Car Insurance Claim Algorithm: Would your Insurance Cost More This Year?

Roshan Ramamoorthi

Sheridan College

PROG 25211: AI and Machine Learning - 1261 26995

Simon Hood

January 31, 2026

Table of Contents

Dataset

Title: Car Insurance Data

Author: Sagnik Roy

Columns: (19 Columns) ID, Age, Gender, Race, Driving Experience, Education, Income, Credit Score, Vehicle Ownership, Vehicle Year, Married, Children, Postal Code, Annual Mileage, Vehicle Type, Speeding Violations, DUIs, Past Accidents

Rows: 10,000 originally and 8149 after data cleanup

Description:

This dataset is the annual data from a car insurance company based in the US. The data originates from 4 main locations, New York, Florida, California, and Maryland. The data incorporates several customer attributes and focuses on whether they had submitted a claim. Overall, there is a good balance between the driver attributes, car and the driver's previous record. I found this data interesting as the census of this data will likely be used to dictate their prices for the following year. There were some empty values that needed to be cleaned up, and the ID column was dropped completely. The author claimed the data is real with some generated values.

Source: (Kaggle) https://www.kaggle.com/datasets/sagnik1511/car-insurance-data

Question

I found the data to be interesting since it resembles a realistic scenario of an insurance company reviewing data to make an informed decision. It would make sense to review which drivers, and which cars accounted for the most claims. The company could

then adjust pricing based on risk, allowing for safer drivers to benefit from cheaper rates and risker drivers to pay more. **Based on the previous year of insurance data, would my insurance go up or down?** Even if I didn't make a claim this year, my personal factors and car could put me on the hook for a higher premium based on the claims of others who are like me.

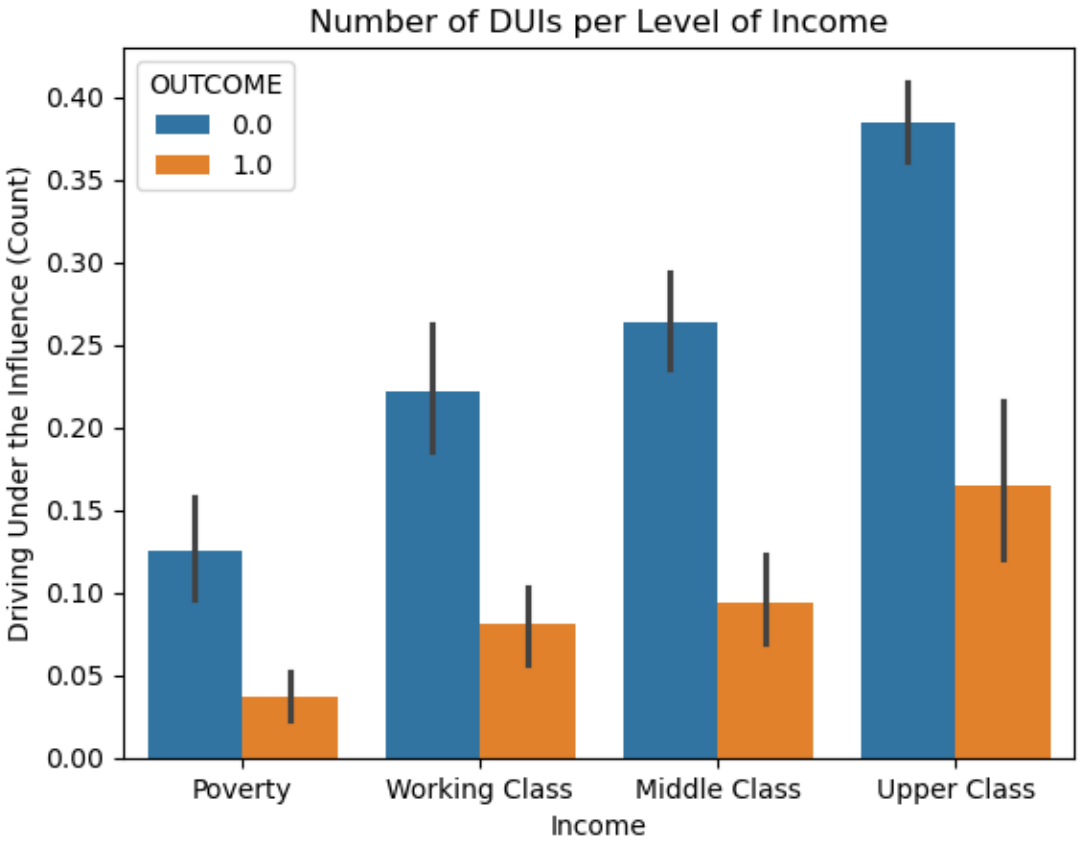
Cleaning the Data

This dataset used a lot of strings and ranges for its data points. Thus, all string data points were converted to integers starting from 0. The ID column was removed in the beginning since it was irrelevant to the algorithm. Unique IDs would only be useful to keep track of sperate rows of data. After the data was converted and the unnecessary columns were removed, any rows with empty data were also removed. The credit score and annual mileage columns had nearly 1000 missing data points. The result after data clean-up was 8149 rows of all numeric and complete data.

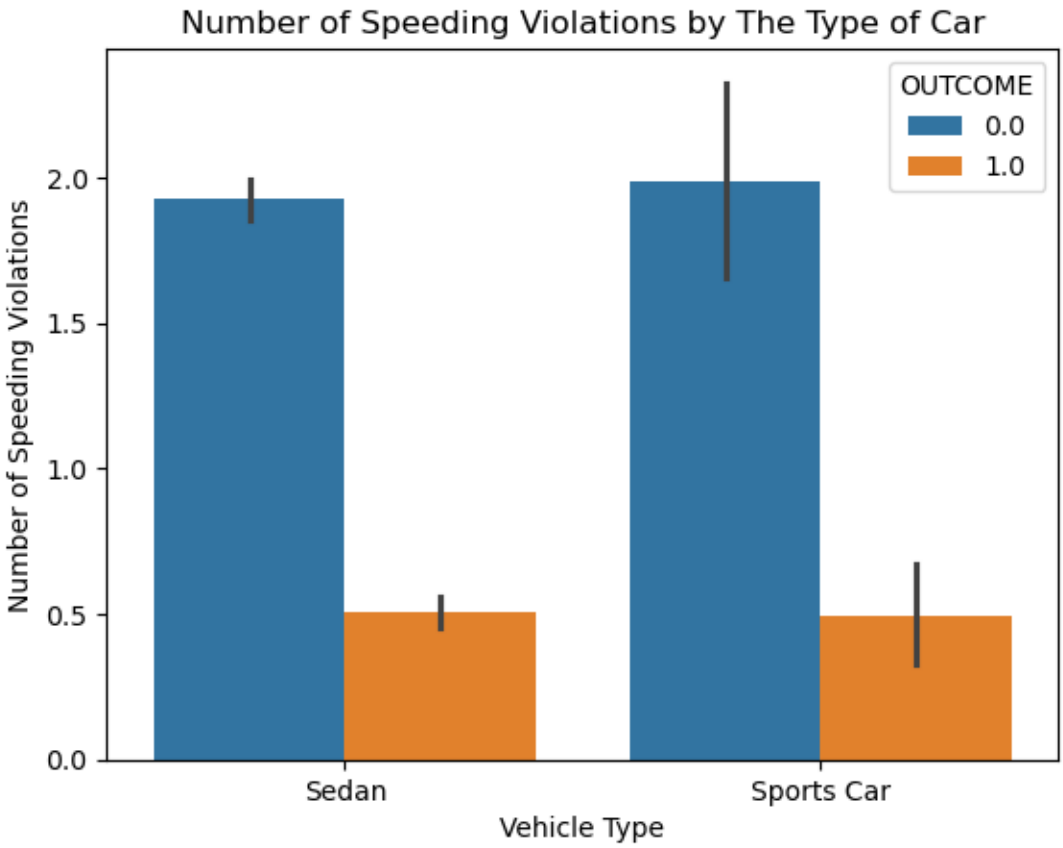
Column Title	Value Description
Age	16-25 Years = 0 26-39 Years = 1 40-64 Years = 2 65+ Year = 3
Gender	Male = 0 Female = 1
Race	Majority = 0 Minority = 1
Driving Experience	0-9 Years = 0 10-19 Years = 1 20-29 Years = 2 30 Years or More = 3
Education	None = 0 High School = 1 University = 2
Income	Poverty = 0 Working Class = 1

	Middle Class = 2 Upper Class = 3
Credit Score	Credit Score / 1000
Vehicle Ownership	Owned (Paid Off) = 0 Not Owned (Finance or Lease) = 1
Vehicle Year	Before 2015 = 0 After 2015 = 1
Married	Single = 0 Married = 1
Children	No Children = 0 One or More Children = 1
Postal Code	10238 (New York) = 0 32765 (Florida) = 1 92101 (California) = 2 21217 (Maryland) = 3
Annual Mileage	True Value in Miles
Vehicle Type	Sedan = 0 Sports Car = 1
Speeding Violations	Number of Violations
Driving Under the Influence (DUIs)	Number of Violations
Past Accidents	Number of Accidents
Outcome	No Claim (Cheaper/Same Insurance Price) = 0 Claim (More Expensive Insurance) = 1

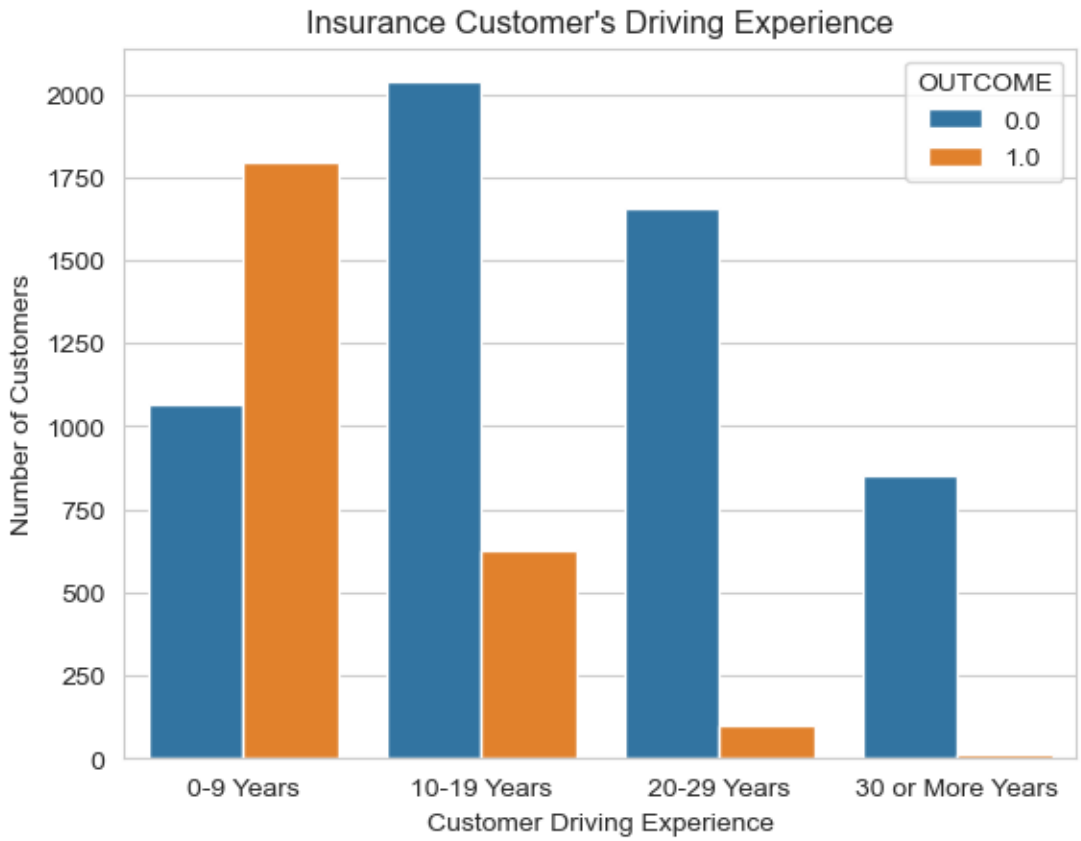
Visualizations



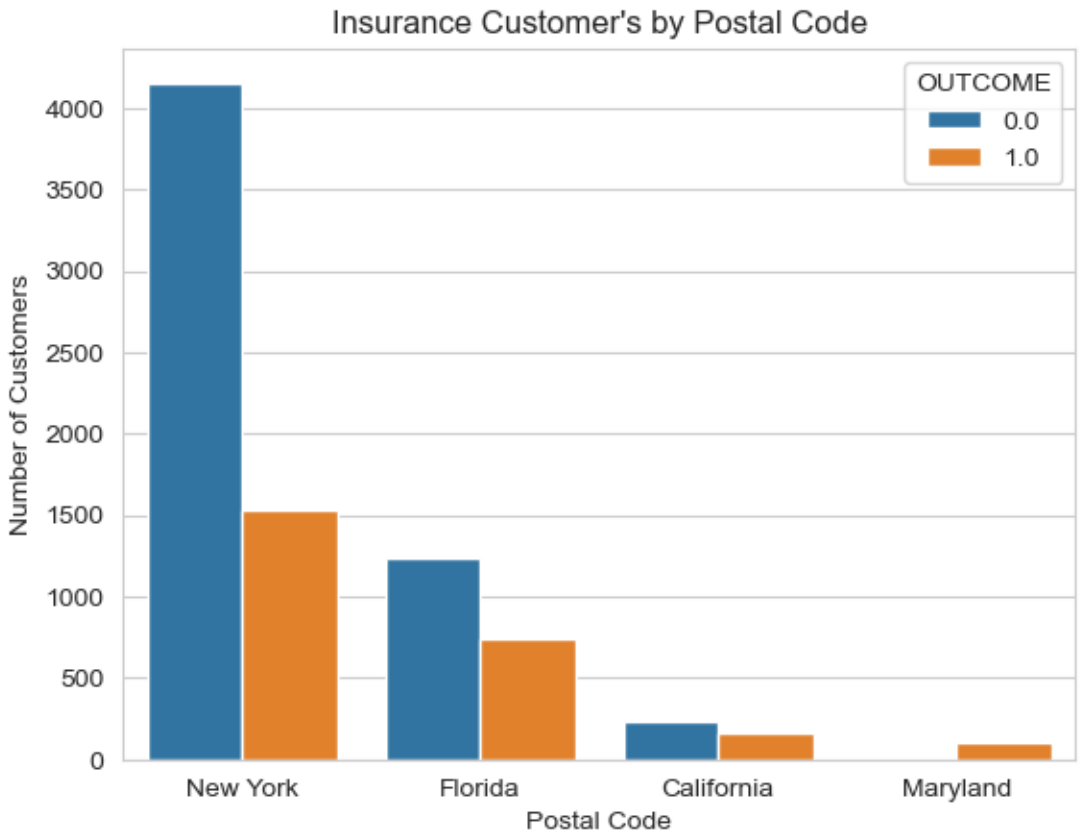
This graph shows the correlation between income level and the average number of DUIs. I found it interesting that DUIs would increase with increased income, an upward trend. This graph also shows that drivers with higher income are making more insurance claims.



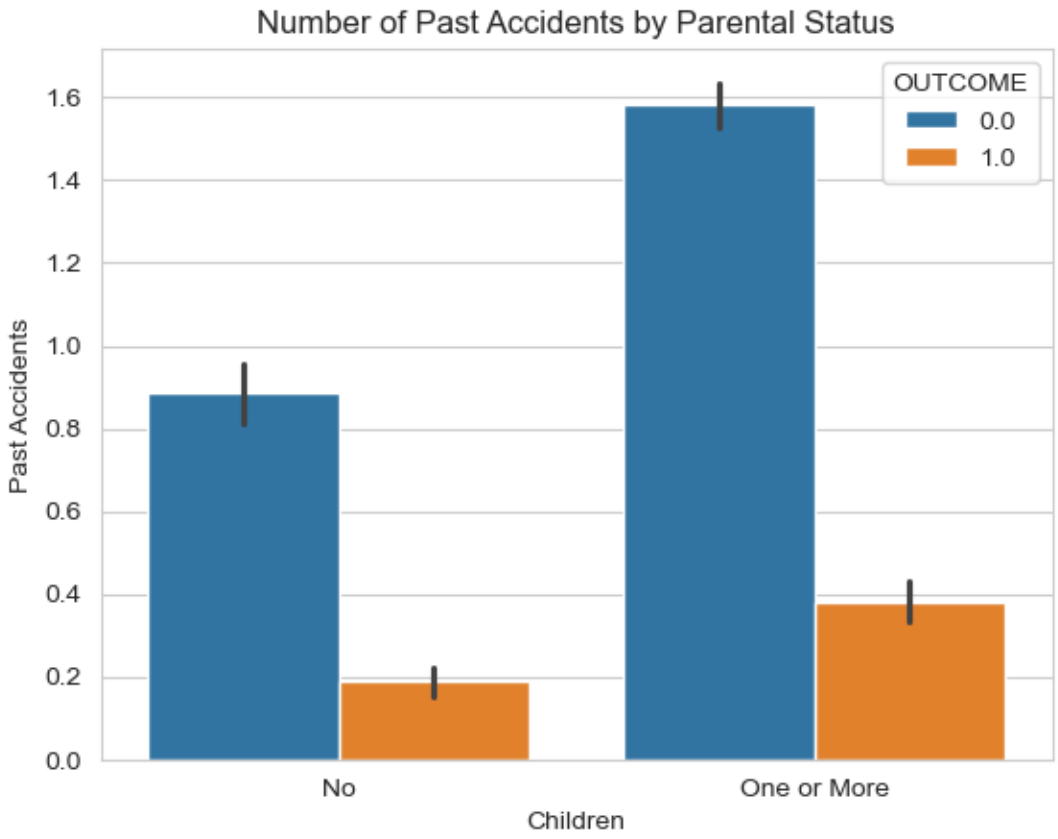
This graph shows the number of speeding violations per type of car. As a person who has had multiple speeding tickets in the past I wanted to see if the data would show any correlation with speeding and a fast car. Although sports car drivers do have slightly more speeding violations, I was surprised to see a very small delta. The two types of cars also tend to have the same number of claims.



This graph shows the number of insurance customers per range of driving experience as well as the number of claims submitted for each range. This graph was not groundbreaking as people with little driving experience tend to have the more insurance claims. However, it did reenforce the validity of the dataset for me as you could see a clear trend of decreasing claims, as the driving experience goes up. It would have been interesting to see smaller ranges to get a better idea of the slope of the trend.



This graph shows the number of insurance customers by their postal code and the number of claims submitted from each postal code. This graph was an important part of refining the data after the first algorithm was made. Due to New York holding more than half of the dataset's spread, the algorithm would be skewed. Since Maryland and California have so little datapoints their claim to no claim ratio is a lot closer to 1:1 or even 0:1 than New Yorks 2.5:1. After seeing this graph, it was clear that removing the 3 other locations would create a strong algorithm for New York.



This graph shows the correlation between the number of past accidents and the parental status of the driver. A parent driver shows to have double the number of previous accidents than drivers without children. The claims submitted are also doubled when the driver is a parent. This could be the case due to a parent driving distracted trying to tend to their children.

Algorithm Training

A logistic regression algorithm was used on this data since there were multiple factors that played into a claim being submitted that were not linear. The outcome column holding the claim status of each customer was separated the data was fed into the splitting function. The function splits the data into two groups based on a given random seed

number and the given test size value. I setup a test size of 30% and I set the maximum iterations to 10,000 on the logistic regression function.

```
x, y = insurance_data.drop("OUTCOME", axis=1), insurance_data["OUTCOME"].values

#Training the model
from sklearn.model_selection import train_test_split

#Randomly splits the dataset into two groups where 70% of the data is used to
train and the remaining 30% is used to test
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3,
random_state=77)

from sklearn.linear_model import LogisticRegression

logmodel = LogisticRegression(max_iter=10000)
#All the math magic happens here
logmodel.fit(x_train,y_train)

predictions = logmodel.predict(x_test)

from sklearn.metrics import classification_report
#The trained model is then used to predict the values of the 30% group, and
this returns the accuracy of the model
print(classification_report(y_test,predictions))
```

Model Evaluation

The initial results of the model were strong with an accuracy of 84% and an F1 score of 89% on no claim and 74% on a given claim. The precision and recall on ‘no claim’ were higher than a ‘given claim’ since the dataset provided double the examples of ‘no claim’. Precision and recall on a ‘given claim’ were above 70% which means that the model could pick out the relevant pieces of data and out of everything picked more than 70% of it was useful. F1- scores given a harmonic mean between precision and recall, which means the model has low false positives and low false negatives. In order words the model as is, could be used to accurately determine if an individual’s insurance would go up or down the following year.

	precision	recall	f1-score	support
0.0	0.88	0.90	0.89	1692
1.0	0.75	0.72	0.74	753
accuracy			0.84	2445
macro avg	0.82	0.81	0.81	2445
weighted avg	0.84	0.84	0.84	2445

To improve the model further for the location of New York I first removed the rows of data that were from the other locations. The other 3 locations made up about 30% of the data which would skew results for future testing. The remaining dataset contained over 5600 rows of data local to New York. I removed the postal code column to reduce redundancy, split the data from the outcome column and ran the same logistic regression training. Most values improved by 1-3% but more importantly the model would be accurate and reliable to New York.

	precision	recall	f1-score	support
0.0	0.91	0.92	0.92	1268
1.0	0.76	0.74	0.75	438
accuracy			0.87	1706
macro avg	0.84	0.83	0.83	1706
weighted avg	0.87	0.87	0.87	1706

Conclusion

Based on the previous year of insurance data, would my insurance go up or down? Assuming that insurance companies are bound to increase the premiums of drivers and cars that make the most claims, this algorithm was able to answer my question. I ran a test set of 4 individuals in my family, myself, my dad, my sister and my brother. The results were impressive; I put in the data that would've been for 2024 for all of us and it predicted our

claims correctly for 2025 even though we didn't live in New York. I personally was predicted to have a claim, and I did for a cracked windshield. My father and brother were predicted not to have a any claim which they did not thankfully. Finally, my sister was predicted to have a claim which she did have for a light bumper accident. Although we haven't had our renewals yet, I am sure the claims will increase our premiums. The test set results did not change after refining of the dataset which makes sense as I used New York as the location since it was the closest to Toronto. Overall, it was extremely satisfying to make a model that accurately predicted a real-life scenario and answered a question that could predict a real financial impact.

myTest = {
"AGE" : [1],
"GENDER" : [0],
"RACE" : [1],
"DRIVING_EXPERIENCE" : [0],
"EDUCATION" : [1],
"INCOME" : [1],
"CREDIT_SCORE" : [0.7],
"VEHICLE_OWNERSHIP" : [0],
"VEHICLE_YEAR" : [1],
"MARRIED" : [0],
"CHILDREN" : [0],
"POSTAL_CODE" : [0],
"ANNUAL_MILEAGE" : [16000],
"VEHICLE_TYPE" : [1],
"SPEEDING_VIOLATIONS" : [0],
"DUIS" : [0],
"PAST_ACCIDENTS" : [0]
}
	myDad = {
	"AGE" : [2],
	"GENDER" : [0],
	"RACE" : [1],
	"DRIVING_EXPERIENCE" : [3],
	"EDUCATION" : [2],
	"INCOME" : [2],
	"CREDIT_SCORE" : [0.7],
	"VEHICLE_OWNERSHIP" : [1],
	"VEHICLE_YEAR" : [1],
	"MARRIED" : [1],
	"CHILDREN" : [1],
	"POSTAL_CODE" : [0],
	"ANNUAL_MILEAGE" : [16000],
	"VEHICLE_TYPE" : [0],
	"SPEEDING_VIOLATIONS" : [0],
	"DUIS" : [0],
	"PAST_ACCIDENTS" : [1]
	}
		myBrother = {
		"AGE" : [0],
		"GENDER" : [0],
		"RACE" : [1],
		"DRIVING_EXPERIENCE" : [0],
		"EDUCATION" : [0],
		"INCOME" : [0],
		"CREDIT_SCORE" : [0],
		"VEHICLE_OWNERSHIP" : [0],
		"VEHICLE_YEAR" : [1],
		"MARRIED" : [0],
		"CHILDREN" : [0],
		"POSTAL_CODE" : [0],
		"ANNUAL_MILEAGE" : [1000],
		"VEHICLE_TYPE" : [0],
		"SPEEDING_VIOLATIONS" : [0],
		"DUIS" : [0],
		"PAST_ACCIDENTS" : [0]
		}
			mySister = {
			"AGE" : [1],
			"GENDER" : [1],
			"RACE" : [1],
			"DRIVING_EXPERIENCE" : [0],
			"EDUCATION" : [2],
			"INCOME" : [2],
			"CREDIT_SCORE" : [0.7],
			"VEHICLE_OWNERSHIP" : [0],
			"VEHICLE_YEAR" : [1],
			"MARRIED" : [0],
			"CHILDREN" : [0],
			"POSTAL_CODE" : [0],
			"ANNUAL_MILEAGE" : [16000],
			"VEHICLE_TYPE" : [1],
			"SPEEDING_VIOLATIONS" : [0],
			"DUIS" : [0],
			"PAST_ACCIDENTS" : [1]
			}

Roshan: [1.]	Dad: [0.]	Brother: [0.]	Sister: [1.]