Predicting Car Insurance Claims

Maria Khan

Artificial Intelligence, Fall 2023

Contents

1 Abstract

The car insurance industry in the U.S is worth hundreds of billions of dollars. Given widespread vehicle ownership and legal requirements for auto insurance, this study which uses annual data from a car insurance company was conducted to predict customer car insurance claims using Artificial Intelligence. Factors such as driving experience, education, income, credit score, and vehicle details among others are considered. Model comparison reveals the Neural Network Model (2-1) as the most accurate. In exploring neural network parameters, increasing layers increases training accuracy but diminishes validation accuracy, indicative of pattern memorization and reduced generalization. These observations offer insurance companies valuable insights for understanding and managing customer behavior.

2 Introduction

Operating a vehicle requires having car insurance. This study utilizes annual data provided by a car insurance company, aiming to predict the likelihood of customers filing car insurance claims. The prediction is based on various factors such as the customer's driving experience, education, income, credit score, vehicle ownership, vehicle year, age, and more. Employing Artificial Intelligence enables the anticipation of customers likely to make insurance claims, offering valuable insights for insurance companies to understand and manage customer behavior. This project focuses on implementing a neural network model to predict which customers are prone to making car insurance claims.

3 Data Analysis

3.1 Dataset

The dataset was obtained from the data science website *Kaggle*. The dataset is titled *Car Insurance Data*. It comprises various features including Age, Gender, Race, Driving Experience, Education, Income, Credit Score, Vehicle Owner and Vehicle Year.

3.2 Dataset Description

The data has 442696 rows and 19 columns. Of the 19 columns, one is for 'ID' which was not needed. The remaining input features are the following:

- Age
- Gender
- Race
- Diving Experience
- Education
- Income
- Credit Score
- Vehicle Ownership
- Vehicle Year
- Married

- Children
- Postal Code
- Vehicle Year
- Annual Mileage
- Vehicle Type
- Speeding Violations
- DUIS
- Past Accidents

3.3 Input Data Visualization

The graph below illustrates the distribution of input features.

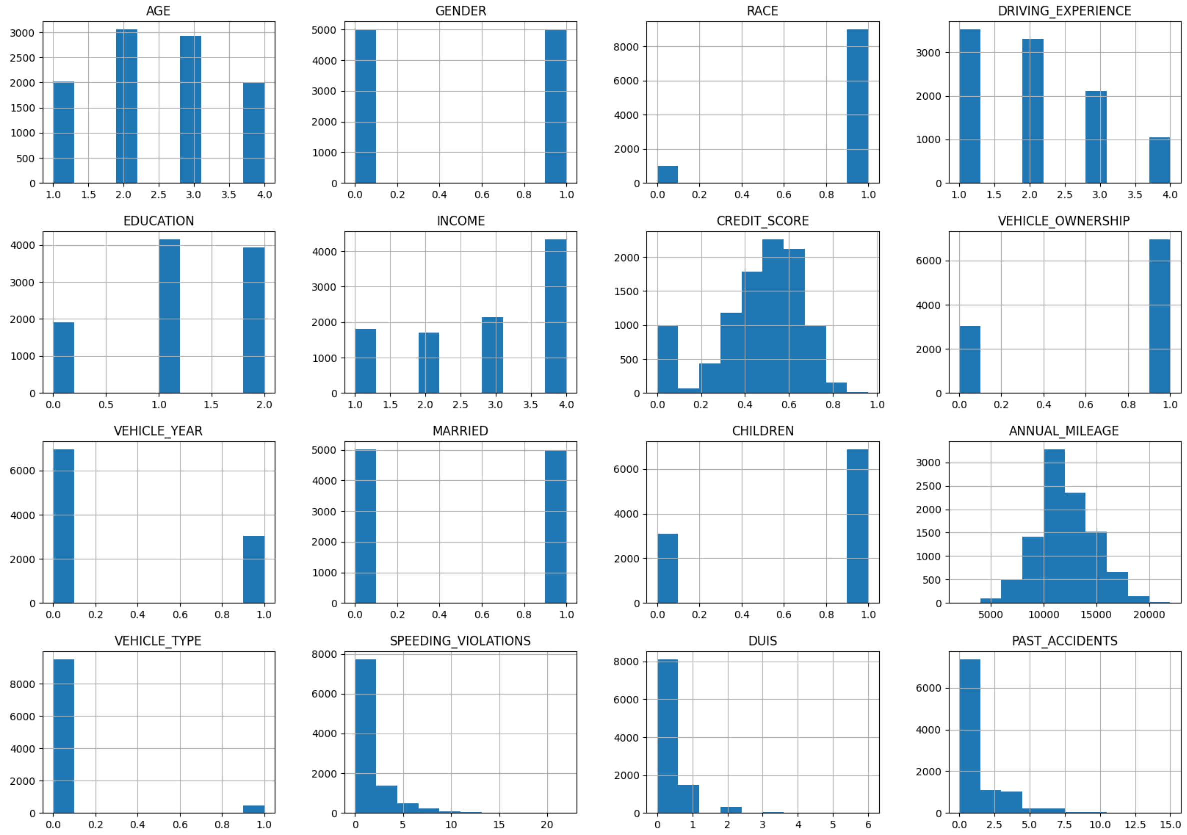

Figure 1: Distribution of Input Features

3.4 Output Data Visualization

The output label Class 0 corresponds to customers who did not file an insurance claim while Class 1 represents individuals who did file a claim. The distribution of output labels is not imbalanced as Class 0 is 66.57% and Class 1 is 31.33%.

Variable	Min	Max	Mean	SD
Age	1.00	4.00	2.39	1.03
Gender	0.00	1.00	0.50	0.50
Race	0.00	1.00	0.90	0.30
Diving Experience	1.00	4.00	2.07	0.99
Education	0.00	2.00	1.20	0.74
Income	1.00	4.00	2.90	1.15
Credit Score	0.00	0.96	0.47	0.20
Vehicle Ownership	0.00	1.00	0.70	0.46
Vehicle Year	0.00	1.00	0.30	0.46
Married	0.00	1.00	0.50	0.50
Children	0.00	1.00	0.69	0.46
Annual Mileage	2000	22 000	11 697	2680.17
Vehicle Type	0.00	1.00	0.05	0.21
Speeding Violations	0.00	22.00	1.48	2.24
DUIs	0	6.00	0.24	0.55
Past Accidents	0.00	15.00	1.06	1.65

Table 1: Range of Values for Input Features

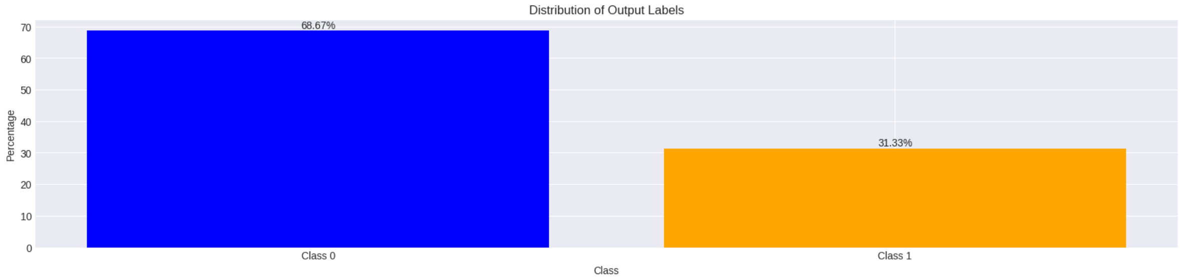

Figure 2: Distribution of Output Labels

4 Data Processing

4.1 Features Encoding

The table below shows the mapping of features which have been converted into numerical format.

Feature	**Encoding**
Age	'16-25' → 1, '26-39' → 2, '40-64' → 3, '65+' → 4
Gender	'female'→ 0,'male'→ 1
Race	'minority' → 0, 'majority' → 1
Driving Experience	'0-9y' → 1,'10-19y' → 2,'20-29y' → 3,'30y+' → 4
Education	'none' → 0, 'high school' → 1,'university' → 2
Income	'poverty' → 1, 'working class' → 2, 'middle class' → 3,'upper class' → 4
Vehicle Year	'before 2015' → 0', after 2015': → 1,
Vehicle Type	'sedan' → 0,'sports car' → 1

Table 2: Encoding of Features

4.2 Data Normalization

The data was not distributed uniformly. Z-Score normalization was used to rescale the distribution with a mean of 0 and a standard deviation of 1 in order to have a standard normal distribution.

The Z-score formula is given by:

$$Z = \frac{X - \mu}{\sigma}$$

where:

$$Z : \text{Z-score}$$

$$X : \text{Observed value}$$

$$\mu : \text{Mean of the population}$$

$$\sigma : \text{Standard deviation of the population}$$

5 Data Analysis

5.1 Relationship between Input and Output Features

The co-relation matrix below represents how strongly and in what direction pairs of variables are co-related to each other. Positive correlations imply that as one variable increases, the other tends to increase, while negative correlations imply an inverse relationship.

Below is a visual representation of a set of box plots where each box plot represents the distribution of a particular input variable with respect to target variable i.e Outcome.

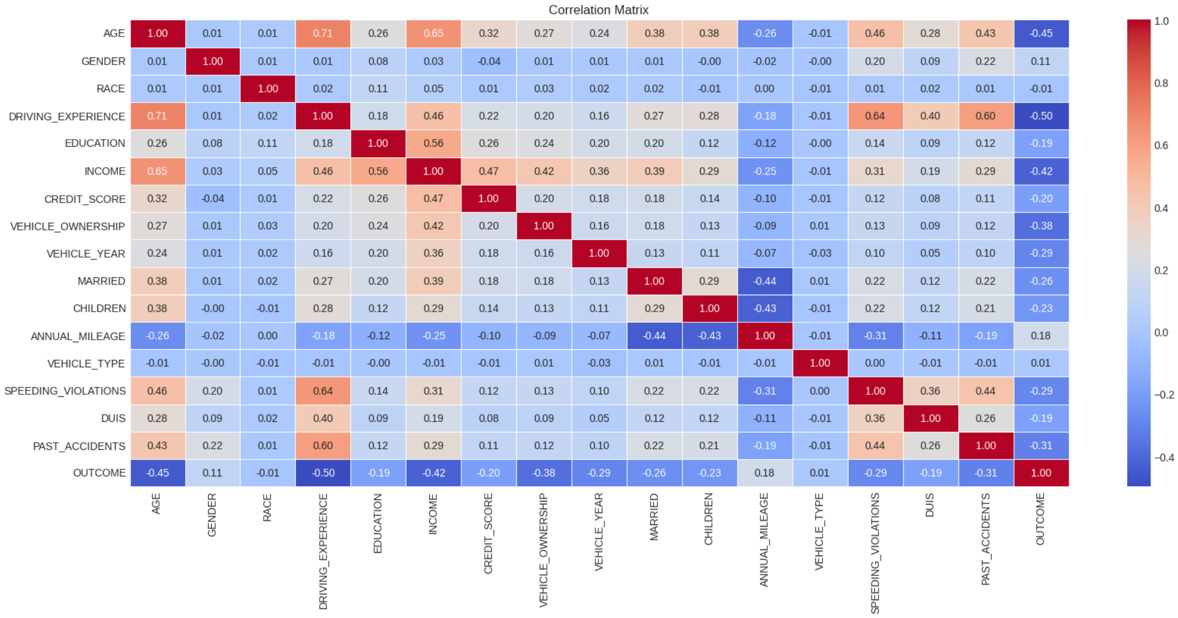

Figure 3: Co-relation Matrix

6 Model Evaluation

6.1 Performance Comparison

A comparison of performance of different models is given in the table below.

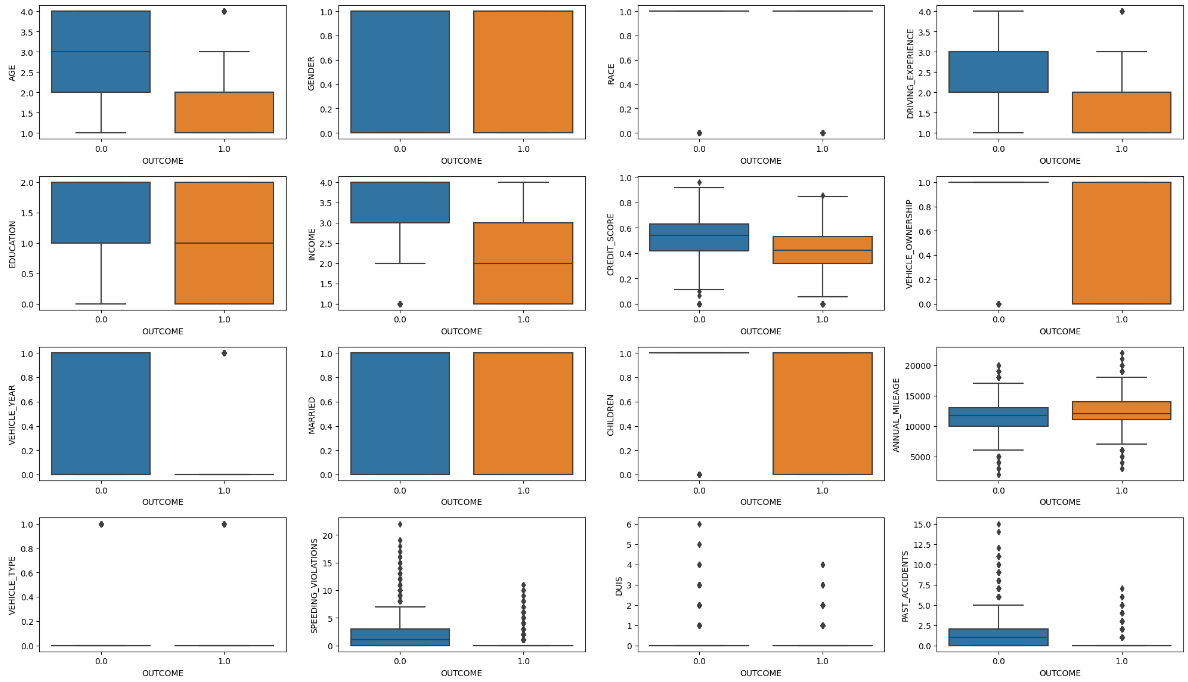

Figure 4: Box Plot

6.2 Best Model

The best performing model is the Neural Network Model (2-1). This is based on the accuracy score of the validation set.

6.3 Learning Curve of the Neural Network

After splitting randomly shuffled data into training and validation set, random baseline classifier is the model with the least validation accuracy while logistic regression model had the highest accuracy among chosen ML algorithm models. After experimenting with a different number of epochs and parameters for the neural network models, it was observed that as the number of layers increased, the training accuracy also increased, however, the validation accuracy decreased. This discrepancy is due to the model memorizing patterns

Model	**Training Acc**	**Validation Acc**	**Epochs**	**Parameters**
Random baseline classifier	-	56.75%	-	-
Logistic regression model	-	83.25%	-	-
Random Forest	-	81.70%	-	-
Neural network model (32-16-8-4-2-1)	90.1%	78.4%	300	1257
Neural network model (16-8-4-2-1)	86.6%	81.5%	150	457
Neural network model (8-4-2-1)	84.4%	82.8%	50	81
Neural network model (4-2-1)	84.1%	83.0%	100	81
Neural network model (2-1)	84.4%	83.4%	50	37
Neural network model (1)	84.5%	83.3%	50	17

Table 3: Model Comparison

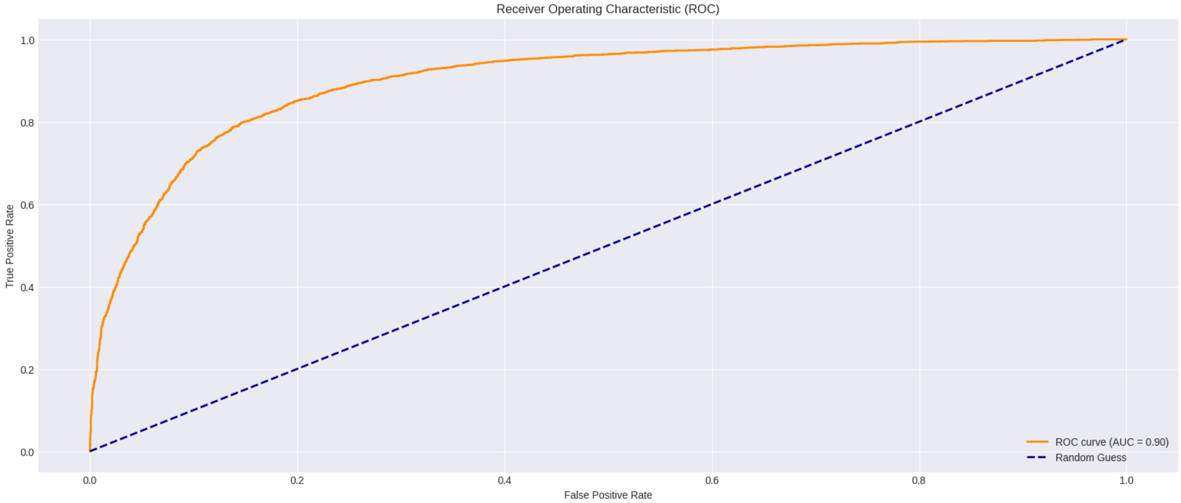

Figure 5: Receiver Operating Characteristic (ROC)

resulting in diminished performance on the validation set.

7 Feature Importance Analysis

The graph below illustrates validation accuracies for individual models, where each input feature is treated independently or as a standalone feature for training the model. These accuracies reflect the relative significance of each input variable. Notably, Driving Experience is the most important input feature, achieving an accuracy of 79.7%.

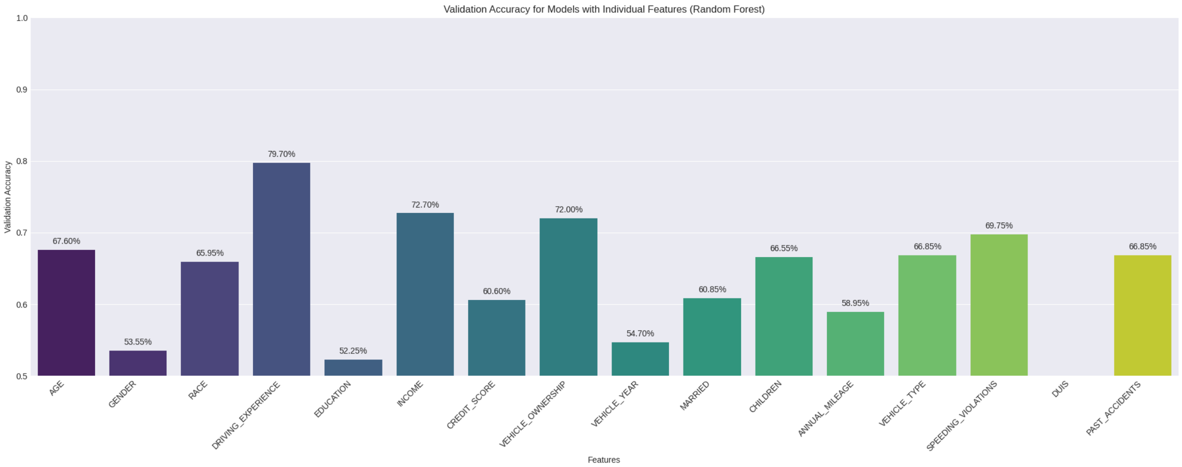

Figure 6: Validation Accuracy for Models with Individual Features (Random Forest)

In the subsequent phase, features were systematically eliminated, starting with the least significant i.e DUIS. The process continued iteratively, removing features based on their relative importance. The model employing all features attains an accuracy of 82%. Nevertheless, various models with reduced features depict distinct accuracies.

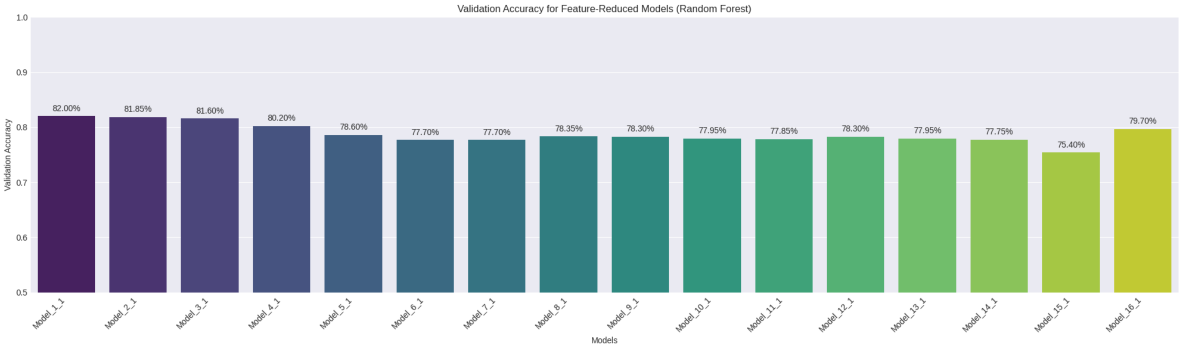

Figure 7: Validation Accuracy for Feature-Reduced Models (Random Forest)

8 Graduation Requirement: Architecture

8.1 Part 1

Overfitting, which is when a model learns the training data to the point that it negatively impacts its generalization capability, is more likely to occur when the model is too complex relative to the amount of training data available. When you have the output as an additional input feature, the risk of overfitting may increase, as the model has the potential to memorize the training data, including the target variable i.e output as part of the input features. This type of model is a feedforward neural network where information flows in one direction, from input to output. When we treat output of a model as an additional input feature, it introduces a form of recurrence or feedback loop in the architecture. This feedback loop enables the network to maintain a memory of previous inputs and hidden states. The model we trained, that is of 128 neurons, is relatively deep, with multiple hidden layers, each containing a decreasing number of neurons. This architecture has the potential to capture intricate patterns in the training data, including noise.

8.2 Part 2

The function/method that serves as a prediction model had a very similar accuracy score as the one obtained using the trained model. ModelCheckpoint callback was used to save the weights of the model when the validation accuracy improves. The scores for the training set were 84.5 for the function and 83.4 for the model. The scores for the validation set were 84.4 for the function and 83.4 for the model.

9 Written Report, Video Presentation and Code Access

The video presentation for this report is broken into phases. They provide a short summary of what was accomplished in that particular phase. The report, code as well as the videos can be accessed here.

10 Conclusion

In this research, I developed a neural network model to predict which customers are likely to submit a car insurance claim. I investigated different machine learning algorithms and explored neural network architectures spanning from single- layer to multi-layered architecture. I documented the performance of each model and determined that the most effective one was the 2-1 nueral network model.

Additionally, I assessed the significance of each feature in influencing the model's performance by examining the relationship between input features and output.