SMU

# DATA MINING – PROJECT 3

# Classification based on COVID-19 Dataset

**Varun Singh**

# Abstract

In this project report, we undertake the task of building a classification model after looking at the different features in the Covid-19 census dataset.
Aimed to predict at risk counties, we analyze the economic factors influencing the demographic makeup of counties. This demographic break up is based on socio economic factors such as income and less educated people, etc.

We have used the data for all the counties in the United States (except Texas), to learn and build a model. And then we predict at risk counties in the state of Texas.

We want equitable sharing of resources in our society. So, we want to present this project report as a potential guide on how to allocate resources and fight whatever is coming our way next. We hope to build a model, based on the Covid-19 Census data, and want to identify the best combination of Classification techniques to be able to achieve generalization where our model can be replicated in any other region or country in the world, only prerequisite is the availability of data about the features we are interested in, which are socio economic indicators.

We will use different techniques in the CRISP-DM model to perform scaling and manipulation of the data in such a way that the final model we create will be able to perform well on data collected from different parts of the world. We will use a wide variety of classification algorithms and then evaluate the performance of the model based on different metrics such as sensitivity, specificity, etc.

# Business Understanding

This project report is aimed to provide insights to the Government officials running the state of Texas. This group of people performs a wide variety of jobs, they handle the budget of the state, the procurement and allocation of resources, fighting back against dire situations like Covid, handle any emergency response required by the state, frame policy for the citizens of the state, etc. These are just some of the jobs, with respect to Covid-19 response that are the responsibility of the elected government officials.

By providing this report, these people will have an additional tool to use to offer the best response to another Covid outbreak or any other similar emergency, like an outbreak of a novel deadly virus. By studying the results of this report, a person will be able to look at the results of different classification models. By analyzing the results of classification algorithms based on the selected features, a lawmaker would be better aware of the counties where people might be at a higher risk of getting Covid, or worse, dying from it.

We aim to analyze different classification models and select the one that performs the best for the features that we selected. This will potentially help to direct the response to another outbreak in the right direction, since we will have predicted which counties in Texas are at a higher risk than others.

For the task of training and testing, we decided to select all the counties in the US, except Texas. There are 3,143 counties in the US and there are 254 counties in the state of Texas. By considering rest of US counties as training data and Texas counties as testing data, our training and testing split of the dataset came out to around 90:10. We are aware that 90:10 is not the most optimum split, but we decided to move ahead with it. Since I found the task of being able to predict the extent of covid cases and deaths for a single state like Texas, based on the rest of the country very interesting.

Another reason for the selection of this task is that Texas is the largest state in US, with the greatest number of counties. And it is a state which is a balance of both congested and highly populated urban areas like New York City or San Francisco, and less densely populated rural areas.

# Data Description and Feature Selection

We have available with us the US covid 19 and census dataset. We wanted to continue our hypothesis and keep looking for a relationship between the number of cases and deaths and other effects of Covid-19 and its relationship with Socio Economic features. In the last project we selected Employment type and Economic type features from the dataset. To be able to look at the patterns in a classless dataset.

This time for the task of supervised learning, we decided to go with only the Economic features. These features include median_rent, median_income, poverty, income_per_capita, households_on_public_assistance and GINI index. We use these features to train our models.

We manipulated some features and calculated metrics per 1000 people. The reason for doing this is that we wanted the values to be averages, and not be influenced by counties with high populations. We did this for deaths, confirmed_cases, poverty and households_on_public_assistance. Other columns median_rent, median_income, income_per_capita and GINI index are metrics that don't need to be modified.

For the class/target variable, we select the number of deaths per 1000 people in all the counties in the US and take the average. Once we identified the average number of deaths in all counties per 1000 people, we could classify counties into low risk and high-risk categories.

We also selected another class/target variable, average of the number of confirmed cases per 1000 people in all counties. Again, we did binary classification of counties, categorizing them into low risk and high-risk categories.

So, we built a model to do classification based on both the number of deaths and confirmed cases.

# Column Names and Description

**confirmed cases(Scale: Ratio):** This numerical column represents counts of confirmed COVID-19 cases. The data is on a ratio scale, meaning it has a true zero point, and ratios of values are meaningful (e.g., one county having twice as many cases as another).

**deaths (Scale: Ratio):** This column represents counts of COVID-19-related deaths. It's also on a ratio scale.

**median_income(Scale: Ratio):** A numerical column representing the median income of residents in each county. It's on a ratio scale, allowing for meaningful comparisons and calculations.

**income_per_capita(Scale: Ratio):** This numerical column represents the income per capita in each county. It’s on a ratio scale.

**median_rent(Scale: Ratio):** A numerical column representing the median rent in each county. It's on a ratio scale, providing a true zero point and meaningful ratios between values.

**gini_index:** A numerical column indicates the gini index in each county. The Gini index of 0 represents perfect equality, while an index of 1 implies perfect inequality.

**households_public_asst_or_food_stamps(Scale: Ratio):** This numerical column represents the households of public asst or food stamps in each county. It’s on a ratio scale.

**poverty(Scale: Ratio):** Numerical data column of poverty in each county. It's on a ratio scale, allowing for meaningful comparisons and calculations.

# Data Preparation

Since we are doing a parallel classification task, we will create two identical datasets, the only difference will be the target variable. We decided to use both the number of confirmed cases and deaths for the target variable. All the features remain the same in both the datasets, only difference is the target variable.

Table 1 - Statistical Summary of features

| | MIN | 1st Qu | Median | Mean | 3rd Q | MAX |
|---|---|---|---|---|---|---|
| Confirmed_Cases | 0 | 58.56 | 75.72 | 76.80 | 92.99 | 316.1 |
| Deaths | 0 | 0.682 | 1.189 | 1.336 | 1.752 | 8.359 |
| Median_Income | 19264 | 41121 | 48049 | 49737 | 55761 | 129588 |
| Median_rent | 140 | 424 | 510.5 | 563.4 | 642 | 1879 |
| Income_per_capita | 9334 | 21805 | 25271 | 26036 | 29123 | 69529 |
| Poverty | 24.24 | 109.53 | 146.18 | 153.79 | 186.94 | 513.68 |
| Gini Index | 0.3271 | 0.4211 | 0.4423 | 0.4449 | 0.4665 | 0.5976 |
| Households_on_food_stamps | 0 | 96.74 | 135.78 | 143.06 | 179.57 | 641.55 |

While preparing the data for the task of classification of counties in the State of Texas. We decided to obviously take average values of features per 1000 people. We did this so that there is some equity in terms of the integrity of the data and to avoid misrepresentation. A highly populated area would do the worst in all aspects if we had looked at the absolute numbers instead of per 1000 people.

We calculated averages for Poverty, Households_on_Public_Assistance, Confirmed_Cases and Deaths. The rest of the selected features like Gini Index, median income, etc. do not require this adjustment.

After this step, we decided to scale the values of the features that we had already identified as the ones we will base this classification task on. We used the scale() function in R to do this. After this we decided to drop NA values using the na.omit() function in R.

After this was done, we calculated the average number of deaths per 1000 people in all the counties in the US, except of course Texas.

Table 2 - Statistical Summary of Deaths column

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|
| 0.0000 | 0.6813 | 1.1867 | 1.3355 | 1.7519 | 8.3587 |

We also calculated the average number of confirmed cases per 1000 people in all the counties in the US, except of course Texas.

Table 3 - Statistical Summary of features

| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
|---|---|---|---|---|---|
| 0.0000 | 58.55 | 75.51 | 76.78 | 92.99 | 316.10 |

Now, we have identified the threshold to classify any county in the US to label it as doing better or worse than average, low risk or high risk. For the average number of deaths per 1000 people the mean = 1.3355, and the average number of confirmed cases per 1000 people the mean = 76.78.

These are the target variables, if a county has a higher death rate per 1000 people than mean, we categorize it as high risk, or low risk county if it has death rate lower than average.

We do the same for confirmed cases. If the number of confirmed cases per 1000 people for a county is higher than 76.78 then it is a high-risk county. Or it might be low risk county if the confirmed cases per 1000 people is less than the mean.

Now once we had processed the data and added target features, we had two slightly different datasets in our hands. Now we split the dataset into training and testing. We used a filter to separate Texas and non-Texas counties. Texas counties go into the testing_data, while the training_data will be used to train the model.

Now we looked at the correlation between the variables that we had selected for model building. From this matrix we can clearly see that the target variable is highly correlated to gini_index, poverty and households_on_public_assistance while income_per_capita, median_income and median rent are inversely correlated to the class variable. This is a good sign, that our intuition and hypothesis that counties with high average income are less likely to be adversely affected by Covid-19. All the parameters whose high value suggest that the standard of living in the counties with respect earnings and dependance upon public assistance, are highly correlated with the high-risk categories of counties. The counties that we classified as high risk based on above average number of deaths and cases.

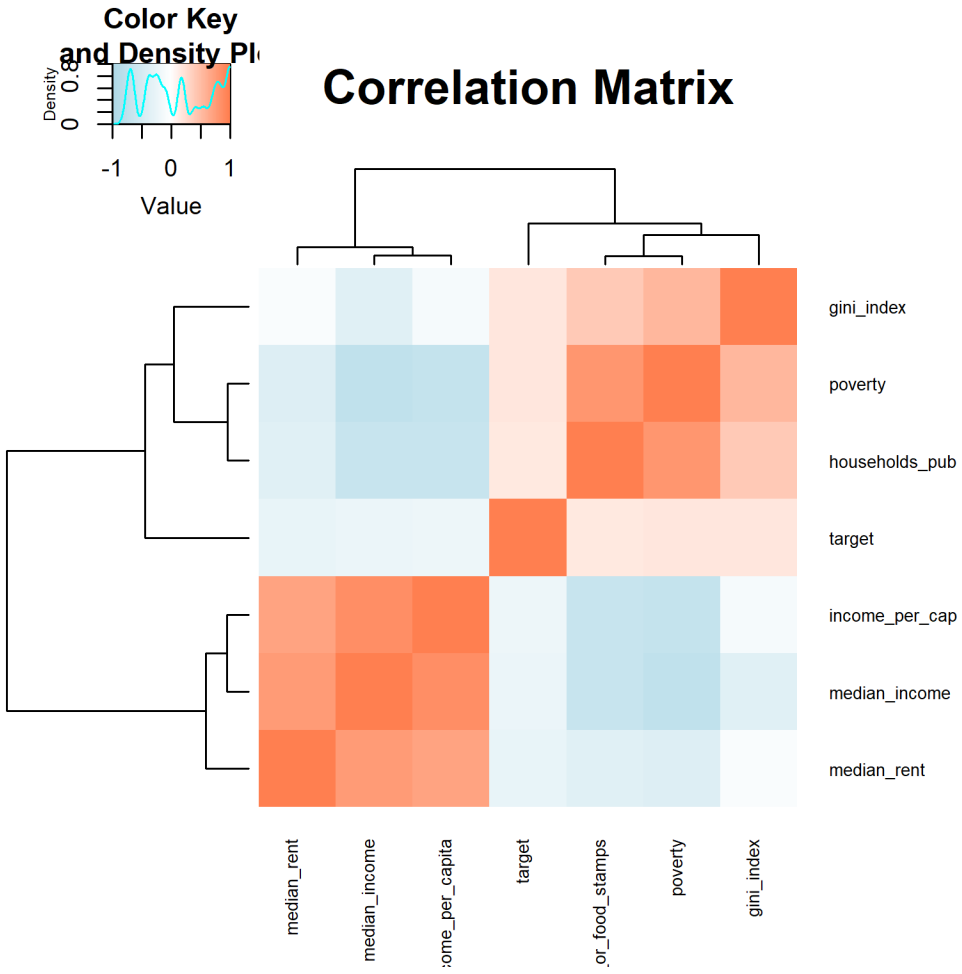


Fig 1 - Correlation Matrix of Features

We visualize the results of the dataset that we have prepared for building a model. We print a map of the United States with the high-risk categories colored in red for counties with deaths greater than the national average that we calculated earlier.

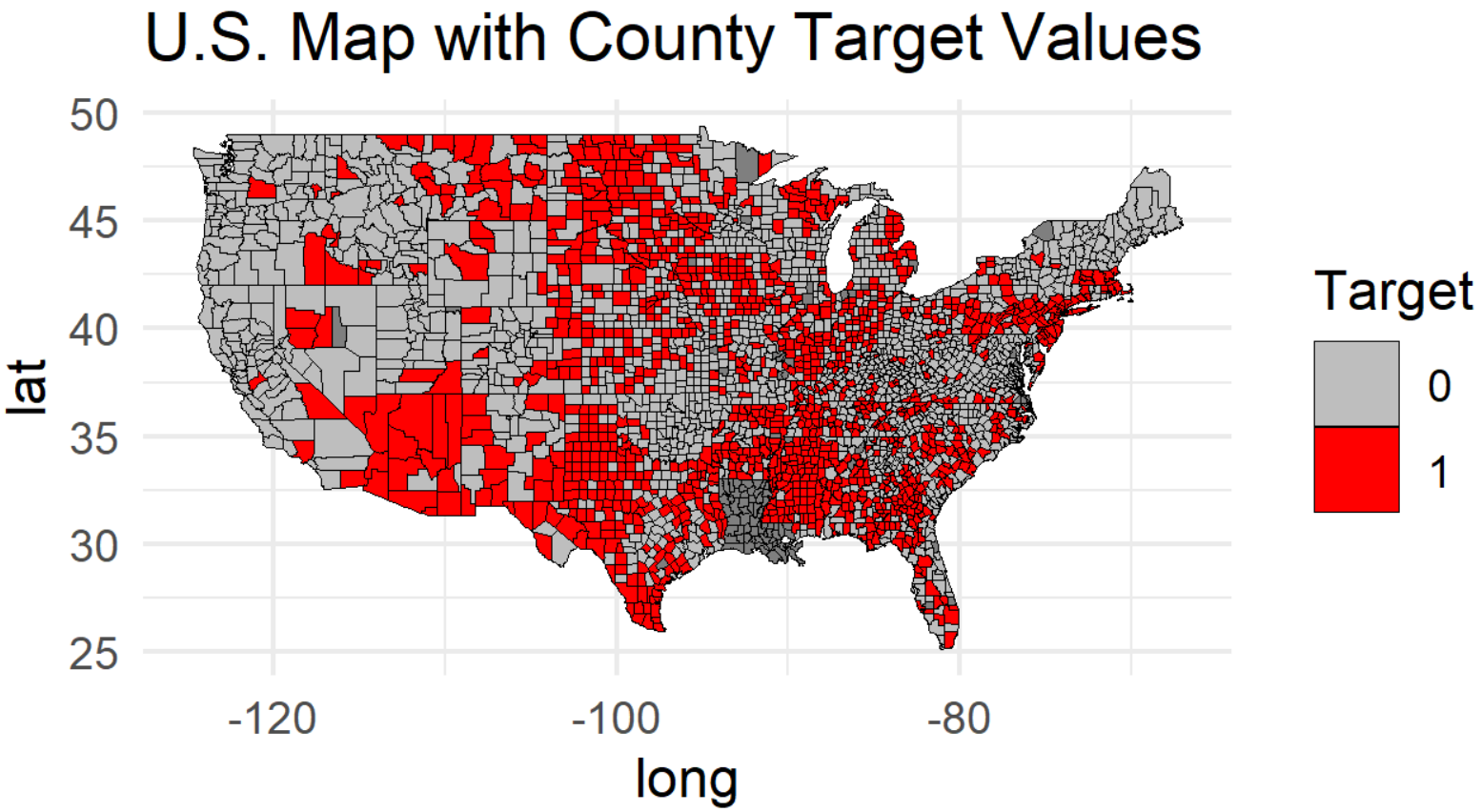


Fig 2 - High Risk Counties based on Deaths.

Now, we visualize the map of the US but now categorize counties as high risk or low risk based on the number of confirmed cases. Fig - 3 shows the map of the US with the high risk counties colored in Blue.

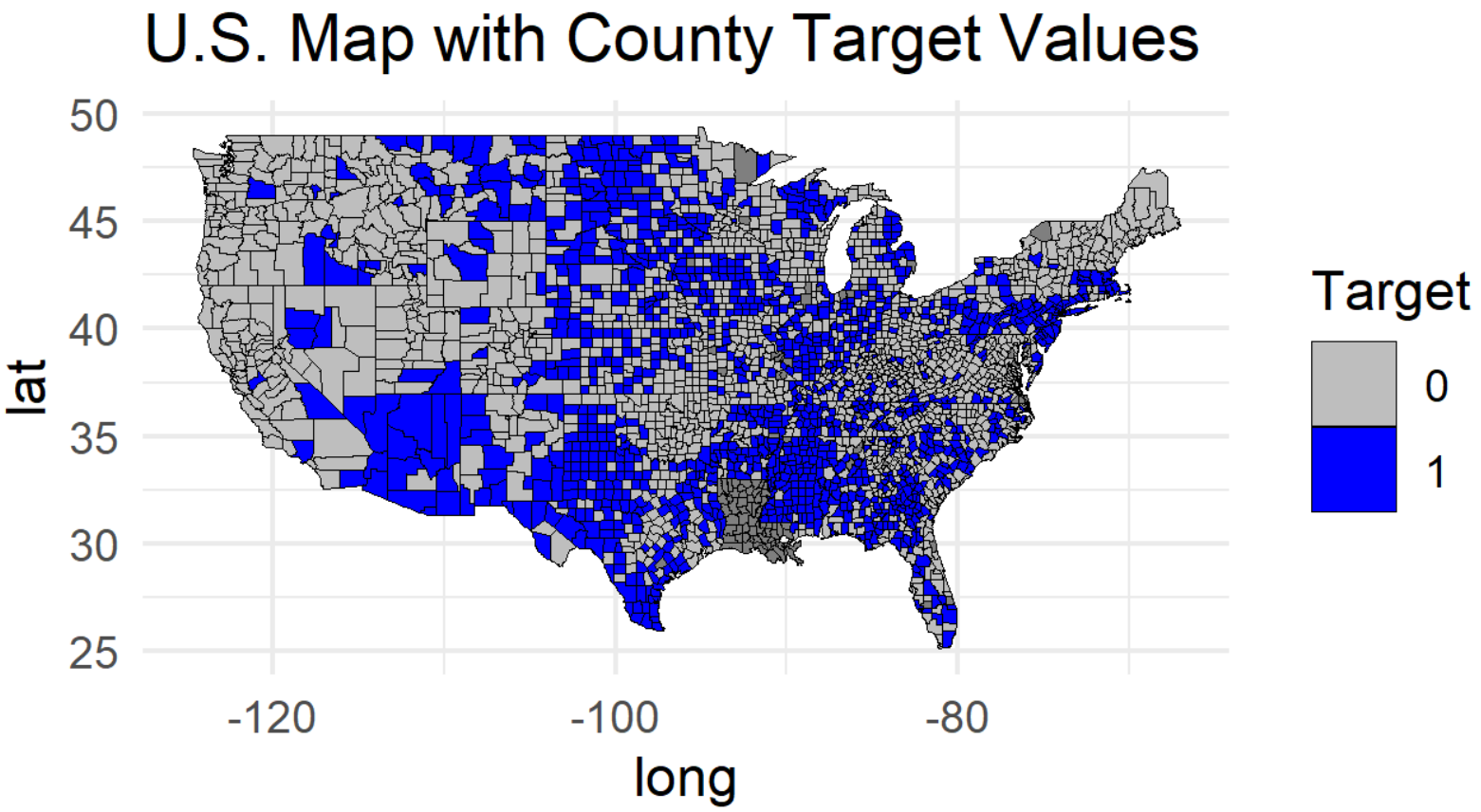


Fig 3 - High Risk Counties based on Confirmed Cases

Now that we have visualized the map of the US, which is essentially our training data. Fig 2 and Fig 3 both show at risk counties based on high number of deaths or the number of confirmed cases. Red has

been used to represent Deaths and blue for Confirmed Cases. We have used this convention throughout this report.

If we look closely at fig 2 and fig 3, we see that there is a lot of similarities in the classification derived from the two target variables.

Now let's look at the visualization of the state of Texas. This is going to be our frame of reference when validating our classification models. This is the visualization of our testing set. The red colored counties on the map are the ones which are at higher risk when taking deaths into account.

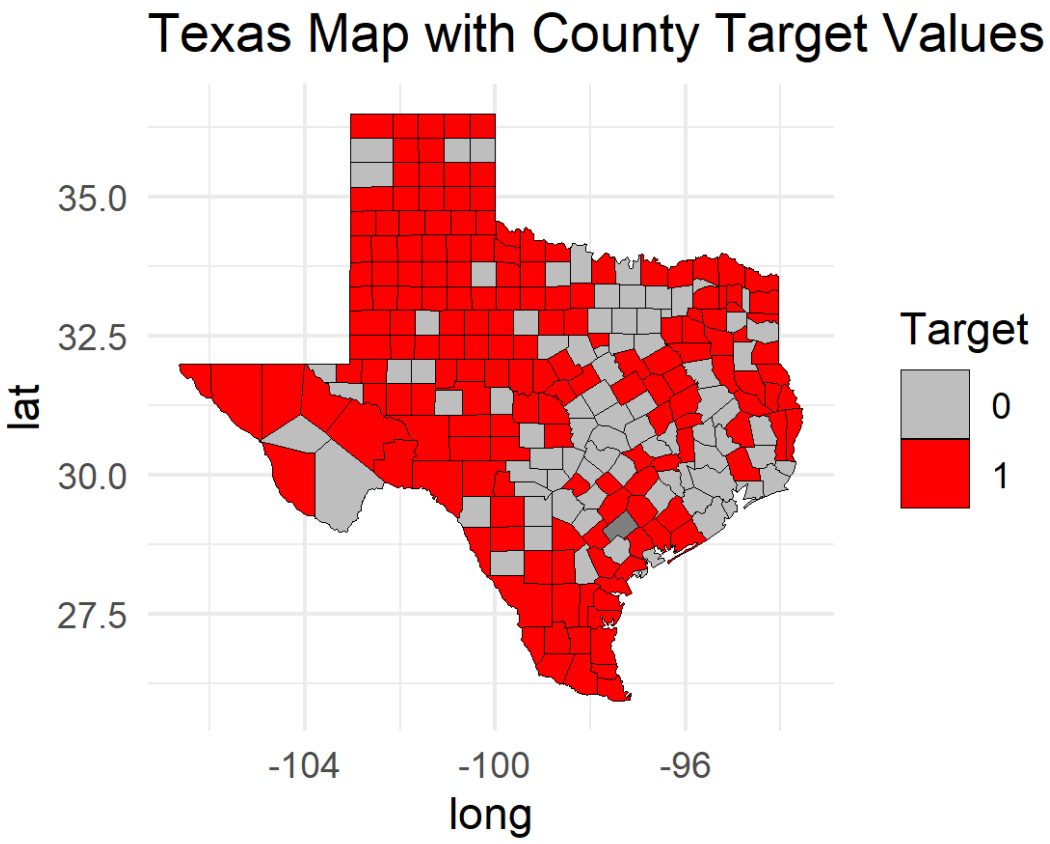


Fig 4 - High Risk counties in Texas based on Deaths

Now we look at the counties classified as high risk based on the number of Confirmed Cases. The below maps show counties in Texas having higher than national average number of cases. And both the maps, fig 4 and 5 look very similar.

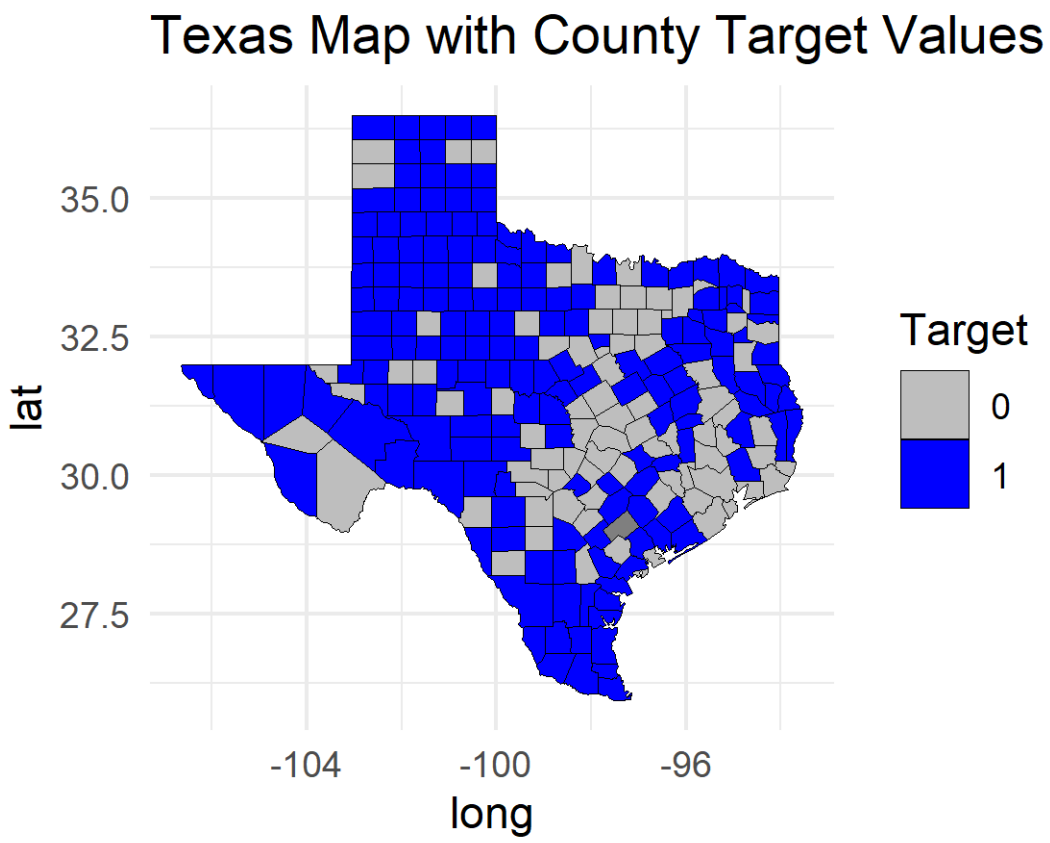


Fig 5 - High Risk counties in Texas based on Confirmed Cases

# Modeling

## Random Forest

Random Forest algorithm works by using subsets of the data and creating decision trees from it. Now based on the different decision trees it averages out the result of each decision tree and identifies the important variables along the way. It averages out the outputs of different decision trees and makes the final classification decision.

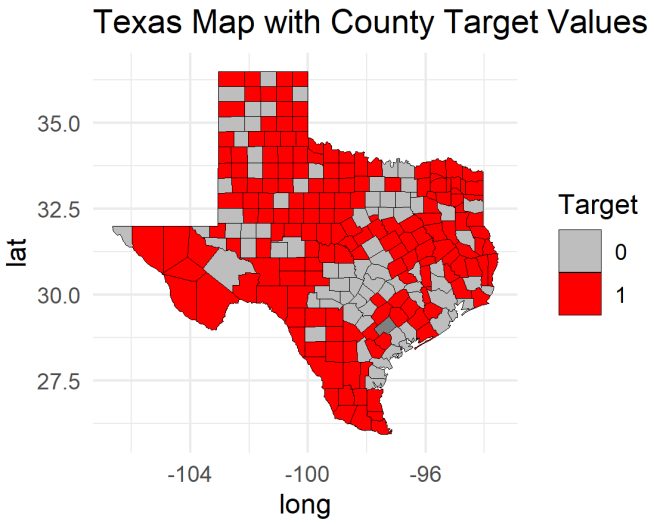


Fig 6 - Random Forest Prediction (Deaths)

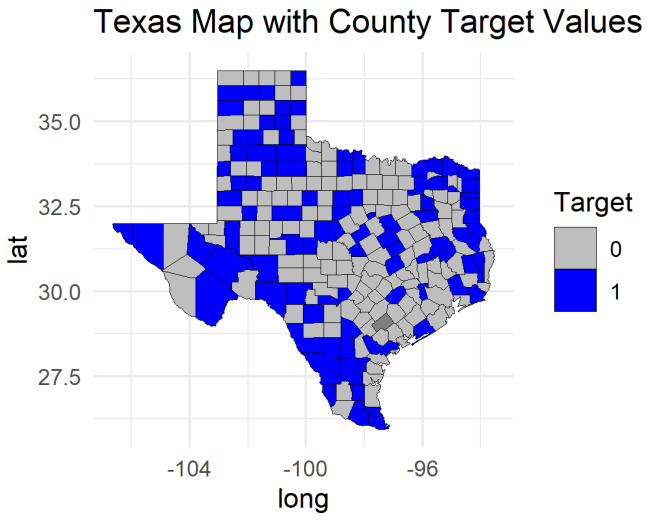


Fig 7 - Random Forest Prediction (Confirmed Cases)

Figs 6 and 7 show the prediction based on Random Forest algorithm for the two different targets. Red represents Deaths while Blue represents higher than average confirmed cases.

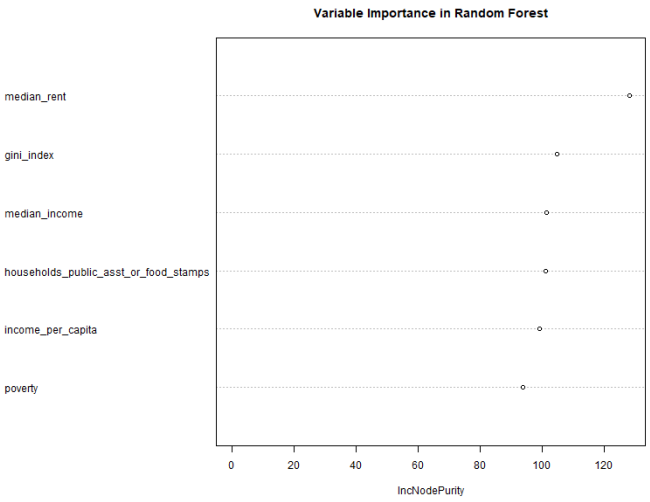


Fig 8 - Variable Importance for Deaths Model

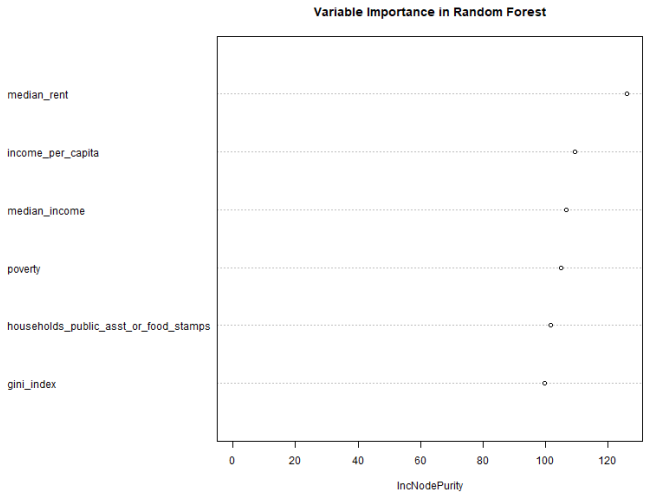


Fig 9 - Variable Importance for Cases Model

In the above figure, Fig 8 and 9, we see a variable importance plot for the Random Forest Models that we created. The plots show the ranking of the features that we used to train the models. Fig 8 shows the variable importance of the features in the Model where target variable was based on the average of Deaths in all the counties in the US. Fig 9 shows the equivalent for the Model where target variable was based on the average number of Confirmed Cases in all the counties of US. In both the plots we see that the median_rent gets the highest rank. This means that this variable helped create and influenced the model the most out of all the selected features.

The two plots differ in rest of the features ranks. For the model creates using number of deaths, median_rent, gini_index and median_income features influenced the model the most. While for the model created using number of confirmed cases of covid 19, median_rent, income_per_capita and median_income were the most important features.

## Model Evaluation

### Random Forest Model based on number of Deaths.

Table 4 - Confusion Matrix

| | | Predicted | |
|---|---|---|---|
| **Actual** | | Negative | Positive |
| | Negative | 47 | 37 |
| | Positive | 35 | 135 |

Table 5 - Evaluation Metrics

| Evaluation Metric | Score |
|---|---|
| **Accuracy** | 0.71653 |
| **Precision** | 0.79411 |
| **Recall** | 0.78488 |
| **F1 score** | 0.78947 |

**Random Forest Model based on number of Confirmed Cases.**

Table 6 - Confusion Matrix

| | | **Predicted** | |
|---|---|---|---|
| **Actual** | | Negative | Positive |
| | Negative | 130 | 3 |
| | Positive | 115 | 6 |

Table 7 - Evaluation Metrics

| Evaluation Metric | Score |
|---|---|
| **Accuracy** | 0.53543 |
| **Precision** | 0.49686 |
| **Recall** | 0.66667 |
| **F1 score** | 0.09230 |

# SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine is used to build models when the number of dimensions is very high in the data. Our data doesn't have too many dimensions but still we wanted to look at the performance of a model based on SVM.

SVM tries to find the hyperplane, a decision boundary which classifies data points into categories. The hyperplane should be such that the margin should be the highest. Margin is essentially the distance between the hyperplane and the closest point from each class.

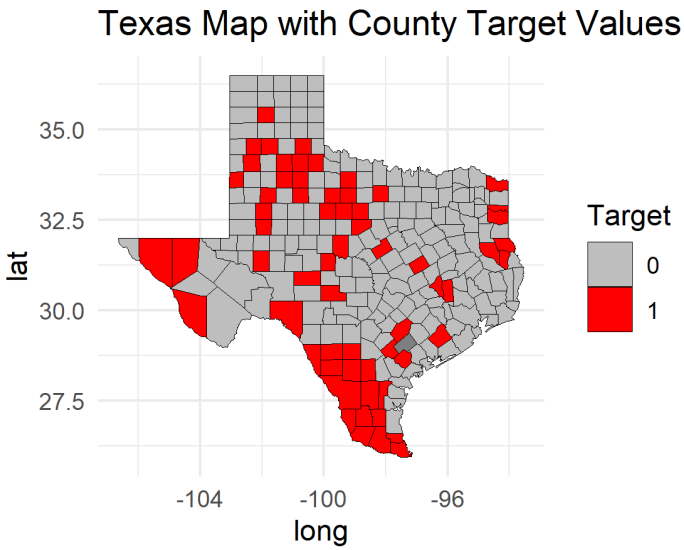


Fig 10 SVM Prediction(Deaths)

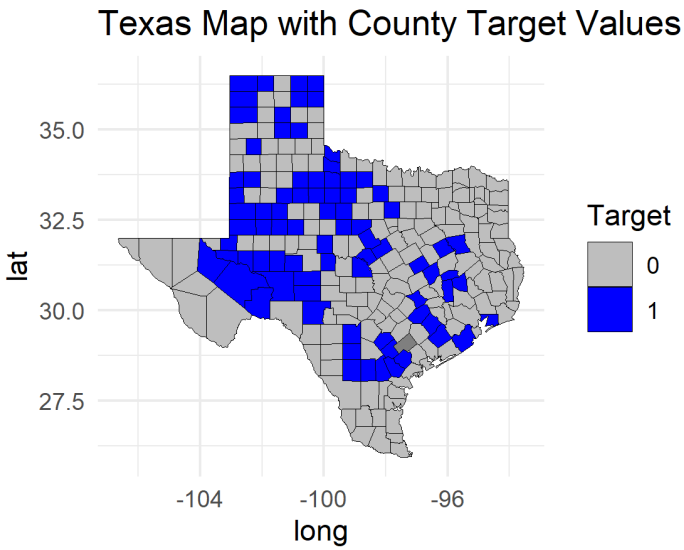


Fig 11 SVM Prediction(Confirmed Cases)

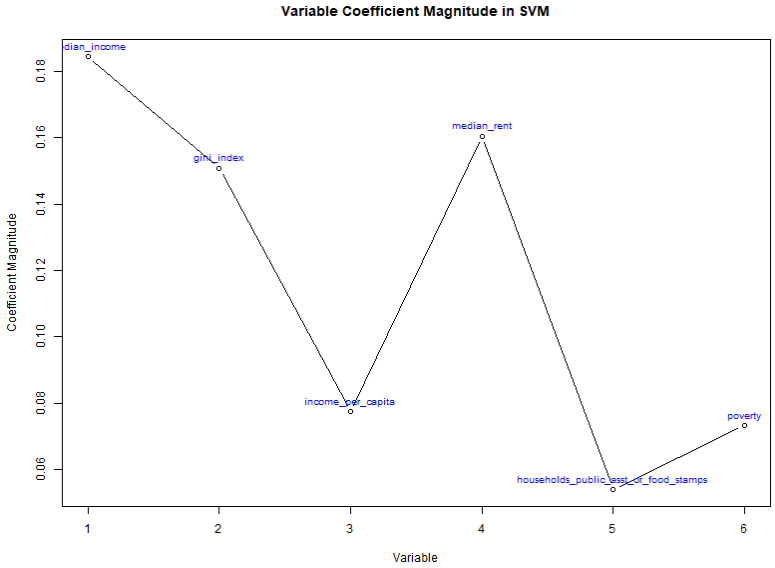


Fig 12 - SVM coefficients deaths model

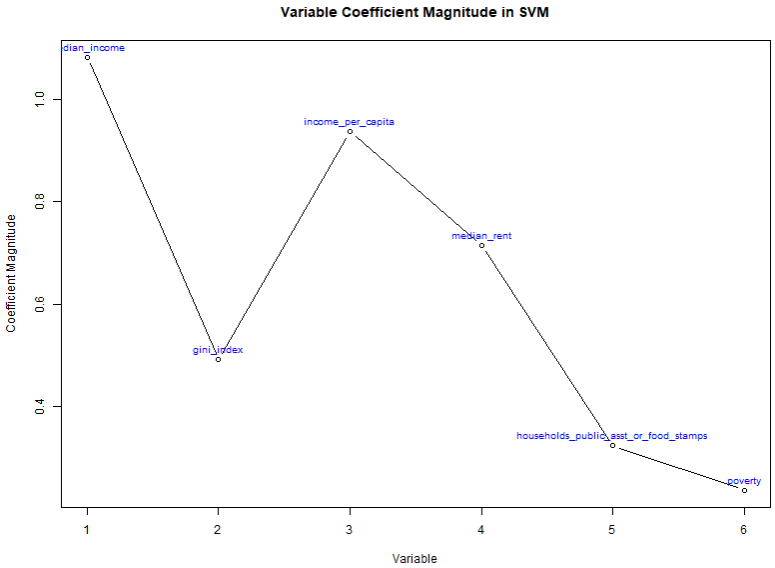


Fig 13 - SVM coefficients cases model

# Model Evaluation

## Support Vector Machine Model based on number of Deaths.

Table 8 - Confusion Matrix

| | | Predicted | |
|---|---|---|---|
| **Actual** | | Negative | Positive |
| | Negative | 75 | 9 |
| | Positive | 116 | 54 |

Table 9 - Evaluation Metrics

| Evaluation Metric | Score |
|---|---|
| **Accuracy** | 0.50787 |
| **Precision** | 0.31764 |
| **Recall** | 0.85714 |
| **F1 score** | 0.46351 |

**Support Vector Machine Model based on number of Confirmed Cases.**

Table 10 - Confusion Matrix

| | | Predicted | |
|---|---|---|---|
| **Actual** | | Negative | Positive |
| | Negative | 92 | 41 |
| | Positive | 80 | 41 |

Table 11 - Evaluation Metrics

| Evaluation Metric | Score |
|---|---|
| **Accuracy** | 0.52362 |
| **Precision** | 0.33884 |
| **Recall** | 0.5 |
| **F1 score** | 0.40394 |

# K- Nearest Neighbor -

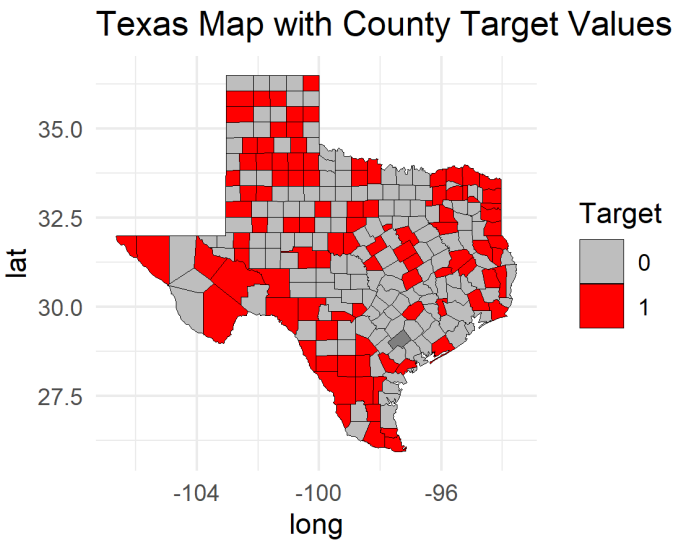


Fig 14 - KNN Prediction (Deaths)

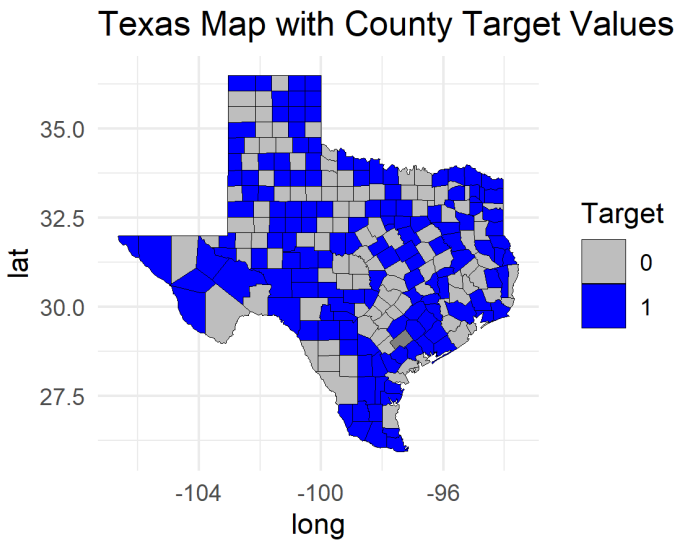


Fig 15 - KNN Prediction (Confirmed Cases)

## Model Evaluation

**K-Nearest Neighbor Model based on number of Deaths.**

Table 12 - Confusion Matrix

| | | Predicted | |
|---|---|---|---|
| **Actual** | | Negative | Positive |
| | Negative | 59 | 25 |
| | Positive | 88 | 82 |

Table 13 - Evaluation Metrics

| Evaluation Metric | Score |
|---|---|
| **Accuracy** | 0.55511 |
| **Precision** | 0.48235 |
| **Recall** | 0.76635 |
| **F1 score** | 0.59206 |

**K-Nearest Neighbor Model based on number of Confirmed Cases.**

Table 14 - Confusion Matrix

| | | Predicted | |
|---|---|---|---|
| **Actual** | | Negative | Positive |
| | Negative | 62 | 71 |
| | Positive | 47 | 74 |

Table 15 - Evaluation Metrics

| Evaluation Metric | Score |
|---|---|
| **Accuracy** | 0.53543 |
| **Precision** | 0.61157 |
| **Recall** | 0.51034 |
| **F1 score** | 0.55639 |

# Naïve bayes

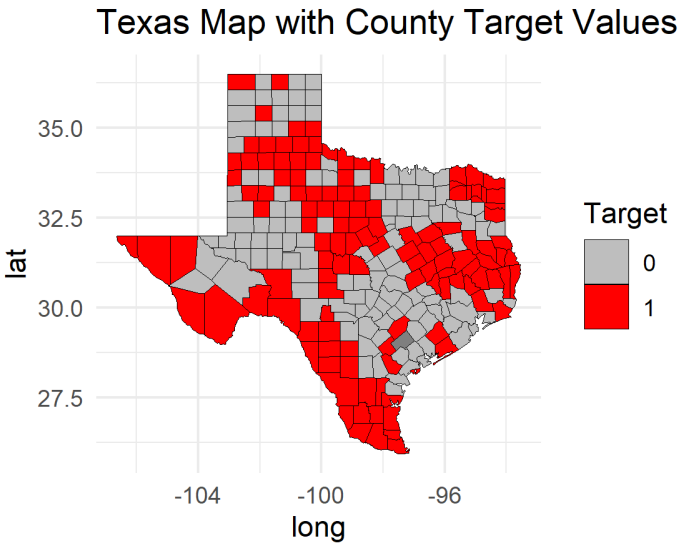


Fig 16 - Naïve Bayes Prediction (Deaths)

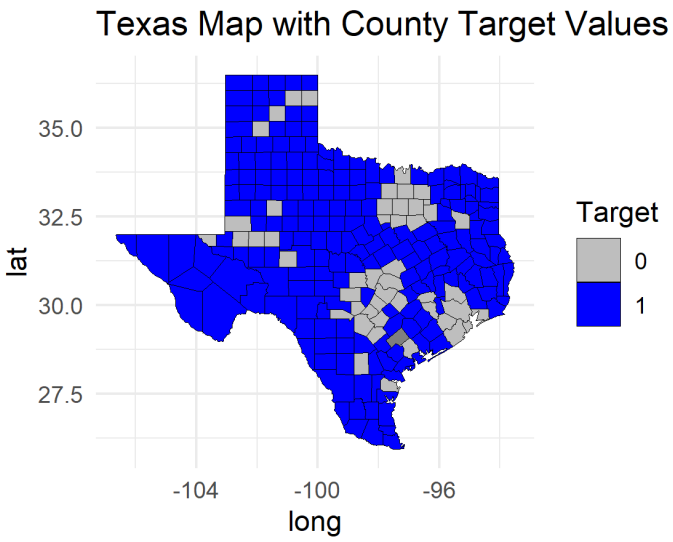


Fig 17 - Naïve Bayes Prediction (Confirmed Cases)

## Model Evaluation

**Naïve Bayes Model based on number of Deaths.**

Table 16 - Confusion Matrix

| | | **Predicted** | |
|---|---|---|---|
| **Actual** | | Negative | Positive |
| | Negative | 62 | 22 |
| | Positive | 65 | 105 |

Table 17 - Evaluation Metrics

| **Evaluation Metric** | **Score** |
|---|---|
| **Accuracy** | 0.65748 |
| **Precision** | 0.61764 |
| **Recall** | 0.82677 |
| **F1 score** | 0.70707 |

**Naïve Bayes Model based on number of Confirmed Cases.**

Table 18 - Confusion Matrix

| | | Predicted | |
|---|---|---|---|
| **Actual** | | Negative | Positive |
| | Negative | 30 | 103 |
| | Positive | 19 | 102 |

Table 19 - Evaluation Metrics

| Evaluation Metric | Score |
|---|---|
| **Accuracy** | 0.51968 |
| **Precision** | 0.84297 |
| **Recall** | 0.49756 |
| **F1 score** | 0.62576 |

# Artificial Neural Network -

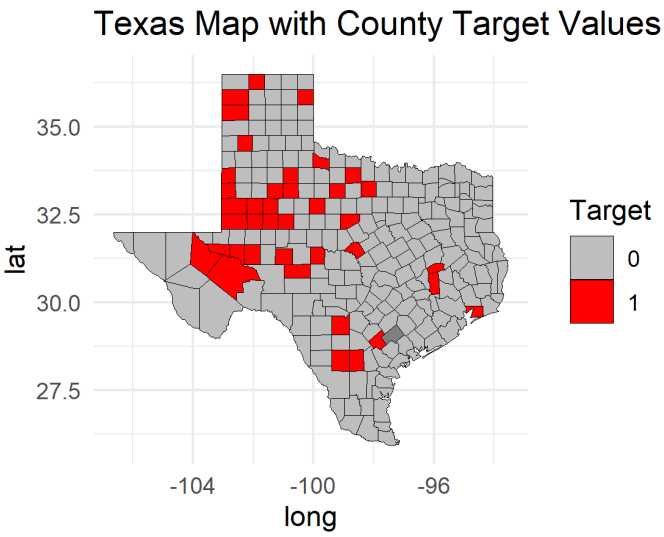


Fig 18 - Neural Network Prediction (Deaths)

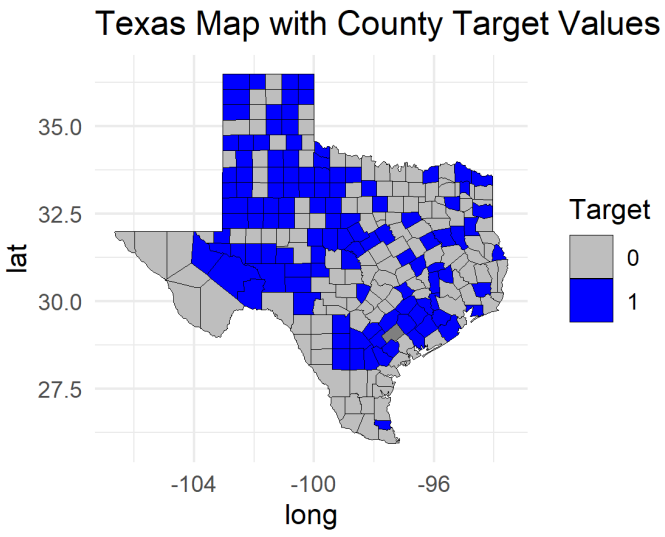


Fig 19 - Naïve Bayes Prediction (Confirmed Cases)

## Model Evaluation

**Artificial Neural Network Model based on number of Deaths.**

Table 20 - Confusion Matrix

| | | Predicted | |
|---|---|---|---|
| **Actual** | | Negative | Positive |
| | Negative | 115 | 18 |
| | Positive | 100 | 21 |

Table 21 - Evaluation Metrics

| Evaluation Metric | Score |
|---|---|
| **Accuracy** | 0.53543 |
| **Precision** | 0.17355 |
| **Recall** | 0.53846 |
| **F1 score** | 0.26250 |

**Artificial Neural Network Model based on number of Confirmed Cases.**

Table 22 - Confusion Matrix

| | | Predicted | |
|---|---|---|---|
| **Actual** | | Negative | Positive |
| | Negative | 75 | 58 |
| | Positive | 63 | 58 |

Table 23 - Evaluation Metrics

| Evaluation Metric | Score |
|---|---|
| **Accuracy** | 0.52362 |
| **Precision** | 0.47933 |
| **Recall** | 0.5 |
| **F1 score** | 0.48945 |

# EVALUATION

## Correlation Matrix

The correlation matrix represents the number of True Negatives, False Positives, False Negatives and True Positives. These values represent the breakdown of the predictions made by the different models that we used. For our task of predicting at risk counties, we want a model that has the least False Negatives and the highest True Positives.

False Negatives are the counties which were at high risk based on the target variables that we used, average number of deaths and number of confirmed covid cases but were classified wrongly as 'safe' counties.

True Positives are the counties which were correctly identified as high-risk counties.

The reason behind the selection of False Negatives and True Positives is that we want to identify counties which are at high risk of being adversely affected by the next wave of Covid-19. So, these variables become essential in selecting the model that performs best out of all the different options that we explored.

Table 24 - FN and TP of Models based on (Deaths)

| | False Negatives | True Positives |
|---|---|---|
| **Random Forest** | 35 | 135 |
| **SVM** | 116 | 54 |
| **k-NN** | 88 | 82 |
| **Naïve Bayes** | 65 | 105 |
| **Artificial Neural Network** | 100 | 21 |

The above table shows the values of False Negatives and True Positives for all the models that we created using the number of deaths greater than national average as the target feature. We see that SVM has the highest False Negatives and right away we come to conclusion that SVM does not give good results at all. If we used SVM for prediction then obviously we are going to get a lot of counties misclassified as 'safe' instead of at high-risk category, to which they belong. Artificial Neural Network also performs horribly. It has high number of False Negatives and the least number of True Positives.

We also see that Random Forest has the highest value of True Positives out of all the models. This tells us that this model correctly identified the high-risk counties. In fact, this model performed the best out of all, it has the least number of False Negatives and highest number of True Positives. At the 2$^{nd}$ spot is k- Nearest Neighbor classification model.

The below table shows the values of False Negatives and True Positives for all the models that we created using the number of confirmed cases greater than national average as the target feature.

Table 25 - FN and TP of Models based on (Confirmed Cases)

| | False Negatives | True Positives |
|---|---|---|
| **Random Forest** | 115 | 6 |
| **SVM** | 80 | 41 |
| **k-NN** | 47 | 74 |
| **Naïve Bayes** | 19 | 102 |
| **Artificial Neural Network** | 63 | 58 |

In the case where target value was the deciding factor for classification, we see the worst performing models are Random Forest and SVM. They have the highest False Negatives and the lowest True Positives.

The Naïve Bayes classification model works the best for identifying the counties where we can expect a high number of covid cases. It has the lowest False Negatives and highest True Positives.

## Accuracy, Precision, Recall & F1 Score -

Accuracy of a model is the percentage of instances the model correctly labelled out of all the instances. A high value for this metric is desirable for a model.

Precision of a model is the percentage of instances correctly predicted as True Positive out of all the Positive predictions. It is used to identify the accuracy of positive predictions. A high value of Precision means that the model is good at correctly predicting True Positives or there is high confidence in the accuracy of the model when they classify a sample as positive. We want this value high for our model.

Recall or Sensitivity is the percentage of instance correctly predicted as True Positive out of all the instances. A high value of Sensitivity means that the model is correctly predicting True Positives out of all the samples and is not missing many True Positives and wrongly classifying them as ‘safe’. We want this value to be high as well, because we want the model to have a low number of False Negative predictions and a high Sensitivity means low number of False Negatives and high number of True Positives.

F1 Score is the Harmonic Mean of Precision and Sensitivity. This metric is used to analyze the balance between Precision and Sensitivity. A high value of F1 score means that the model performs well.

Table 26 - Model Performance Comparison (Deaths)

| | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| **Random Forest** | 0.71653 | 0.79411 | 0.78488 | 0.78947 |
| **SVM** | 0.50787 | 0.31764 | 0.85714 | 0.46351 |
| **k-NN** | 0.55511 | 0.48235 | 0.76635 | 0.59206 |
| **Naïve Bayes** | 0.65748 | 0.61764 | 0.82677 | 0.70707 |
| **Artificial Neural Network** | 0.53543 | 0.17355 | 0.53846 | 0.26250 |

When we analyze the above table, table 26, for the models where target variable was based on number of deaths, we see that Accuracy, Precision and F1 score values are highest for Random Forest model. The high value of Accuracy is obviously desirable, apart from that, Precision and Sensitivity are also important. We want to be confident in the model's ability to identify True Positives with high accuracy, which the Precision tells us. The Sensitivity of Random Forest is not the highest, but it is still indicating good level of accuracy prediction of True Positives out of all the predictions made by the model. The F1 score shows a good balance between the two values.

At the second spot we see that Naïve Bayes model also performs moderately well. The overall accuracy and Precision is lower compared to the Random Forest Model. But a high value of Sensitivity means that this model makes accurate predictions of Positives, and there is less chance of chances of a False Negative. The F1 score also shows a good balance of Precision and Sensitivity.

The below table, Table 27, shows the model performance metrics for the models where target variable was based on number of confirmed cases.

We see that all the models had a low overall accuracy when we tried predicting based on number of confirmed cases. Random Forest performs the worst, it has low accuracy and Precision. F1 score for Random Forest is the lowest. This model should be avoided for predicting the counties where there will be higher than average number of covid cases.

When we look at Naïve Bayes Model metrics, we see that it has low accuracy on test data. It has a good Precision value, which means the model is good at predicting Positives. But it has a low value of Sensitivity, which means the model also misclassifies True Positives as False Negatives. The F1 score is better as compared to other models. This model can be used for prediction with a lot of contingencies. Only positive classifications from this model are useful, any negative classification of county using this model will be counterproductive. As there is high confidence in the model's ability to correctly identify Positive instances or high-risk counties. But if the models says the county belongs to low risk category, then we don't have much confidence in this prediction.

Table 27 - Model Performance Comparison (Confirmed Cases)

| | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| **Random Forest** | 0.53543 | 0.49686 | 0.6667 | 0.09230 |
| **SVM** | 0.52362 | 0.33884 | 0.5 | 0.40394 |
| **k-NN** | 0.53543 | 0.61157 | 0.51034 | 0.55639 |
| **Naïve Bayes** | 0.51968 | 0.84297 | 0.49756 | 0.62576 |
| **Artificial Neural Network** | 0.52362 | 0.47933 | 0.5 | 0.48945 |