**PREDICTING THE LENGTH OF HOSPITAL TREATMENT FOR COVID-19 PATIENTS**

Oscar A Alvarez

University of Texas Rio Grande Valley

[Oscar.a.alvarez02@utrgv.edu](mailto:Oscar.a.alvarez02@utrgv.edu)

**1 Introduction**

The COVID-19 pandemic has made a tremendous change in people lifestyles. Due to its highly contagious and, in some cases, lethal characteristics, hospitals and healthcare centers have experience one of the highest demands in history. The number of patients has grown so much to the point where the resources become insufficient, and the needed care that each patient requires it is sometimes not provided. As per today, it is strongly believed that the time of treatment provided to an individual is related to the age and special conditions of the patient. However, due to several exceptions to these rules, assuming that the severity of the disease is just related to these two conditions is highly risky and innacurate. Therefore, a study that includes more variables that could affect the outcome has been done. However, a proper model has not been defined to predict how much time of treatment would a patient need. The purpose of this project was to define a proper software tool that can predict the length of stay for each patient that enters a healthcare center based on information provided in Kaggle.com. This tool should assist healthcare centers to optimize the use of available resources and provide better services to patients.

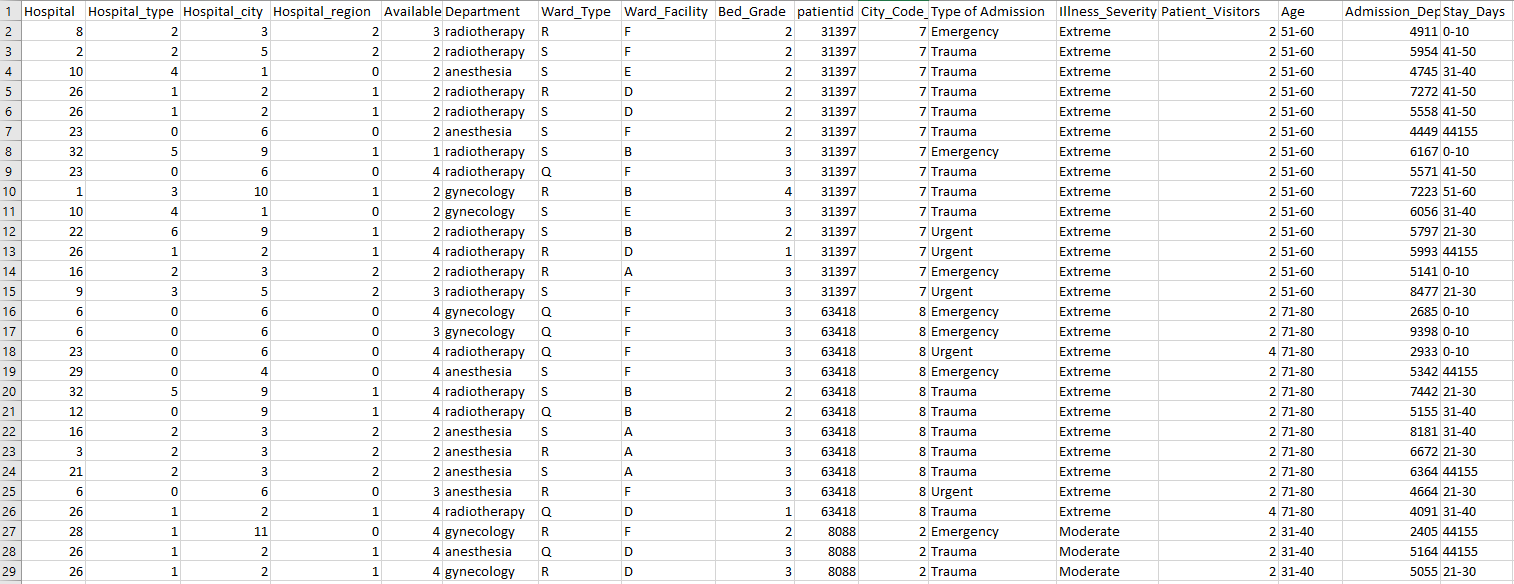
**2 Dataset**

The dataset was obtained from Kaggle.com. It consists of 17 features and 11 different classes. The total number of records is 318,439 from people who got recovered from COVID-19. The features represent data from the patient with the hope that a pattern could be detected. The features are:

* Case Id. Identification number assigned to case
* Hospital. Identification number assigned to each hospital in the study.
* Hospital Type. Number assigned to identify the type of the hospital
* Hospital city.
* Hospital region.
* Available extra rooms in hospital.
* Department. Department overlooking the case (radiotherapy, anesthesia, gynecology, TB & chest disease, surgery)
* Ward Type.
* Ward Facility.
* Bed grade. Condition of the bed.
* Patient ID
* City Code Patient
* Type of admission. Trauma, emergency, urgent.
* Illness severity. Extreme, moderate, minor.
* Patient Visitors.
* Age.
* Admission deposit.

As per the classes, the eleventh different categories represent the length of stay of the patients involved in the study. The categories are:

* Zero to ten days.
* Eleven to twenty days.
* Twenty-one to thirty days.
* Thirty-one to forty days.
* Forty-one to fifty days.
* Fifty-one to sixty days.
* Sixty-one to seventy days.
* Seventy-one to eighty days.
* Eighty-one to ninety days.
* Ninety-one to one hundred days.
* More than one hundred days.



**3 Method**

Since the goal is to predict the category of the case based on its parameters, and there are multiple categories, the task becomes a classification problem. The mathematics behind stablish that the input space is divided into regions delimited by decision boundaries. In order to determine the classification regions, the method of approach used for this project was logistic regression. Hence, the non-linear function applied was:

Where represents the probability of x belonging to a class and is known as the activation function. For this case, the activation function used was the sigmoid function:

where z =

The sigmoid function denotes that the decision boundary exists when g(z) is equal to 0.5. This means that at that point, there is a probability of 50% for input x to belong to the target class. Unfortunately, this method works better only when only two categories exist. Logistic regression involves the use of the gradient descent algorithm to update after computing the gradient vector and updating the cost function multiple times to fit the parameters for minimum cost.

Gradient vector

Cost function

To assess the issue with the multi-class classification, the strategy of one versus all was applied. In python code, the sklearn library provides the LogisticRegression class to apply the mathematical steps above. The fit method from the LogisticRegression class does the classification work and determines a model that fits the input parameters to the appropriate targets.

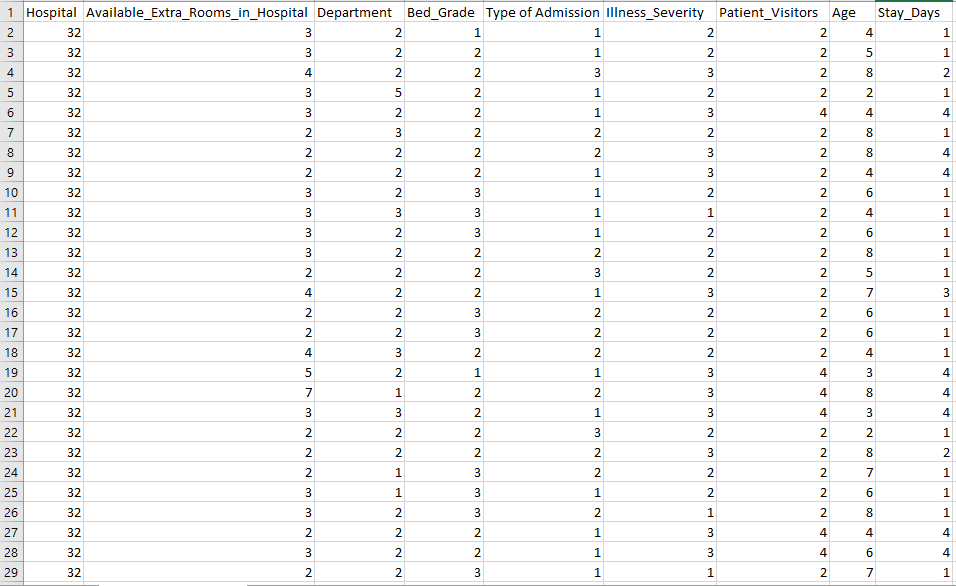
**4 Experimentation**

The first approach resulted in a software model with a 27% accuracy level. Due to this low level of accuracy, the dataset was revised and cleaned up. The non-critical features were removed based on the criteria that they should not affect the outcome. The features removed were:

* Case ID, Patient ID, City Code ID. Removed since the outcome should not be affected by the identification number of the subject.
* Hospital type, hospital city, hospital region. Removed since there is already a feature in the dataset that should cover these three.
* Ward type and ward facility.
* Admission Deposit.

Given the features removed, there were only 8 different categories left. Another change done to the dataset was the assignment of numerical values to categories within the features. Instead of having strings as values, numerical values were assigned. The last change done to the dataset was the reduction of the number of classes. Instead of having eleven, the number of classes was reduced to four:

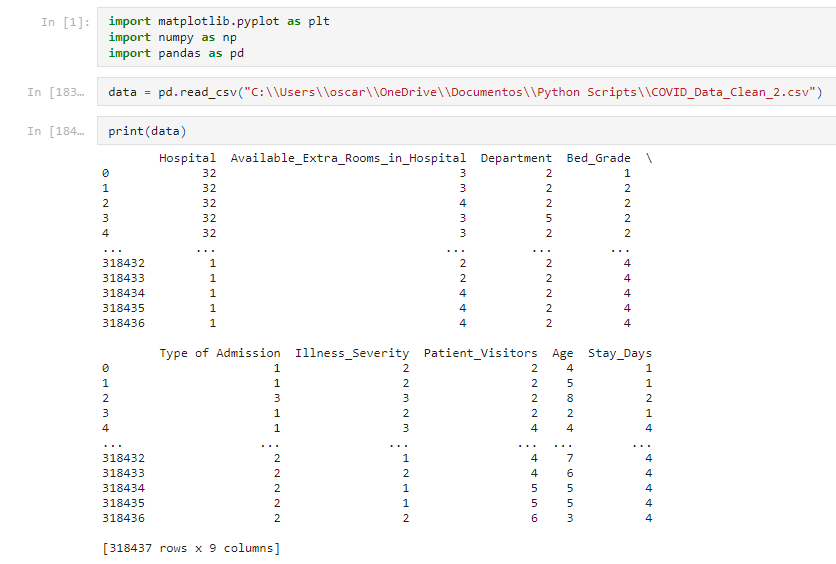
* Zero to ten days.
* Eleven to twenty days.
* Twenty-one to thirty days.
* More than thirty-one days.

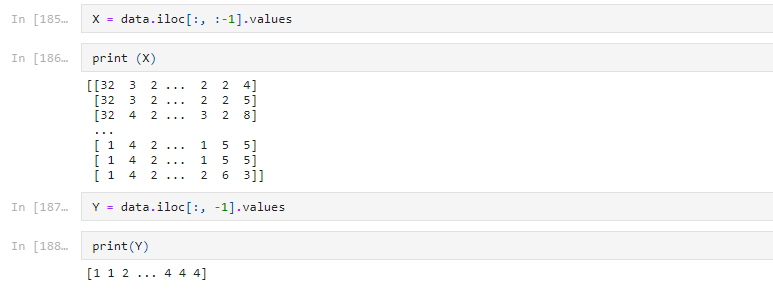


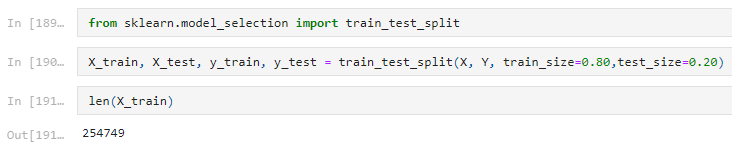
**4.1 Python Code**

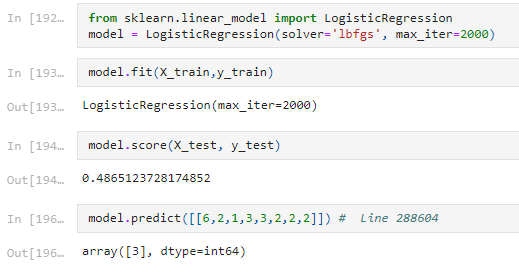
The code developed was designed with the purpose of being able for usage on any classification task using any dataset. It consists of a few steps:

1. Importing necessary libraries.
2. Reading the dataset.
3. Defining the inputs and outputs.
4. Splitting the dataset into training set and testing set.
5. Importing the necessary classes from sklearn (LogisticRegression & train\_test\_split)
6. Fitting the model using the training set and calculating the accuracy using the testing set.









**5 Results / Conclusions**

The model accuracy was 48.6%. Different tests were executed modifying the dataset by reducing the number of features and labels. None of the tests performed delivered better results than the one presented above. The last test involved only the use of two features (age and illness severity) and the accuracy level was of 40%. These poor results led to conclude that the information provided about patients is not sufficient or does not present clear patterns to build an accurate classification software. Since the dataset is new, and the challenge has not been investigated, the conclusion that a concrete pattern does not exist and therefore this issue cannot be resolved is possible.

**6 Future Studies**

This dataset can be improved by adding more features that can impact the outcome. There could me more unknown conditions that can help the tool to find a pattern. Also, the method for classification can be changed by implementing neural networks and using the softmax activation function. Changing the method could bring higher accuracies, yet if there are no patterns in the dataset, the model will keep returning low accuracy levels.

**7 References**

* Weidong Kuang. “Fundamentals of Deep Learning with Python” chapter 4 and 6.
* Mobius. COVID-19 Hospitals Treatment Plan. Obtained from <https://www.kaggle.com/arashnic/covid19-hospital-treatment>
* 3.2.4.1.5 sklearn.linear\_model.LogisticRegressionCV. Obtained from <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegressionCV.html#sklearn.linear_model.LogisticRegressionCV>