# **Python/DL Project Report**

**Team 10**:

1. Sai Rohith Guntupally (8)
2. Chaitanya Mallepudi (14)
3. Madhu Varma Rudraraju (20)

**Project title**:

**Accident Prediction** (A country wide traffic accident dataset in united states.)

**Goals and Objectives:**

Since road accidents is one of the major concerns in the United States, we would like to analyze and discover what are the causes for accidents such as the impact of precipitation or other environmental factors.

The main objective is to utilize the analysis and predict the probability of road accidents.

**Dataset Description:**

This is a countrywide car accident dataset, which covers **49 states of the United States**. The accident data are collected from **February 2016 to December 2019**, using several data providers, including two APIs that provide streaming traffic incident data.

These APIs broadcast traffic data captured by a variety of entities, such as the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road-networks. Currently, there are about **3.0 million** accident records in this dataset.

**Approach**:

We are adding a column named accident probability based on severity of accident.

When the severity is very low, probability is 0; else it is 1.

We pre-processed the dataset and removed the unnecessary features..

We calculated the accuracy of the model using Naïve-Bayes classification.

We implemented Deep Learning techniques, and to get accurate results out of it we have chosen this approach.

**Increment 1**:

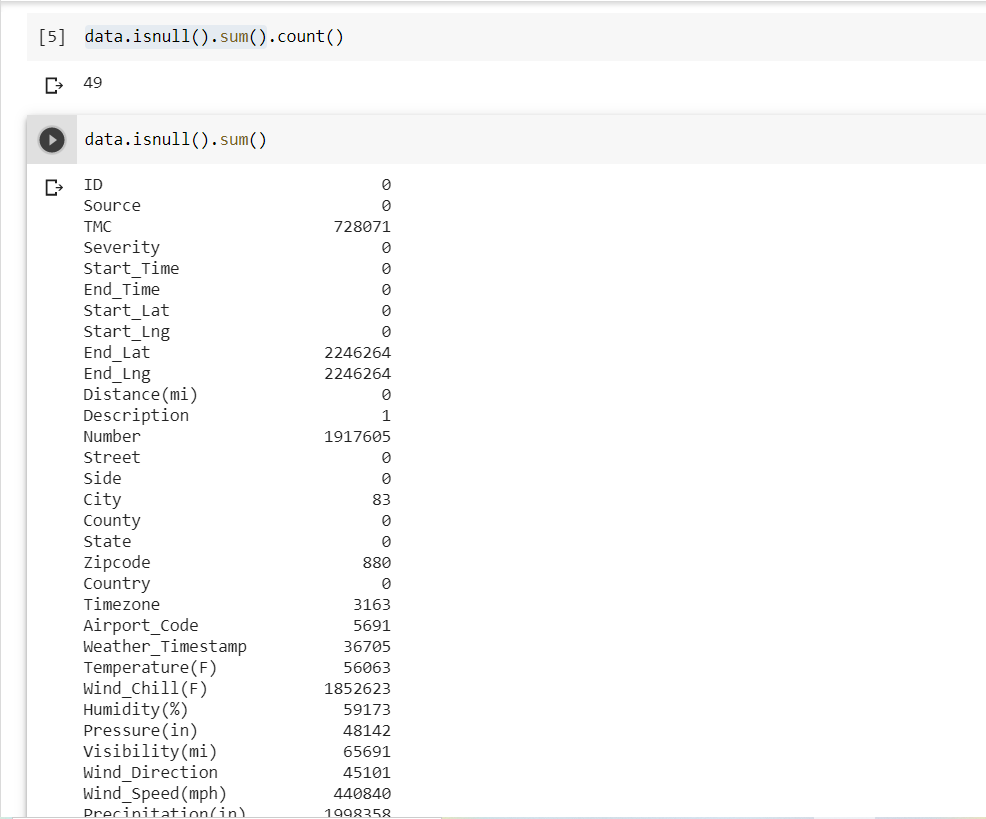
Loaded the dataset and familiarized the records in the dataset.



Performed the EDA by finding out the number of null values in the columns and fill them with the mean values.

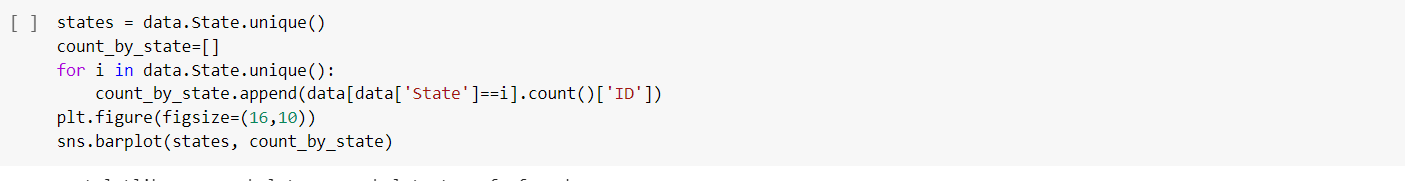


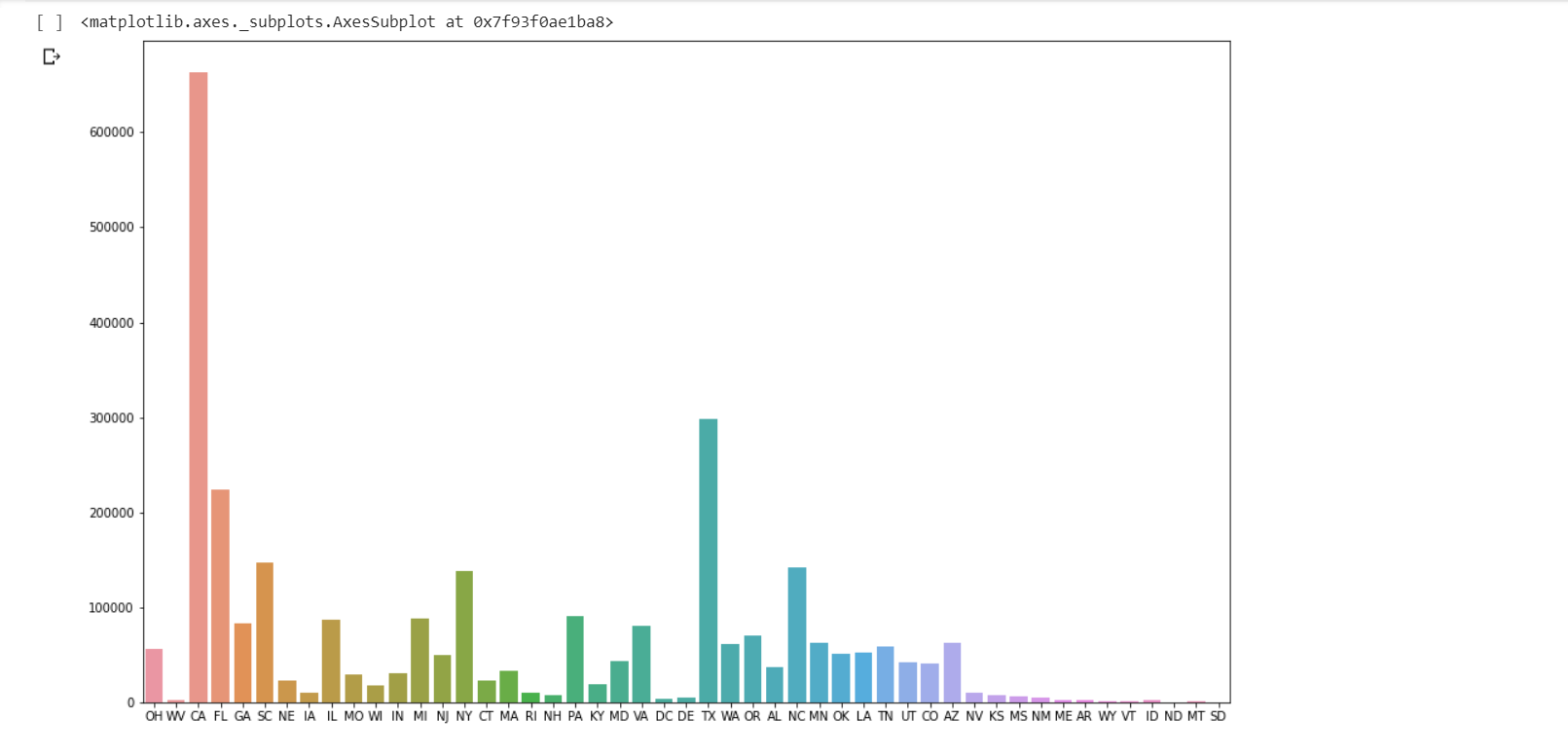
After performing the EDA; we can see there are no null values in the respective columns.



With the available data, we have visualized the following:

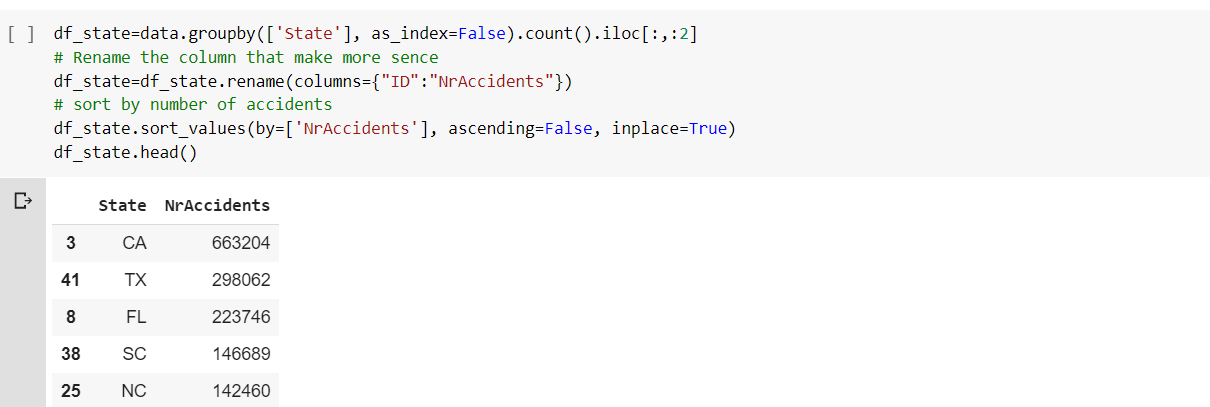
1. State wise accidents count



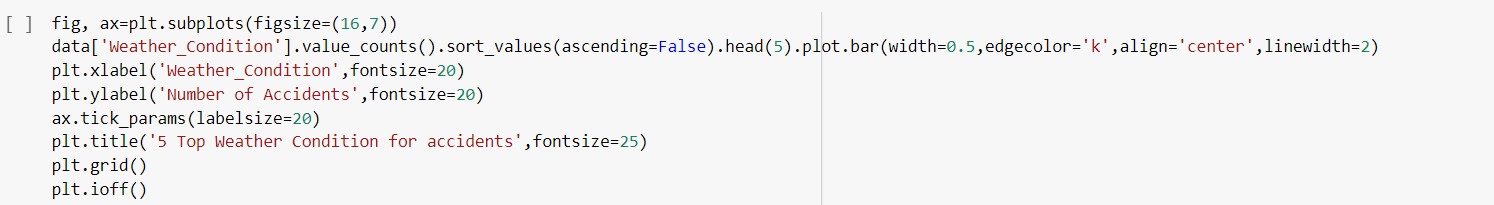


From the above we cannot infer everything clearly.

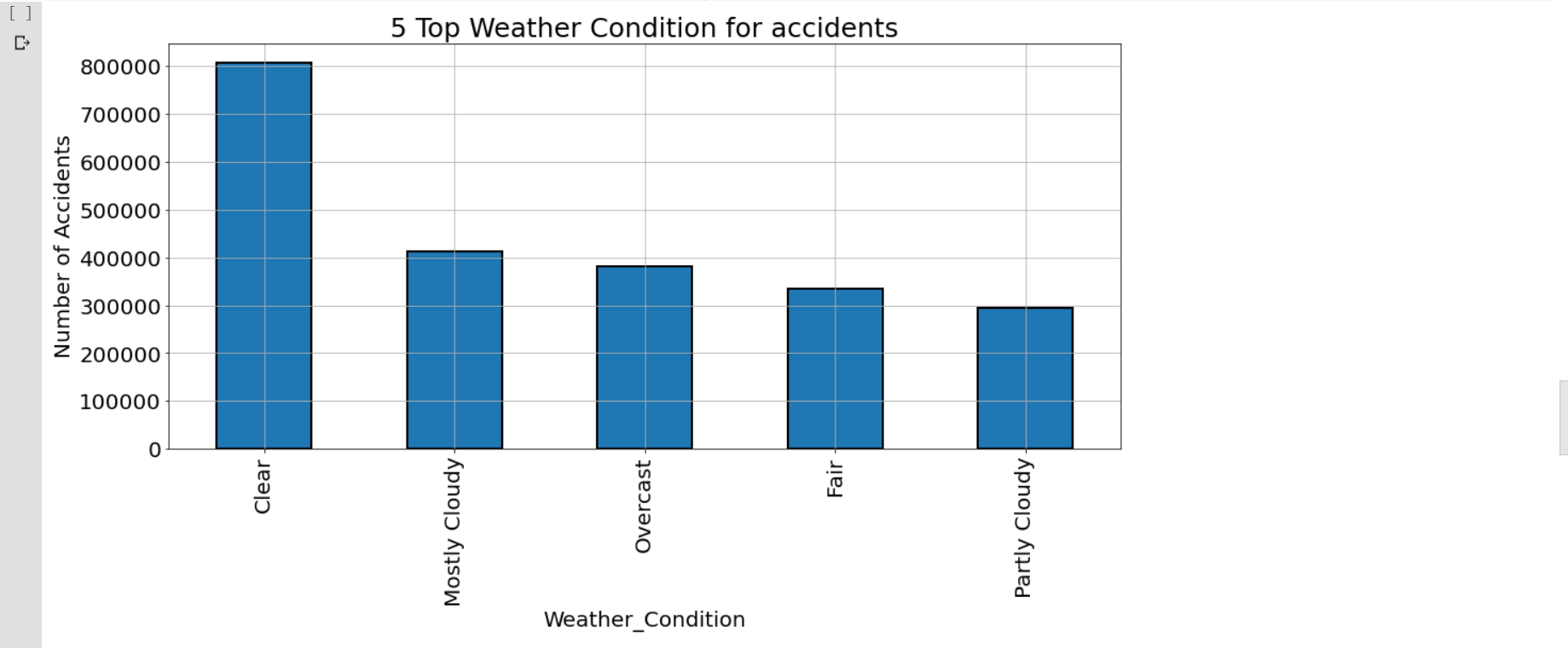
So, we have extracted the top 5 cities with most number of accidents.



The next thing we done is to find the what kind of weather conditions are influencing the accidents.

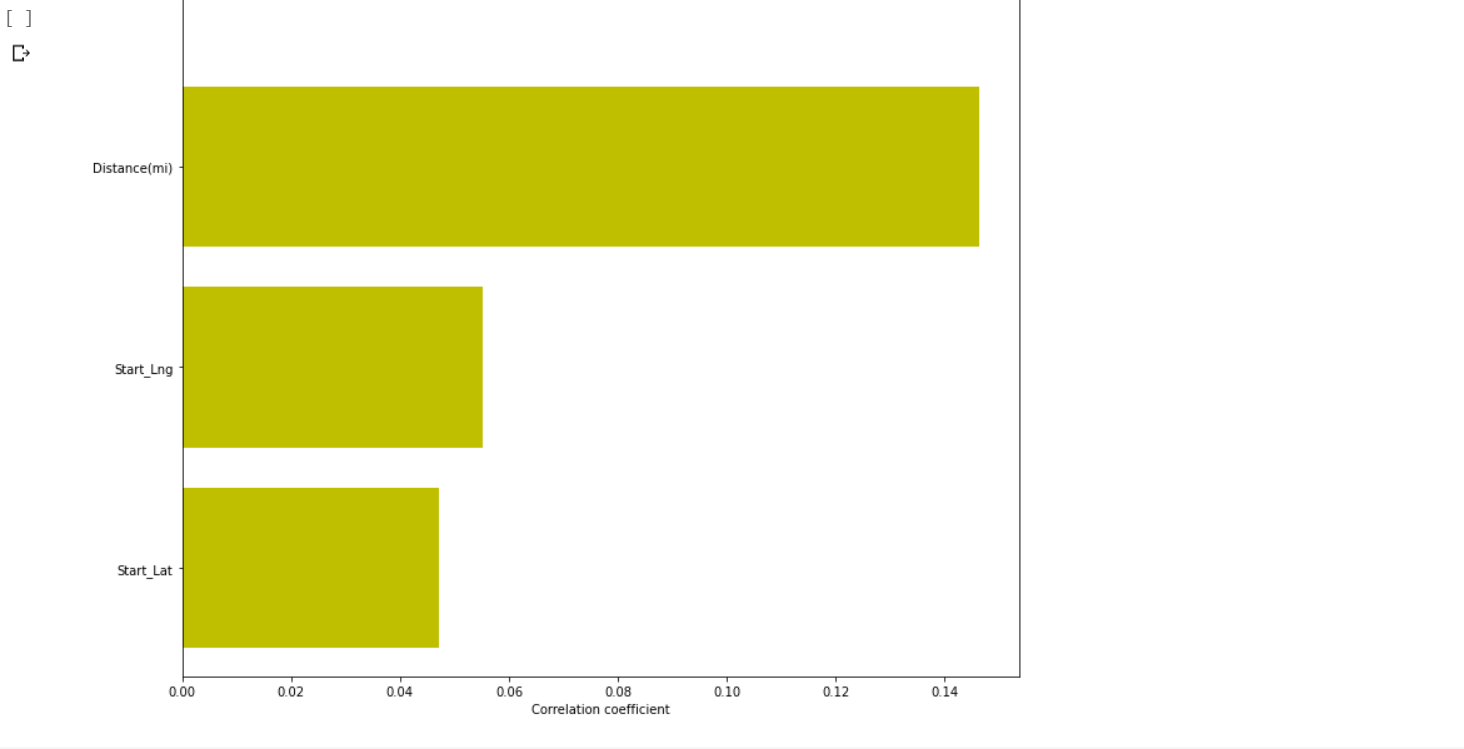


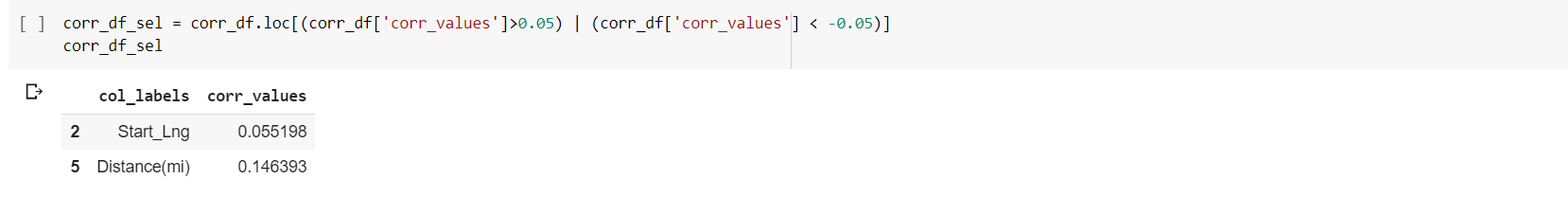
We have observed the following:



Next we have found the correlation between the features and accident cause; if the correlation coefficient is high, it means that it is the most affecting factor for the cause of accident.



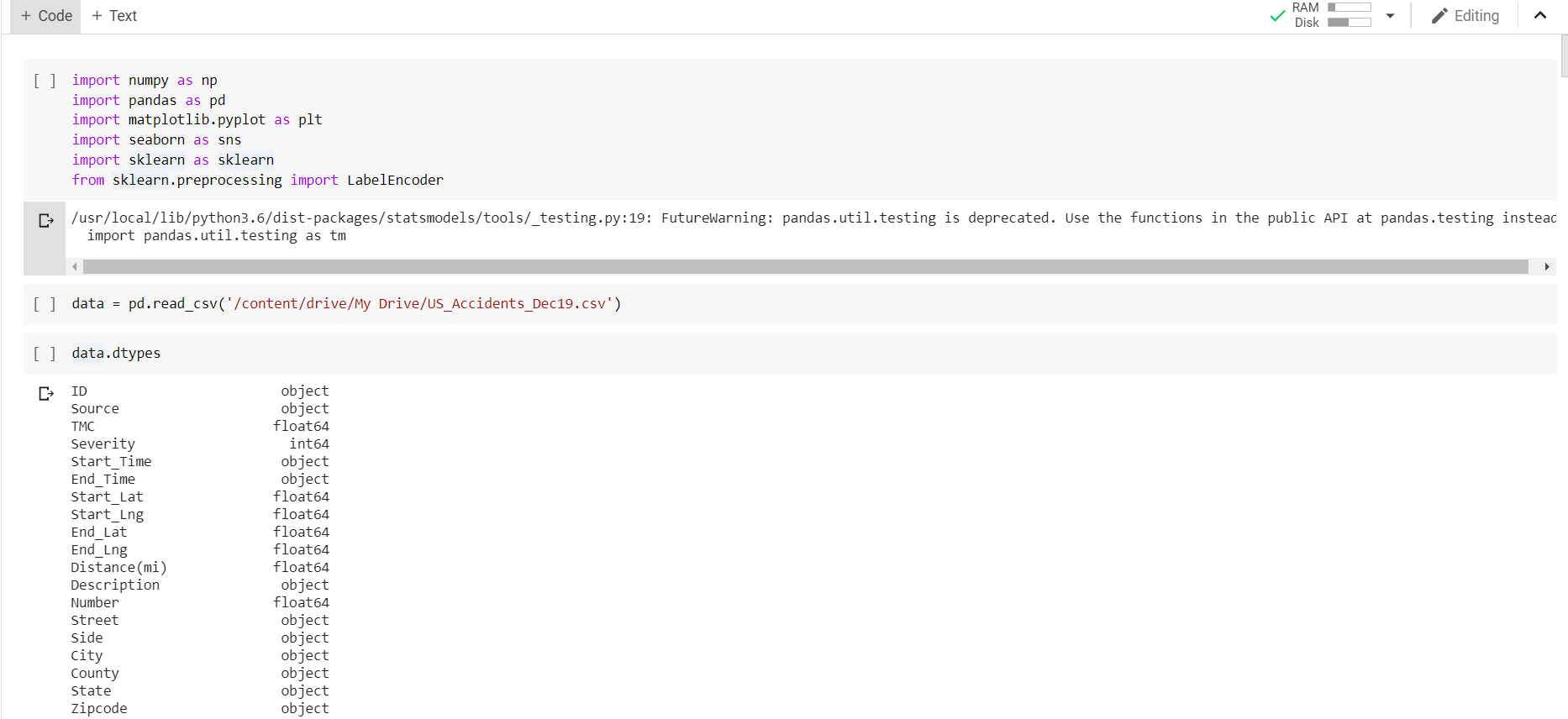




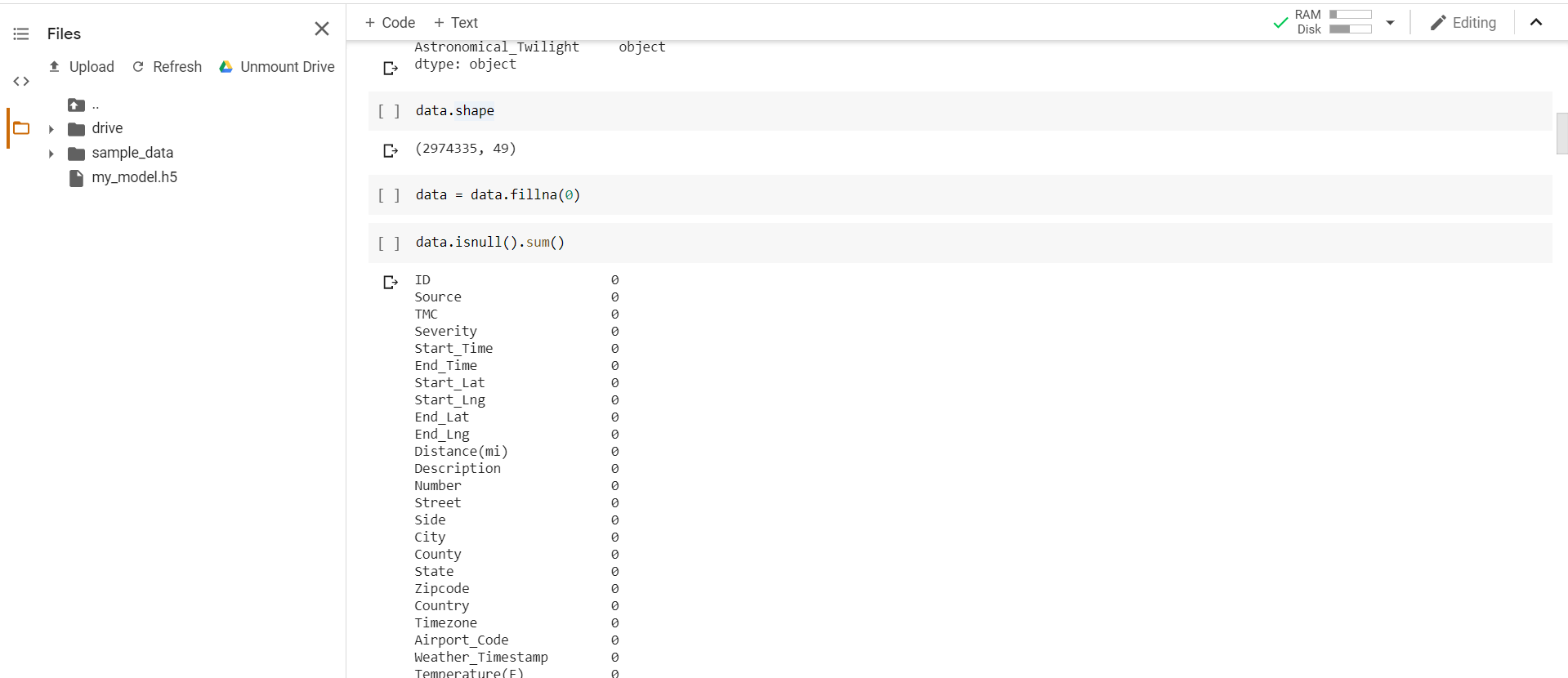
**Increment 2**:

We have taken the accidents data set from kaggle , Done pre-processing, predict the accident probability using regression, done binomial classification, passed the model to neural network to improve the accuracy of the model and then build the model using keras and saved it, then predicted the probability using the model.

Import the accident dataset and see the dtypes present in the dataset.



Then as a part of pre-processing remove the null values in the dataset.

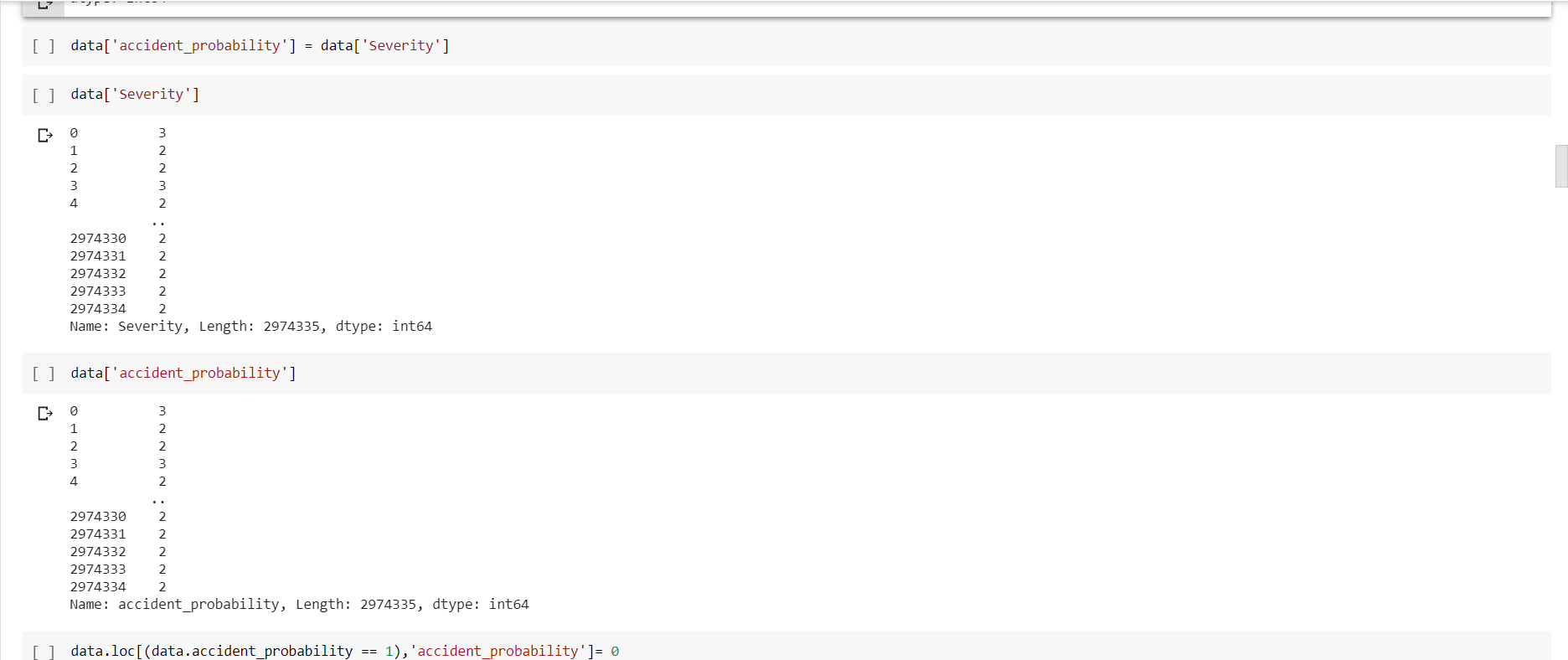


Now, let us include the accident probability variable to the dataset.

The main purpose of this variable is to predict the accident probability based on the given inputs,

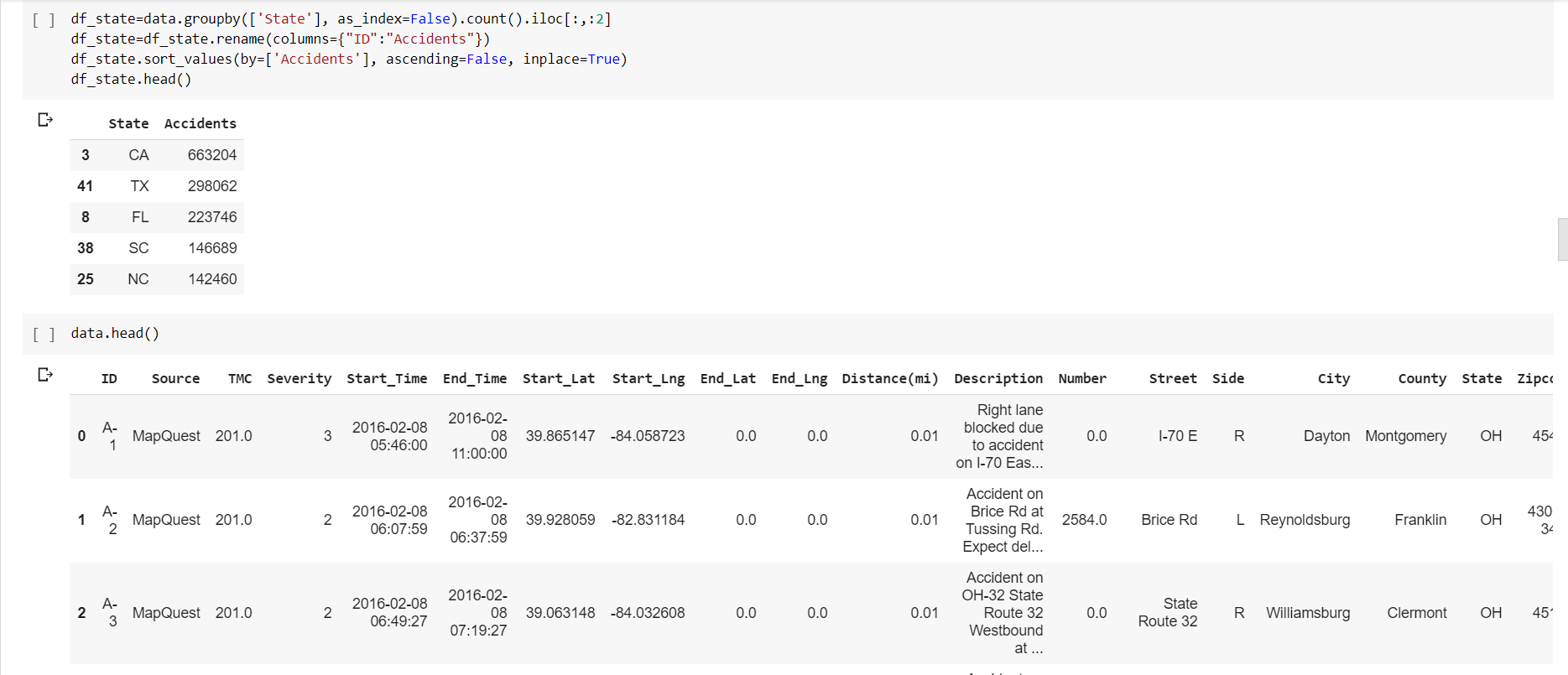
We have assigned this variable value by taking it as dependent variable on Severity.

If the severity value is very low (1) the accident probability is 0 else 1.

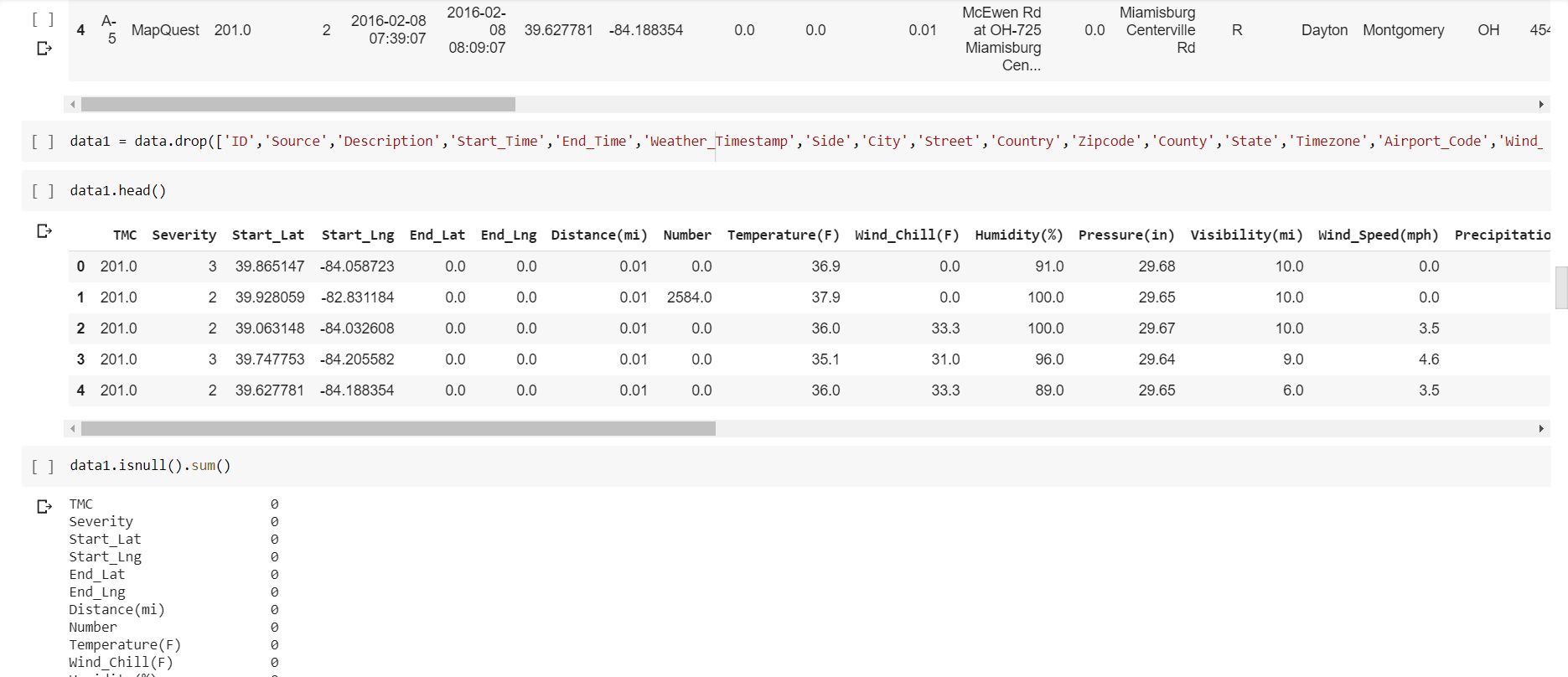




Below are the top 5 states which has most number of accidents.



We are dropping, many of the features which are of no correlation from which we need to predict.



We had mapped the Boolean values or binomial values to 0 or 1.



We had classified the model using Naïve-Bayes classification model since we need to assume the features are to be independent.



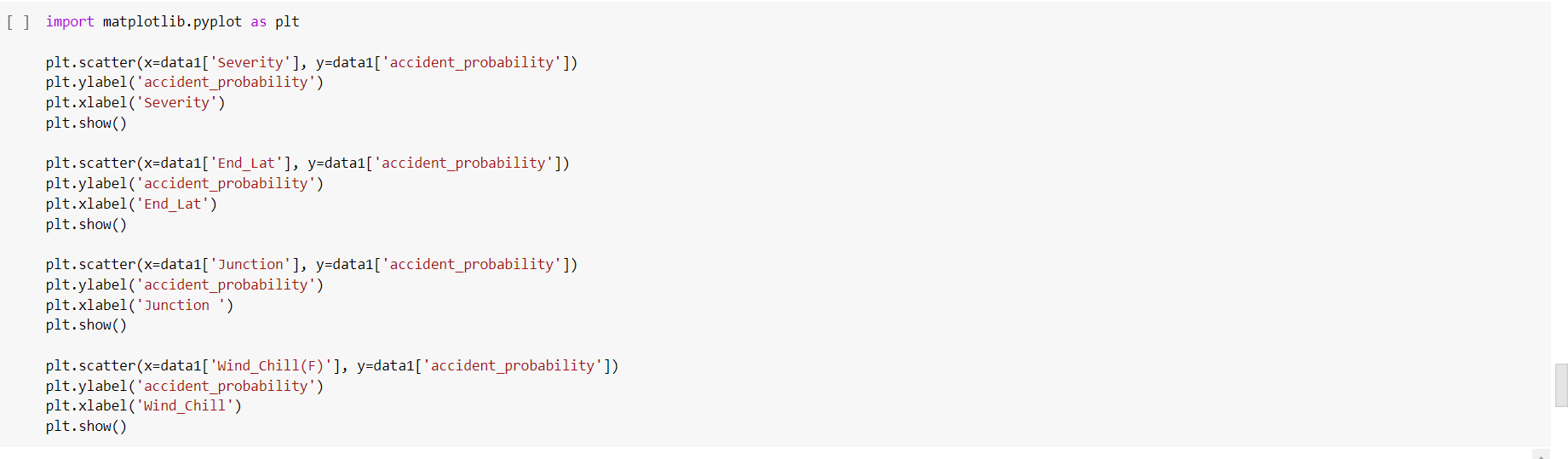
We got the accuracy of 60.72%. Is this enough? Surely, we need to increase the accuracy of the model.

We have taken the top 5 correlated variables that are effecting the accident probability.



We can infer that Severity, End\_Lat, Junction, Wind\_Chill, Distance are those features.

We had plotted those factors and represented graphically how they are related using matplotlib.

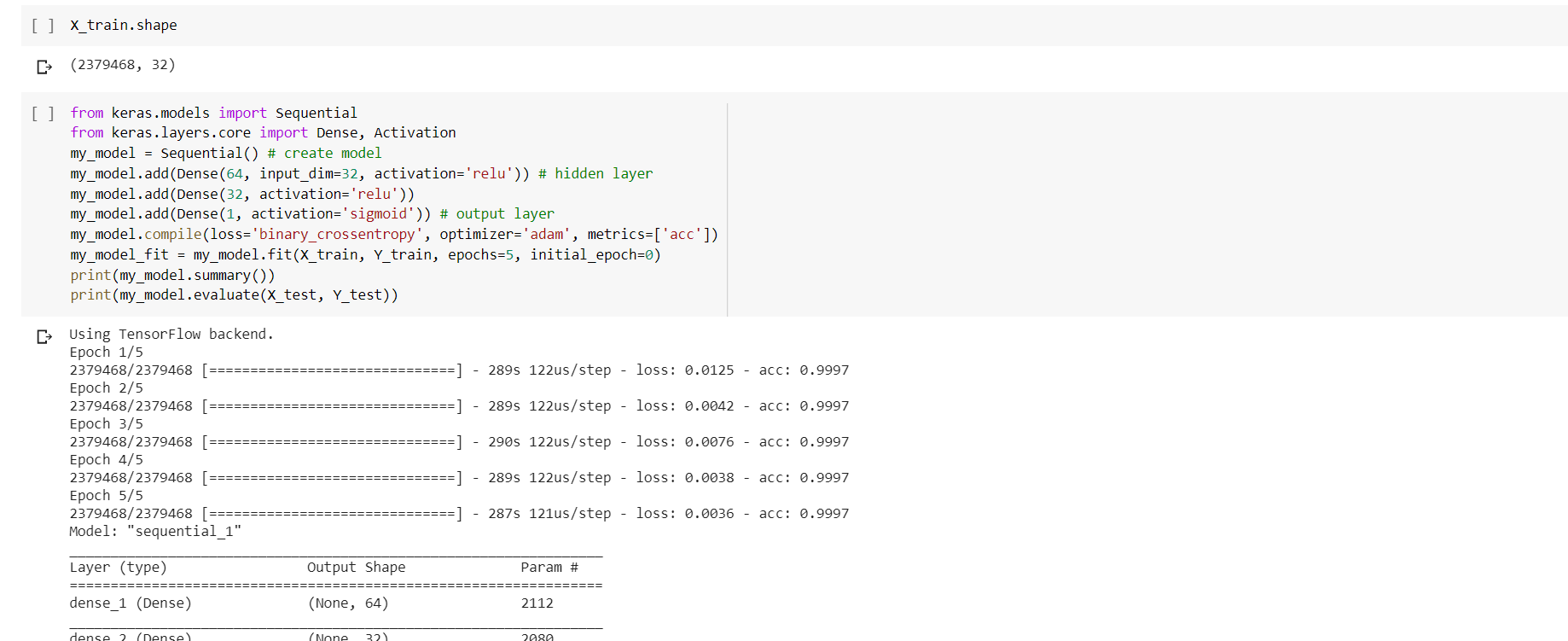


Below are those visualizations.





Now the next thing we need to do is to increase the accuracy of the model. So, let us pass the model to neural network using sequential model.

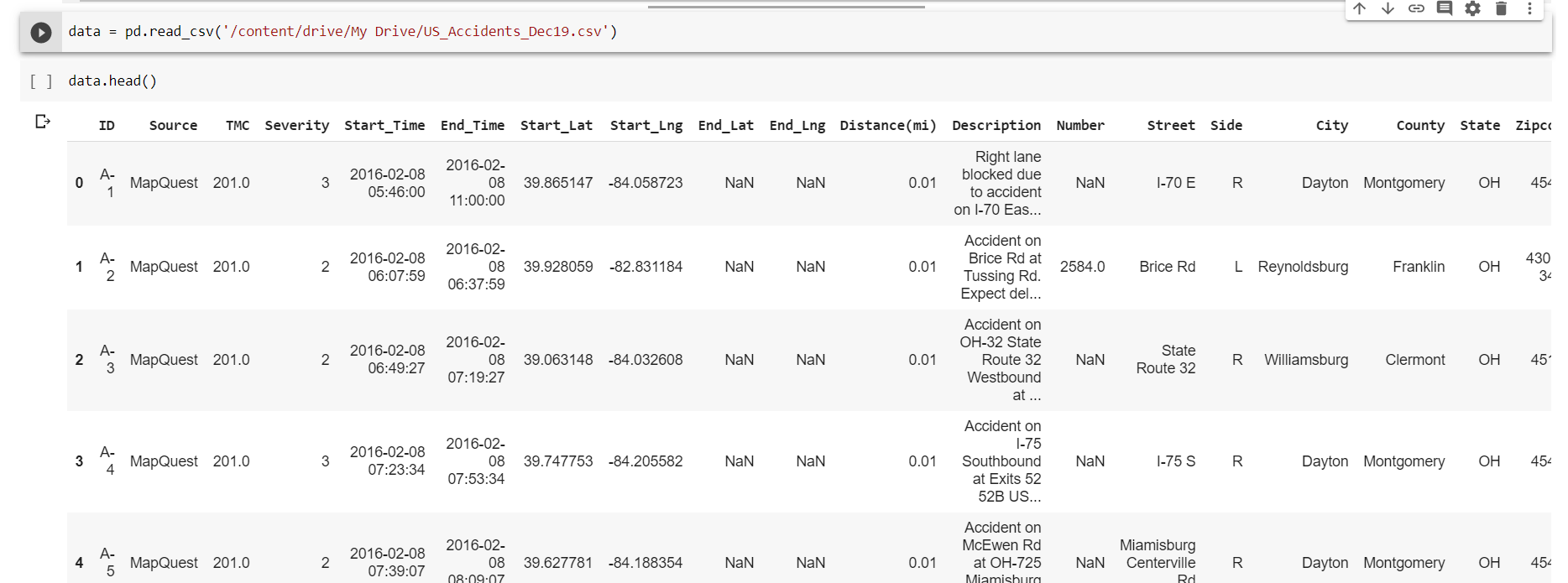


We can see the accuracy increased to 99%.

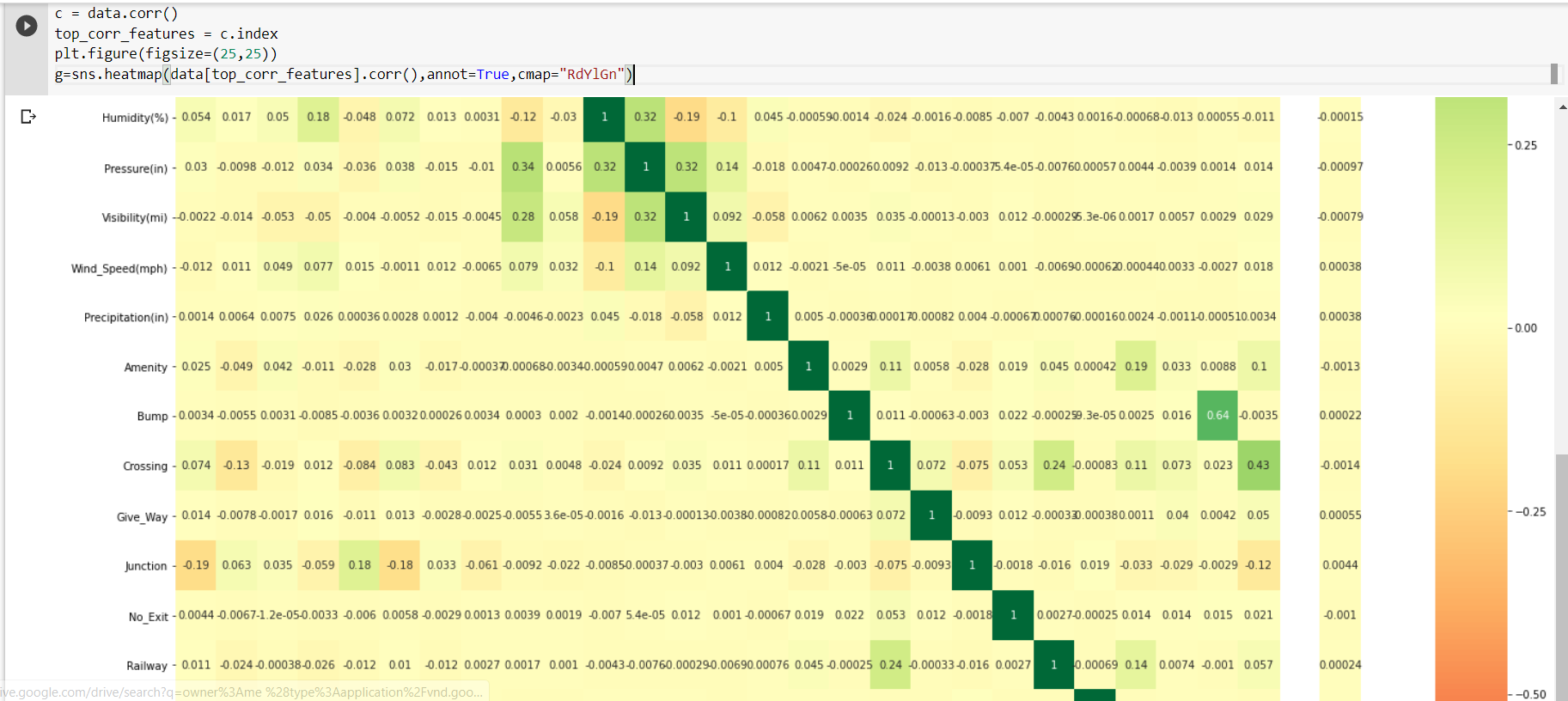
**Increment 3**:

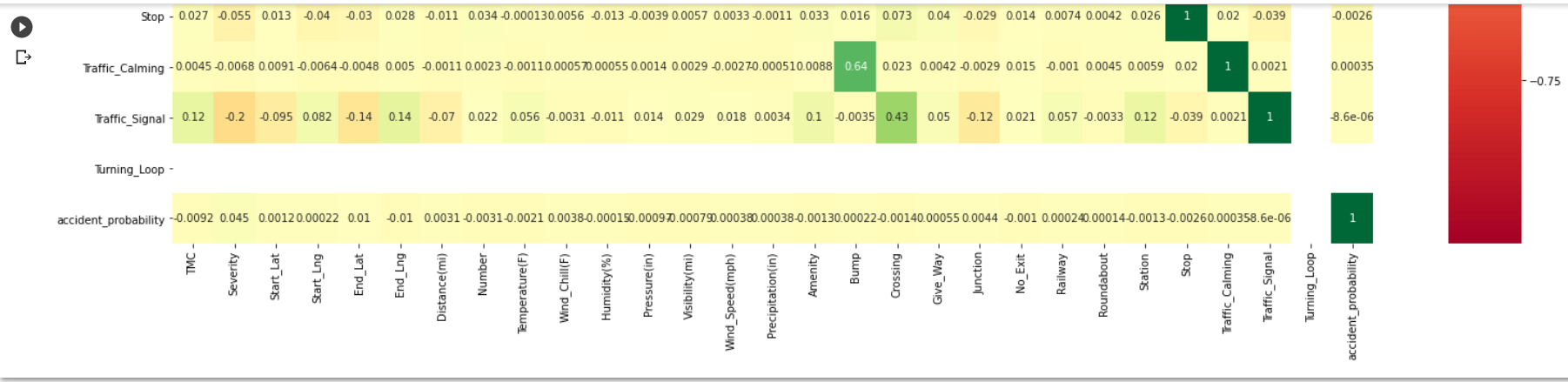
Since the accuracy is 99%, we need to see whether the train data is overfitted or not.

So, we are using some ensembling and boosting techniques to make sure the variance is low despite of complexity of model and the coefficients of correlated and non-correlated features are evenly distributed.

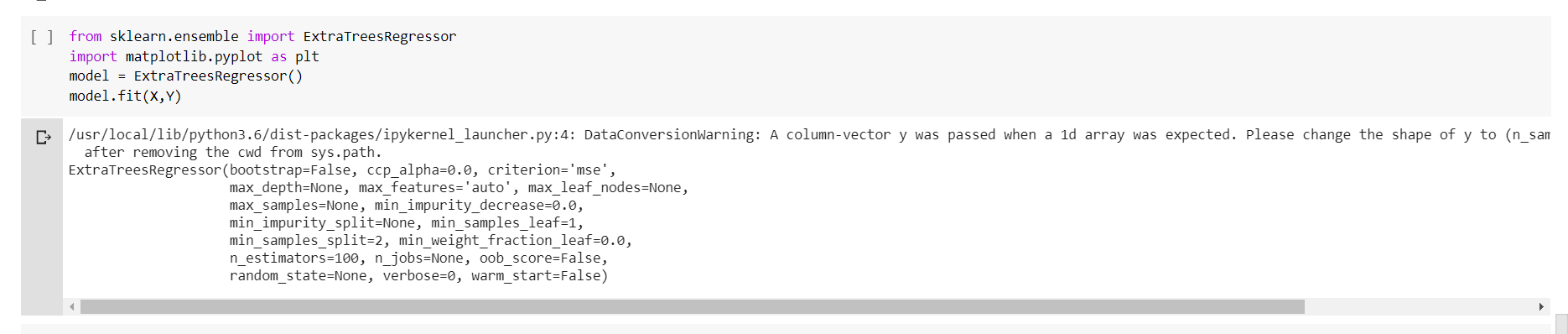


With the help of correlation matrix, do the pre-processing for better performance of the model.

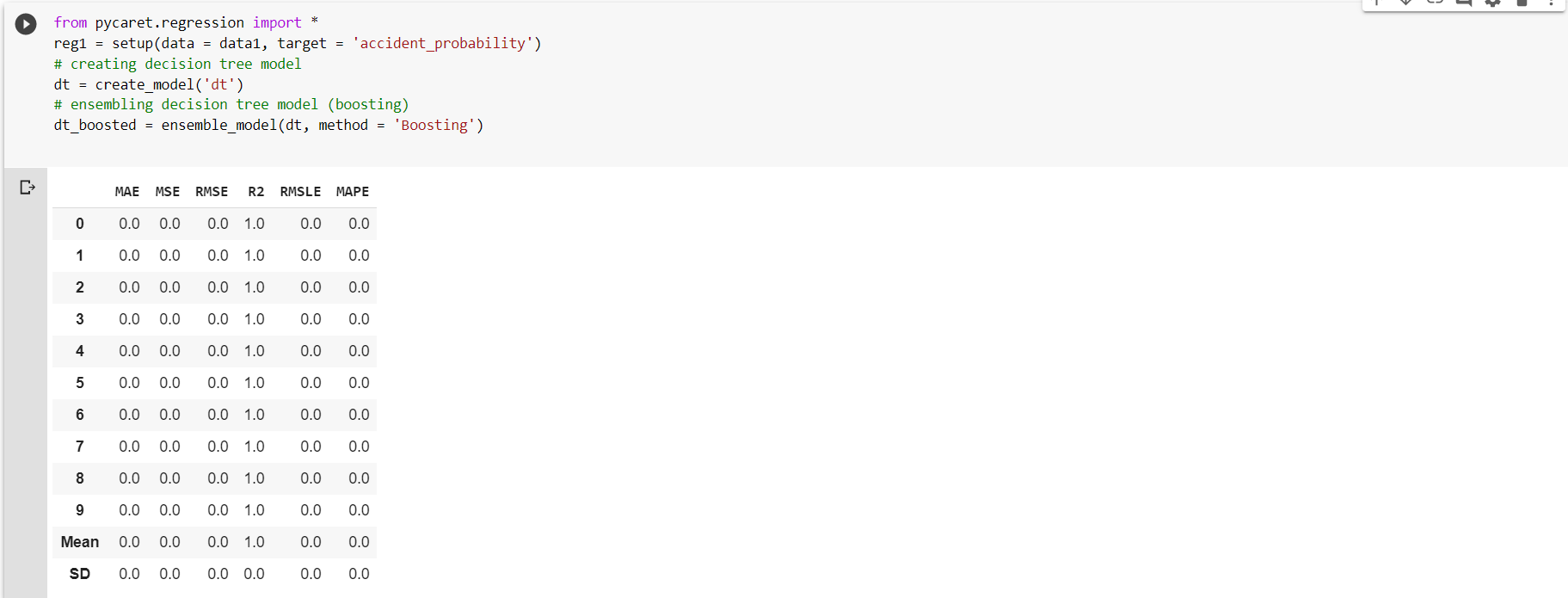


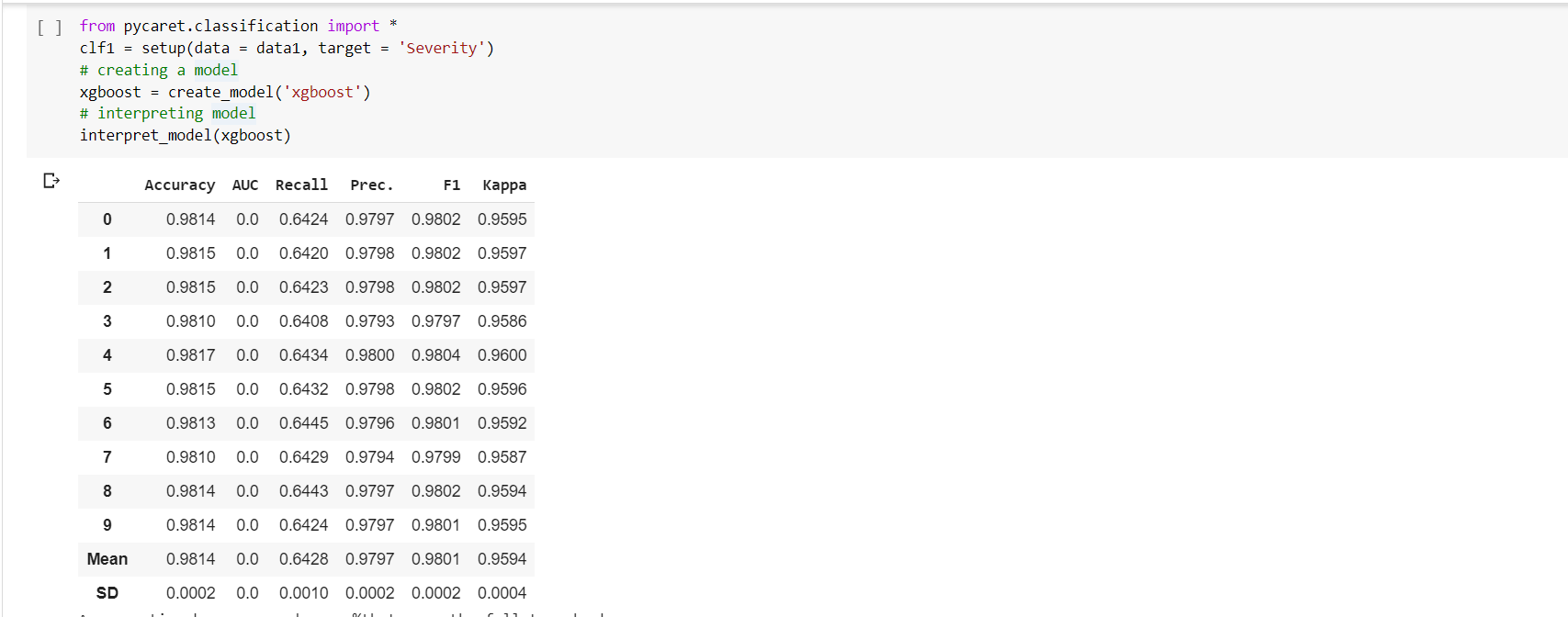


Fit the model using Extra Tree Regressor to have the low variance for the high complex model and better performance of the model. We can also use random forest, but extra tree regressor have the simple calculations.

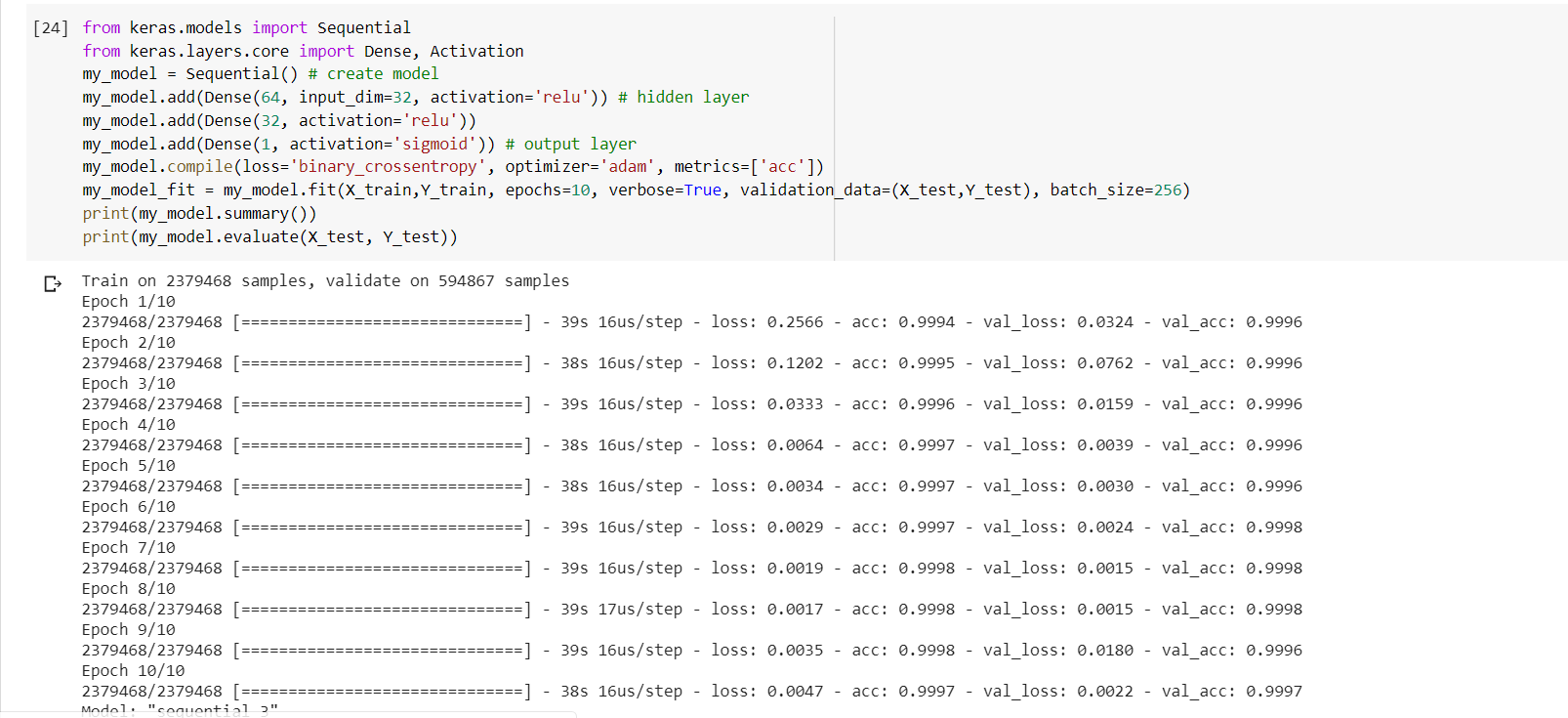


We have tried to boost the model using the boosting techniques using the pycaret which is a open source module and improved the mean and SD using predefined automated functions and adjusts the weights of the related and non correlated features.



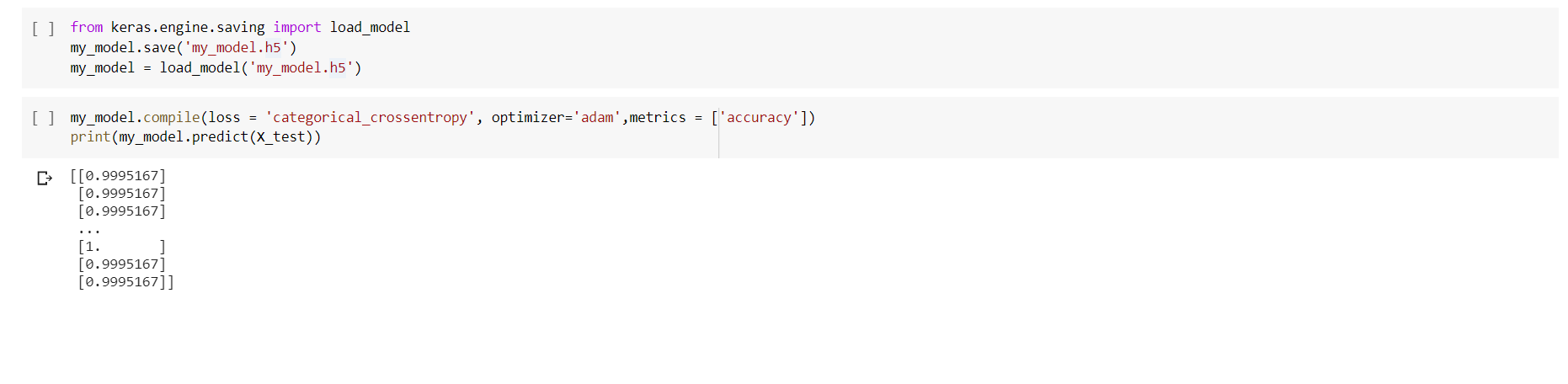


Then we had built the sequential model using the keras library and predicted the output and saved the model.



Since we got very high train and validation accuracy. Our deep learning model which predicts accuracy is ready. Now we can save the model and test the model with test data.





Here we got the accuracy of 99.97% which gives very high accurate output.

**Conclusion**:

Hence, we have developed a Deep learning model which predicts the probability of accident with very high accuracy and made sure that the result is not from the overfitted model.

**Languages and Libraries:**

* Python
* Matplotib
* Ski-kit learn
* Numpy
* Pandas.
* Keras

**Software and platforms:**

* Jupyter Notebooks
* Google colab.
* Pycharm Professional.

**Github Link**:

<https://github.com/MRChaitanya/Python-DeepLearning/wiki/Final_Report>

**Increment -1**

<https://github.com/MRChaitanya/Python-DeepLearning/wiki/Project-Increment--1>

**Increment -2**

<https://github.com/MRChaitanya/Python-DeepLearning/wiki/Project-Increment--2>

**References**:

<https://www.kaggle.com/us-accidents>

<https://pycaret.org/>

<https://stackoverflow.com/>

<https://www.tensorflow.org/api_docs/python/tf/keras/Model>