CSC 374 Project 2 - Alexis Ceballos and Yunah Kim

Task 1: Using Exploratory Data Analysis to Explore the Data

In utilizing approaches to Exploratory Data Analysis (EDA) of the provided 'COVID19MEXICO2021.csv' dataset, we were able to explore questions of how to most effectively interpret and make sense of the data to create a useful model that yields high accuracy and predictive power. The key question of interest for our EDA and corresponding model includes whether a COVID-19 diagnosis can be predicted for an individual, given specific clinical and demographic information. This clinical and demographic information provided in the dataset is organized by columns into three primary groups: medical history, demographic information, and hospitalization. Upon first creating and running a heatmap (See Figure 1) for the dataset, with no dataset columns being omitted or altered, we are able to see that, at initial glance of our original dataset, there is a box of strong positive (near 1.0) correlations between many of the medical history columns. From the heatmap, we are also able to see that for the feature correlations with the 'CLASIFICACION_FINAL' column, or row, there are relatively moderate to strong positive correlations with column features like 'SECTOR', 'INTUBADO', 'HABLA_LENGUA_INDIG', 'INDIGENA', 'TOMA_MUESTRA_LAB', 'RESULTADO_LAB', and 'UCI'. We also see relatively moderate to strong negative correlations with column features like 'ENTIDAD_UM', 'ENTIDAD_RES', 'TIPO_PACIENTE', 'EDAD', 'EMBARAZO', 'TOMA_MUESTRA_ANTIGENO', and 'RESULTADO_ANTIGENO'. For visual aid, we also ran heatmaps for each of the column groups: medical history (See Figure 2A), demographic information (See Figure 2B), and hospitalization (See Figure 2C). Such correlations drawn from our original, unaltered dataset help to inform us that some of these features should be more carefully considered and examined on a conceptual level as well as in terms of data cleanup (considering duplicates, invalid entries, and inconsistent/odd entries).

We further examined the dataset by checking the size of the dataset, which provided (8830345, 40), with 8830345 referring to the number of rows and 40 referring to the number of columns. There were also 8830345 non-null rows, which provides that at initial glance, there are no rows we would need to remove from the dataset. It should be noted that a dataset of 8830345 is large and may make creating a model through sklearn difficult, and so the data should be cleaned up and reduced to an optimal size. We also ran a check for missing zeros in the original dataset, and this provided that only column 'EDAD' had an issue with missing zeros, while other feature columns had 0 missing zeros. Later on, we also discovered that there were some oddly high values for 'EDAD' and so we eventually removed or dropped this feature from our dataset and model. Running the command *com_df.info()* showed that there were 7 different columns of data type object, while the rest were of type int64.

There is a column in the dataset named 'CLASIFICACION_FINAL', which indicates the COVID-19 diagnosis provided to the patient, and this includes values 1-7. This is an essential column for our classifications. We cleaned the information in this column, for the sake of developing our predictive models, by removing values 4, 5, and 6, which were representative of invalid tests, no lab tests, and suspicious cases, respectively, and combining values 1, 2, and 3, where 1 and 2 both represented COVID-19 positive and 3 represented SARS-CoV-2-Positive. We were able reassign these COVID-19 positive and negative case values to a binary, where positive cases (original column values of 1, 2, and 3) were reassigned to 1 and negative cases (original column value of 7) were reassigned to 0. This cleanup will allow us to more effectively apply the dataset into our predictive models for COVID-19 diagnosis based on the dataset.

We also noticed that some columns had values 99 , 98, 97, such as columns like 'DIABETES', ‘ASMA’, ‘UCI’, ‘TABAQUISMO’, RESULTADO_ANTIGENO, ‘HIPERTENSION’, and ‘OBESIDAD’. We decided to remove the rows that had these values which appeared less significant in comparison to the other values of that column, since the model would not have enough samples with such values within the dataset to hold enough predictive power.

In prioritizing certain columns of the dataset, we chose to remove some columns that would, from prior knowledge, have low expected impact on a COVID-19 diagnosis like date of update (FECHA_ACTUALIZACION) and id registration numbers ('ID_REGISTRO'). We continued to remove features, primarily demographic features, based on our understanding of COVID-19, such as origin ('ORIGEN'), sex ('SEXO'), nationality (‘NACIONALIDAD’), and ability to speak indigenous language ('HABLA_LENGUA_INDIG'). We also dropped feature columns with object data types. We dropped all variables with ‘FECHA’ (dates) as part of the variable name, such as 'FECHA_INGRESO', 'FECHA_SINTOMAS', and 'FECHA_DEF', in addition to ‘FECHA_ACTUALIZACION’. We also dropped 'TOMA_MUESTRA_LAB', 'TOMA_MUESTRA_ANTIGENO', and 'RESULTADO_LAB' because they are very closely related to the COVID-19 diagnosis, conceptually.

Ultimately, the final list of columns we dropped from the dataset we used included: 'FECHA_ACTUALIZACION', 'ORIGEN', 'SEXO', 'ID_REGISTRO', 'MUNICIPIO_RES', 'HABLA_LENGUA_INDIG', 'ENTIDAD_RES', 'TOMA_MUESTRA_LAB', 'RESULTADO_LAB', 'TOMA_MUESTRA_ANTIGENO', 'PAIS_NACIONALIDAD', 'PAIS_ORIGEN', 'FECHA_INGRESO', 'FECHA_SINTOMAS', 'FECHA_DEF', 'INTUBADO', 'EMBARAZO', 'NACIONALIDAD', 'INDIGENA', 'MIGRANTE', 'EDAD'. After dropping these columns, we checked for duplicate rows because we dropped columns, which could create duplicate rows with less features involved for our modeling. This puts our working dataset size as 180165 rows and 18 columns. Looking at Figure 3 and Figure 4, we see that the number of positive COVID-19 cases and negative COVID-19 cases are more balanced after dropping our selected feature columns. From here, we isolated the column 'CLASIFICACION_FINAL' by assigning it to y and set X to the dataset with the 'CLASIFICACION_FINAL' column dropped in order to prepare our data for our predictive models.

Task 2: Exploring the Performance of Supervised Classification Methods and Ensemble Learning Methods

Following such approaches EDA, we were able to apply our data to different learning models in efforts to predict COVID-19 diagnosis for an individual given clinical and demographic information (though now we see greater dependence on clinical information). In exploring the performance of selected supervised classification methods to solve the defined problem, we applied the data to a decision tree classifier model. To create this model, we needed to split the dataset into two datasets: a train dataset and a test dataset. Using the model_selection module in sklearn, we applied our previously defined X and y to the train_test_split function, where we specified the test_size parameter as 0.25 or 25%, as well as a random_state of 42. We decided to use a test size of 25% of the 180,165 rows of data we narrowed our dataset to (discussed in task 1 section of report), based on research we conducted on EDAs and predictive learning models, and best practices. Further, based on our EDA, we can assume that the applied train_test_split is representative in terms of the dataset, and we are able to double check this with our decision tree model's confusion matrix afterwards.

Initially, when we only dropped columns 'FECHA_ACTUALIZACION', 'ORIGEN', 'SEXO', 'ID_REGISTRO', 'HABLA_LENGUA_INDIG', 'MUNICIPIO_RES', 'FECHA_INGRESO', 'FECHA_SINTOMAS', 'FECHA_DEF', 'PAIS_NACIONALIDAD', 'PAIS_ORIGEN', and 'INTUBADO', and created a decision tree model with specified parameters of max_depth =30 and min_samples_split =30, with min_impurity_decrease=0.01, leaving the rest of the parameters at default, the accuracy score of the model was 0.97. Most of our initial run throughs for the decision tree model fluctuated around scores of 96 and 98%. Dropping 'EDAD' on top of the previously mentioned columns, along with these initial parameters, gave an accuracy score of 0.98, in alignment with the 96-98% that was mentioned. For our first tests of the decision tree model, these accuracy scores were quite high. This prompted us to reconsider what columns we were using and reevaluate our methods. After going back to the drawing board and running more tests of new decision tree models, we saw that there were additional columns we needed to drop. We ran several other decision tree models with changed columns in use, as well as small changes to the parameters and got accuracy scores like 0.78. We also reevaluated our datasets values and looked for invalid entries or "odd" entries, and decided whether or not to remove them. Some of these details are included in the EDA report section.

For our final decision tree model, we dropped columns of 'FECHA_ACTUALIZACION', 'ORIGEN', 'SEXO', 'ID_REGISTRO', 'MUNICIPIO_RES', 'HABLA_LENGUA_INDIG', 'ENTIDAD_RES', 'TOMA_MUESTRA_LAB', 'RESULTADO_LAB', 'TOMA_MUESTRA_ANTIGENO', 'PAIS_NACIONALIDAD', 'PAIS_ORIGEN', 'FECHA_INGRESO', 'FECHA_SINTOMAS', 'FECHA_DEF', 'INTUBADO', 'EMBARAZO', 'NACIONALIDAD', 'INDIGENA', 'MIGRANTE', 'EDAD'.

Using these dropped columns, after much trial and error, we decided to look more into specifying and changing parameters in the decision tree model. We tested our decision tree model with different values for the following parameters: max_depth, min_samples_split, min_samples_split to evaluate the accuracy of the model. In doing so, we got a range of accuracy from 50% - 86% and we decided to choose the parameters values that gave us the highest accuracy for the model. Those parameters values were the following: max_depth =30 and min_samples_split=30, and this provided an accuracy score of 0.86, which can also be referred to as 86%.

Looking at the confusion matrix for this decision tree model (See Figure 5), we see that the numbers are relatively well balanced when considering the number of true label and predicted label classifications against each other. The confusion matrix shows 20954 (approximately 46.52%) accurate negative COVID-19 diagnosis predictions and 17634 (approximately 39.15%) accurate positive COVID-19 predictions. This aligns with the dataset we are using because we have a slightly greater number of negative cases (as seen in Figure 4). Additionally, there are 4673 (approximately 10.37%) false negatives and 1781(approximately 3.95%) false positives.

We also did the same procedure as before, we tested different values for the following parameters for the random forest model: min_samples_split, and n_estimators. The big difference between the accuracy of this model was adding max_depth to the model which added about 10% accuracy to the model when the value was greater than 7, and the n_estimators was = 10. From there on we only changed the value of min_samples_split and that gave us a better accuracy for the model. The random forest model we ended up creating used the same dropped columns, but with specified hyperparameters of max_depth = 10, min_samples_split= 100, and n_estimators= 0. This model yielded an accuracy score of .88 or 88 %.

According to the confusion matrix (See Figure 6) of the random forest classifier model that we developed, the model is better than the decision tree model at classifying the true negative labels. The random forest confusion matrix yields that there are 22123 (approximately 49.12%) accurate negative COVID-19 diagnosis predictions and 17240 (approximately 38.28%) accurate positive COVID-19 predictions. There are 5067 (approximately 11.25%) false negatives and 612 (approximately 1.36%) false positives. From this, as mentioned above, we are able to draw that the random forest classifier model is more accurately predicting a higher number of true negatives in comparison to the decision tree. We can also draw from this confusion matrix that the classifications, with predicted and true labels against each other, are relatively balanced, providing helpful insight into the predictive model based on our dataset.

Future endeavors should investigate this dataset applied to other Ensemble Learning Methods like the bagging classifier, as well as potentially other supervised learning classification methods. For the sake of this project, and due to time constraints, we decided to focus on decision trees and random forest models to apply this dataset in exploration of our goal to predict COVID-19 diagnosis on the basis of information on a patient/individual.

Figures

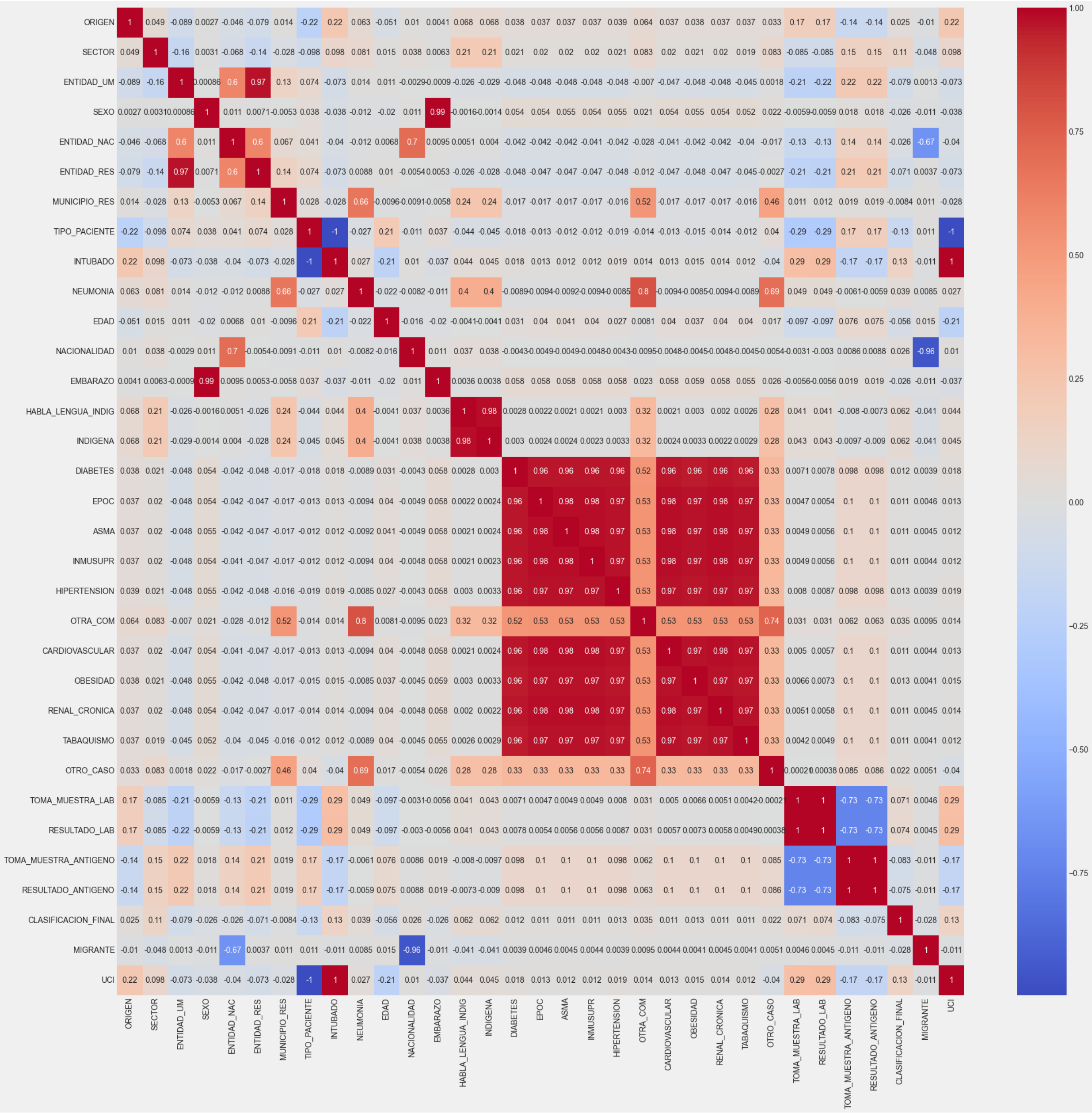

Figure 1. Original heatmap of dataset column correlations (with no columns excluded)

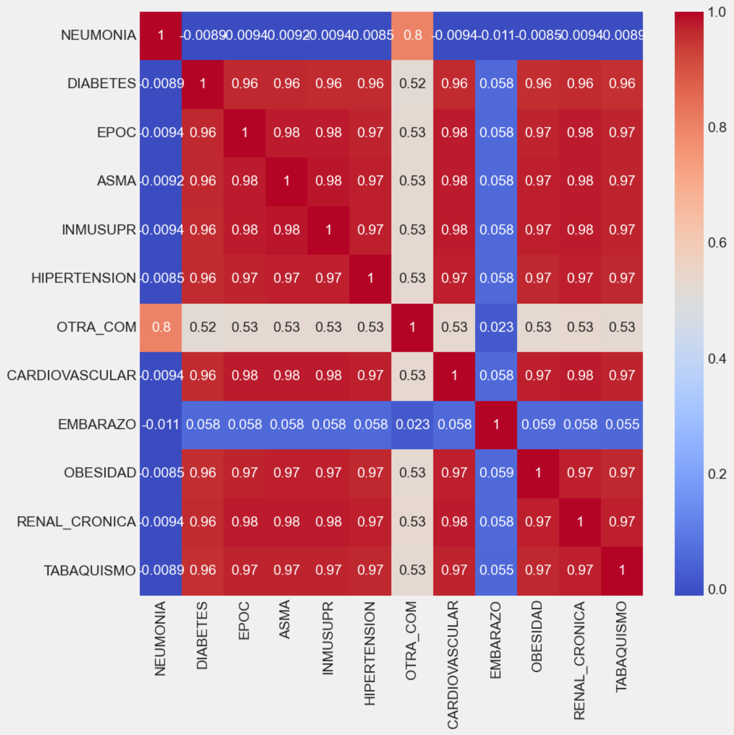

Figure 2A. Heatmap for Medical History column features (using original, unaltered dataset).

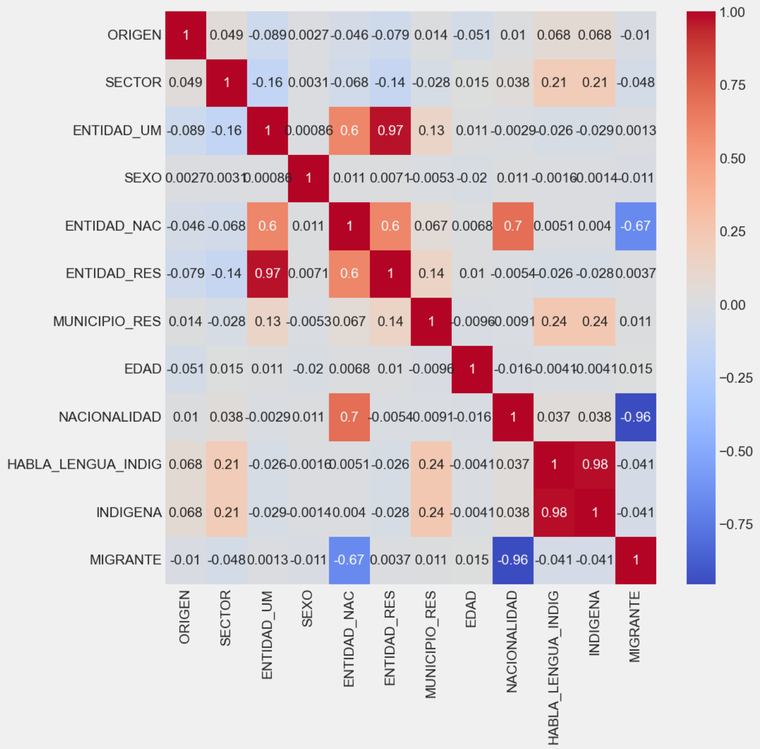

Figure 2B. Heatmap for Medical History column features (using original, unaltered dataset).

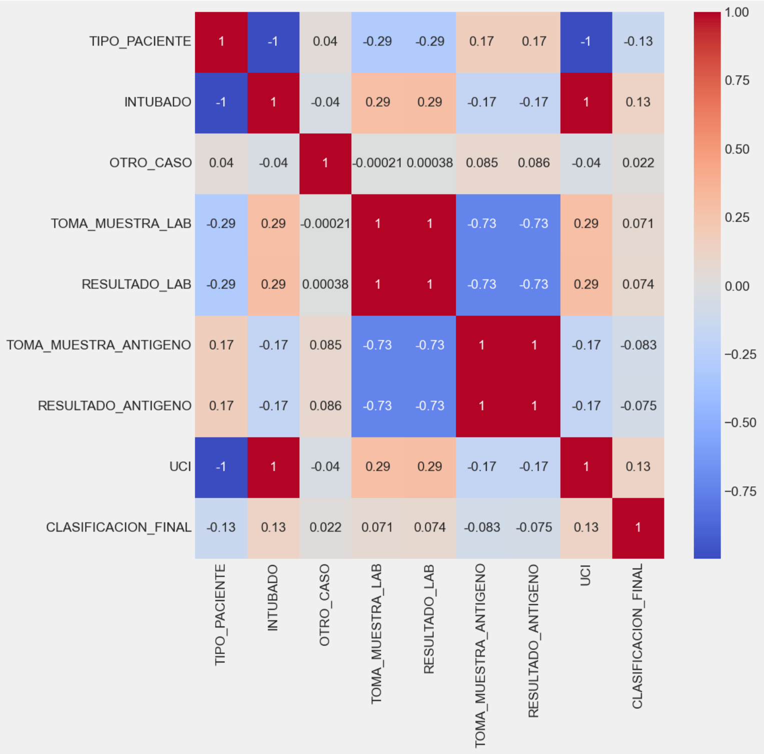

Figure 2C. Heatmap for Hospitalization column features (using original, unaltered dataset).

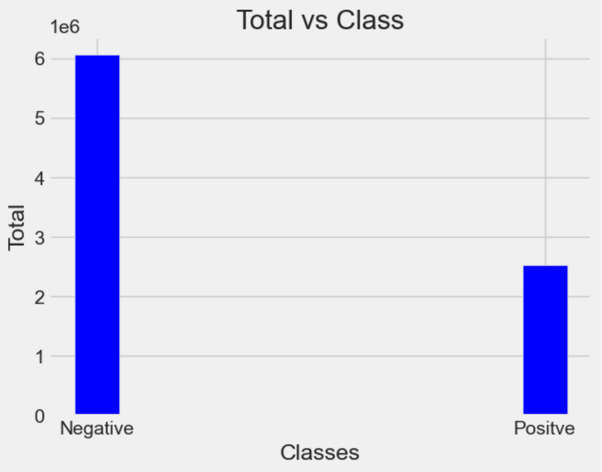

Figure 3. Based on original dataset, graph of positive COVID-19 cases and negative COVID-19 cases. Note the imbalance between the two.

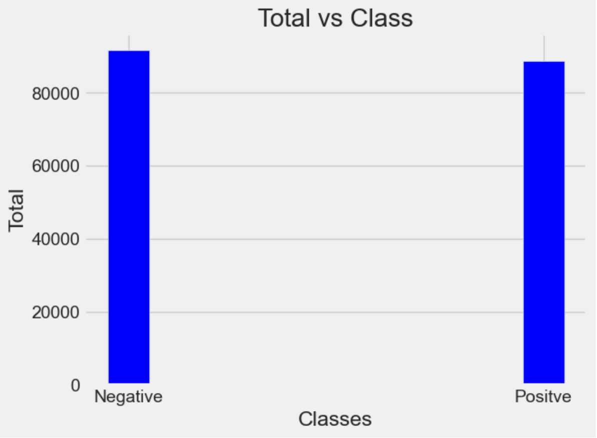

Figure 4. After dropping our selected feature columns, this is a graph of the positive COVID-19 cases and negative COVID-19 cases. Note that this is much more balanced than the previous in Figure 3.

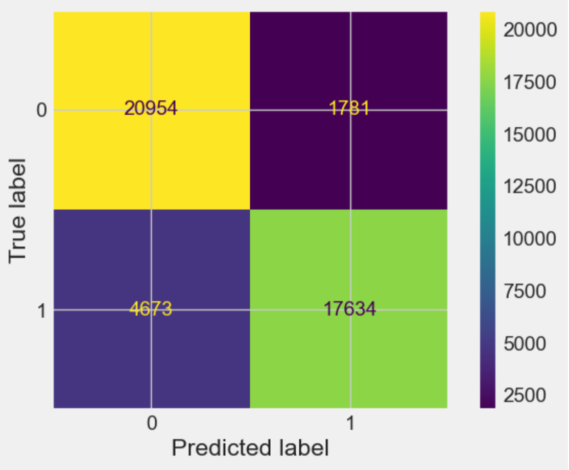

Figure 5. Confusion matrix for the decision tree model with the hyperparameters of max_depth = 30, and min_samples_split= 30. The model accuracy was .86 or 86 %.

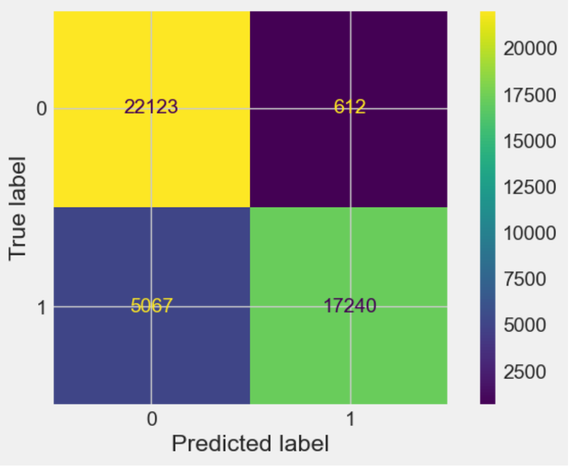

Figure 6. Confusion matrix for the Random Forest model with the hyperparameters of max_depth = 10, and min_samples_split= 100, n_estimators= 0. The model accuracy was .88 or 88 %.

Sources

https://www.analyticsvidhya.com/blog/2021/08/how-to-perform-exploratory-data-analysis-a-guide-for-beginners/
https://www.analyticsvidhya.com/blog/2022/07/step-by-step-exploratory-data-analysis-eda-using-python/
https://www.geeksforgeeks.org/how-to-do-train-test-split-using-sklearn-in-python/
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
https://medium.com/analytics-vidhya/understand-train-and-test-split-on-your-data-set-ml-process-cont-fbad7c497850
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop.html
https://www.geeksforgeeks.org/ensemble-methods-in-python/
https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop_duplicates.html
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html#sklearn.ensemble.BaggingClassifier
https://www.geeksforgeeks.org/ml-bagging-classifier/