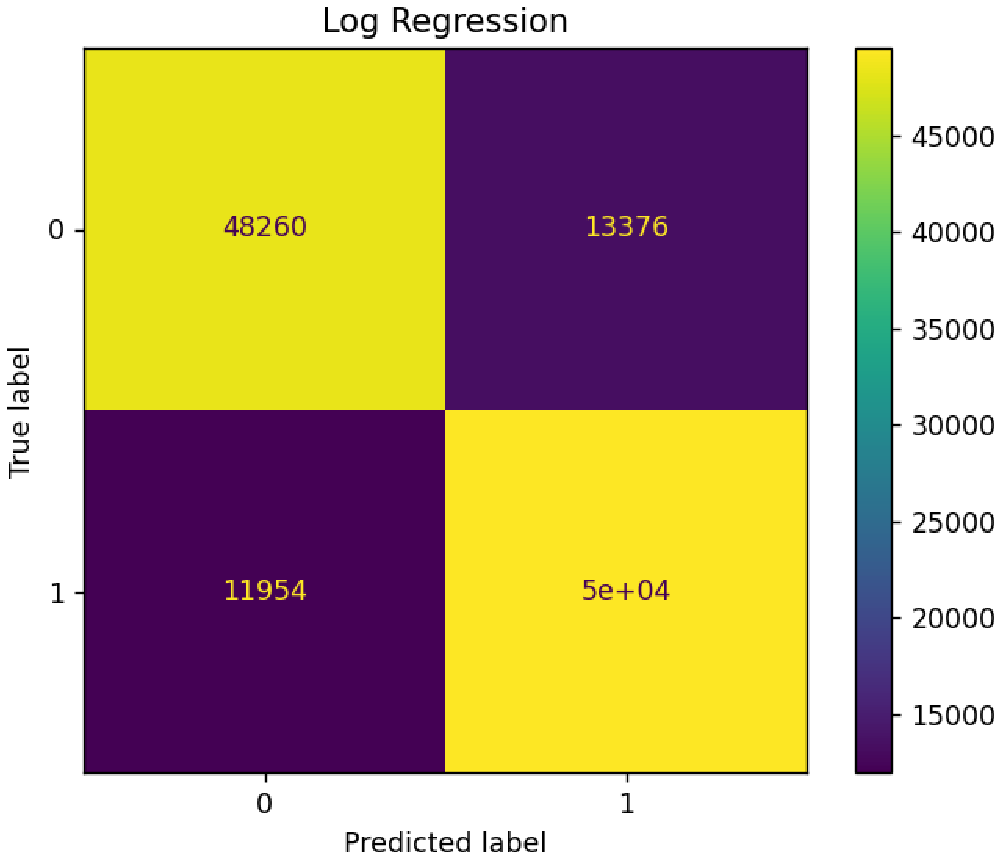


Figure 2. Data Produced the Logistic Regression

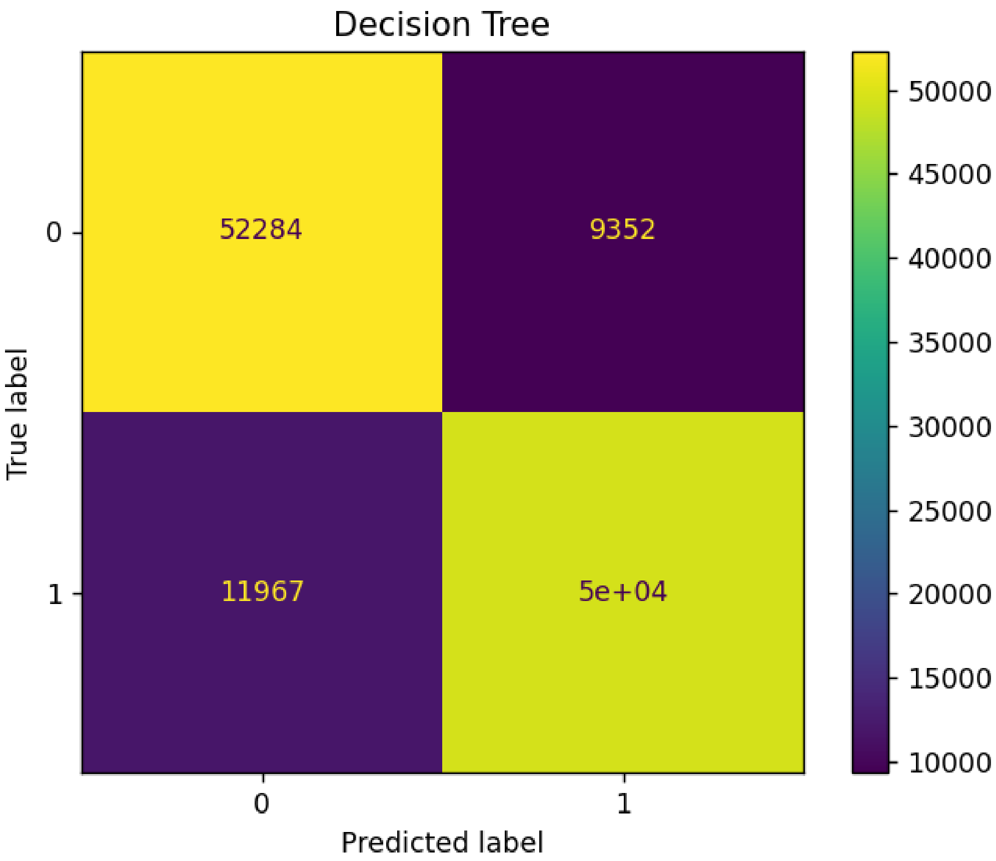


Figure 3. Data Produced the Decision Tree

The confusion matrix shows the exact results of the prediction; it also shows how many the algorithm predicted correctly and incorrectly from the test dataset. Both algorithms have a high number of false negatives, which suggests that they are not performing well in identifying true positives; with the decision tree having more false positives than the logistic regression. This could suggest that the tree may be overfitting the training data. While Decision Trees and Logistic Regression share a similarly high number of false negatives, the latter has a more significant amount of false positives, meaning that the parameters for Logistic Regression may need to be adjusted, whether it be the equation representation, or the conditions of outputting a 0 or a 1.

V. Conclusions and Recommendations

The goal of the project is using a set of data to analyze the patient whether they are at risk of dying from Covid-19. With the program we can correctly identify who are in dire need of medical attention and allow countries who are suffering from the lack of equipment to quickly analyze patients and their current status. Even to this day, more variants of Covid-19 still can appear in countries.

We found that the decision tree might be overfitting the training data, we suggest pruning the tree to delete unnecessary or insignificant nodes; this can drop the fit score but it will help in making the algorithm more accurate on data not in the training set.

We could also lessen the amount of features, only including those the most important of the set; this can help in reducing noise and increasing the performance of both models. We also recommend trying out different hyperparameters to find out which hyperparameters work best for this specific dataset.

We could also test multiple other classification algorithms to know how well they work in this dataset compared to others; this increases our scope and helps us gleam more into which classification algorithm is best suited for the data set.

VI. References

Brownlee, J. (2020b). Logistic Regression for Machine Learning. MachineLearningMastery.com. https://machinelearningmastery.com/logistic-regression-for-machine-learning/

Bock, T. (2021, June 9). Decision Trees Are Usually Better Than Logistic Regression - Displayr. Displayr. https://www.displayr.com/decision-trees-are-usually-better-than-logistic-regression/

Cheesinglee. (2016, September 28). Logistic Regression versus Decision Trees. The Official Blog of BigML.com. https://blog.bigml.com/2016/09/28/logistic-regression-versus-decision-trees/

Coronavirus Disease 2019 (COVID-19). (2020, February 12). Centers for Disease Control and Prevention.https://web.archive.org/web/20200302201644/https://www.cdc.gov/coronavirus/2019-ncov/hcp/clinical-guidance-management-patients.html

COVID-19 Coronavirus Pandemic. (2023, April 17). Worldometers https://www.worldometers.info/coronavirus/

Decision Tree Advantages and Disadvantages. (n.d.). EDUCBA. https://www.educba.com/decision-tree-advantages-and-disadvantages/

Logistic Regression For Machine Learning and Classification - Kambria. (2019, July 9). Kambria. https://blog.kambria.io/logistic-regression-for-machine-learning/

Page, J. & Hinshaw, D. & McKay, B. (2021, Feb 26). In Hunt for Covid-19 Origin, Patient Zero Points to Second Wuhan. Market https://www.wsj.com/articles/in-hunt-for-covid-19-origin-patient-zero-points-to-second-wuhan-market-11614335404

Saini, A. (2023). Decision Tree Algorithm – A Complete Guide. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

What Is Undersampling? (2022, April). CORP-MIDS1 (MDS). https://www.mastersindatascience.org/learning/statistics-data-science/undersampling/

VII. Contributions of Each Members

Chong, Hans Kirzen

- Code creation
- Helped looked for machine learning algorithms
- Writing Dataset
- Writing Introduction

Salen, Rommel Kendric

- Research
- Data analysis
- Writing Logistic Regression Explanation
- Writing comparison between algorithms

Tuco, Kevin Bryan

- Research
- Code creation
- Data analysis
- Writing decision tree explanation
- Writing recommendations

Uy, Wesley King

- Code creation
- Data preprocessing
- Writing Results and analysis
- Writing Data Preprocessing