Grenze International Journal of Engineering and Technology, June Issue

Accident Prediction using Machine Learning

Yash Mehta[1] and Abhay Kolhe[2]
[1-2]Department of Computer Engineering, Mukesh Patel School of Technology Management & Engineering, SVKM's Narsee Monjee Institute of Management Studies (NMIMS) Deemed-to-be University, Mumbai-400056, India.
Email: myash299@gmail.com, akkolhe71@gmail.com

***Abstract*— Accidents are major cause in cities. Various factors contribute to accident. The accident could be because of the driver, driving at a high speed or the driver condition. But other factors like weather, visibility, Road Condition, etc also contributes to making of accidents. The paper leverages established various machine learning techniques implemented on a dataset, which contains over 3000 data. The data is trained and tested with various techniques and the accuracy is tested for each of them. Also, an additional dataset with just 300 values was also trained using the same technique to see for the differences and similarity in the accuracy of the model. The model's output gives the estimate of the factor which contributes largely towards the making of accident. Additionally, it also shows the day and the time, when accidents have occurred the most.**

***Index Terms*— Machine learning, Dataset, Hyperparameter tuning, Model, Feature Importance and ML Techniques.**

I. INTRODUCTION

Machine learning is an evolving branch of computational algorithms that are designed to emulate human intelligence by learning from the surrounding environment [8]. They are considered the working horse in the new era of the so-called big data [8]. Techniques based on machine learning have been applied successfully in diverse fields ranging from pattern recognition, computer vision, spacecraft engineering, finance, entertainment, and computational biology to biomedical and medical applications [8].
Road crash is bigger problem or a use concern. Every year millions of people die as a result of road crash. And this number keeps growing exponential over the years. Thus, road crash prediction models play a vital role in highway safety. Crash frequency refers to the prediction of the number of crashes in a certain time period[9]. To predict this frequency, we have developed a machine learning model to predict the factors which are likely to increase the chances of the accident. The model was tested against a dataset of size 3000 past accidents.

II. PROCEDURE

A. Selecting a Dataset

We Selected a Dataset with 3000+ data, in-order to increase the accuracy of the model. The dataset columns are Age of driver, Engine CC, Time, Road Surface, Weather condition, Type of Vehicle, which are the main reasons which contributes to Accident. Also, additional a trimmed Dataset size of 300 values was also selected, so that we can compare the behavior and accuracy of the model under different condition and different amount of data. Also, we can compare how the dataset behaves with different machine learning approaches.

Grenze ID: 01.GIJET.10.2.99_1

B. Data Cleaning

The dataset was checked for null or missing values. This is done so that the model accuracy doesn't decrease. The missing data can be filled with values using different techniques like: the missing value row can be deleted, a random value can be inserted, Average value of the columns present data can be given to all missing values, Backward fill or forward fill. In each case the accuracy declines by some percentage. In our case, we choose to concatenated the "Date" and "Time" columns, filtered out the rows with missing values, and then we choose to drop the redundant columns and the rows with missing values. We choose this method, to maintain the integrity of our data, which may result in data loss. Also, for preserving the sample size and stopping the dataset from getting affected by any noise.

```
#combining two columns
accidents['Date_time'] = accidents['Date'] +' '+ accidents['Time']

for col in accidents.columns:
    accidents = (accidents[accidents[col]!=-1])

for col in casualties.columns:
    casualties = (casualties[casualties[col]!=-1])

accidents['Date_time'] = pd.to_datetime(accidents.Date_time)
accidents.drop(['Date','Time'],axis =1 , inplace=True)
accidents.dropna(inplace=True)
```

Figure 1. Treating the missing Values in Python

C. Normalize the data

Data normalization is a fashion used in data mining to transfigure the values of a dataset into a common scale. This is important because numerous machine learning algorithms are sensitive to the scale of the input features and can produce better results when the data is regularized.
We used Logarithmic metamorphosis to homogenize the data.

D. Test-Train-Split

Firstly, we divided our data into features (X) and labels (y). The dataframe gets divided into X_train, X_test , y_train and y_test. X_train and y_train sets are used for training and fitting the model. The X_test and y_test sets are used for testing the model if it's predicting the right outputs/labels. So that we can explicitly test the size of the train and test sets.

```
[ ] accident_ml = accidents.drop('Accident_Severity' ,axis=1)
    accident_ml = accident_ml[['Did_Police_Officer_Attend_Scen
                               , 'Light_Conditions', 'Sex_of_Dr

    accident_ml.head()

    # Split the data into a training and test set.
    X_train, X_test, y_train, y_test = train_test_split(accide
                                                        accidents['A

    print("done")

    done
```

Figure 2. Splitting the data into training data and test data in Python

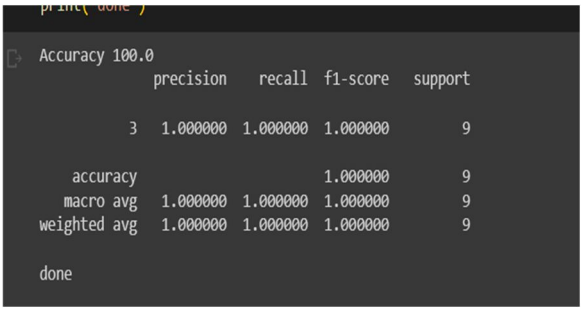
III. Machine Learning Techniques

We trained the dataset using different machine learning techniques. These techniques are used to determine the pattern in the dataset. Basically, this technique is use to undercover the pattern underneath the data. The pattern can be used to determine the further values in the dataset or the graph. Can also be used to determine the value for the missing data, which was assumed in the dataset.

A. Random Forest

When doing ensemble learning for classification, regression, and other issues, random forests, often referred to as random choice forests, create a lot of decision trees throughout the training process. The output of a random forest is the categorization that the majority of trees select in issues involving classification. Regression tasks return the mean or average forecast of each individual tree. Random decision forests counteract the propensity of decision

trees to overfit their training set. While generally speaking, random forests outperform choice trees, their accuracy is inferior to that of gradient enhanced trees. Performance, however, may be impacted by the quality of the data.



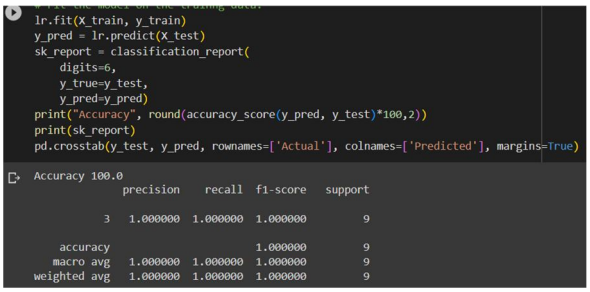
```
print("done")

Accuracy 100.0
              precision    recall  f1-score   support

           3   1.000000  1.000000  1.000000         9

    accuracy                       1.000000         9
   macro avg   1.000000  1.000000  1.000000         9
weighted avg   1.000000  1.000000  1.000000         9

done
```

Figure 3. Accuracy of the Random Forest Method, when tested against our dataset

As we know, the model was also tested using only 300 data values as well. In that case also, we get the same accuracy for the less amount of data. This is because, in random forest, the output of multiple decision tree is used to reach the output for the data.

B. Logistic Regression

Logistic regression is a statistical approach for developing machine learning models using dichotomous dependent variables, i.e. binary. Logistic regression is a statistical technique used to describe data and the connection between one dependent variable and one or more independent variables. Independent variables may be nominal, ordinal, or interval in nature. The term "logistic regression" is derived from the logistic function that it employs. The sigmoid function is another name for the logistic function. This logistic function has a value between zero and one.



```
lr.fit(X_train, y_train)
y_pred = lr.predict(X_test)
sk_report = classification_report(
    digits=6,
    y_true=y_test,
    y_pred=y_pred)
print("Accuracy", round(accuracy_score(y_pred, y_test)*100,2))
print(sk_report)
pd.crosstab(y_test, y_pred, rownames=['Actual'], colnames=['Predicted'], margins=True)

Accuracy 100.0
              precision    recall  f1-score   support

           3   1.000000  1.000000  1.000000         9

    accuracy                       1.000000         9
   macro avg   1.000000  1.000000  1.000000         9
weighted avg   1.000000  1.000000  1.000000         9
```

Figure 4. Accuracy of the Logistic Regression Method, when tested against our dataset

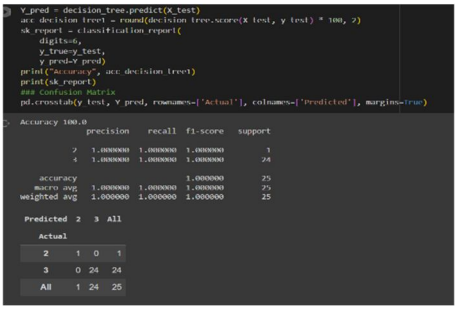
While the accuracy remained consistent over the smaller sample, this does not always imply a flawless model. Logistic regression uses a sigmoid function, which looks like a "S" shape, to translate data between 0 and 1. This modification guarantees that the model's output include probability. However, when working with a smaller dataset, the model may struggle to capture the complete range of potential values. This could lead to overfitting, where the model excels with the training data but struggles to generalize to unknown data.

C. Decision Tree

A decision tree is a non-parametric supervised learning technique that may be used to perform classification and regression problems. It is a hierarchical tree with a root node, branches, internal nodes, and leaf nodes. Decision trees are used for classification and regression applications, resulting in simple models. A decision tree is a hierarchical decision support model that displays options and their probable consequences, including chance occurrences, resource expenditures, and utility. A root node, branches, internal nodes, and leaf nodes create a hierarchical, tree-like structure in the tree structure.

It is a versatile instrument with a wide range of uses. The name implies that it uses a flowchart-like tree structure to display the predictions resulting from a sequence of feature-based splits. It begins with a root node and finishes with a leaf decision.

The accuracy of the model with dataset of size 300 was 100%, but as soon as it was tested on dataset with 3000+ values, the accuracy of the model decreased. This was because, when the model was trained using decision tree, we divided the input into different branches to get the output. When the training and testing dataset was too large, the tree had too many branches, eventually, the accuracy tends to deplete.

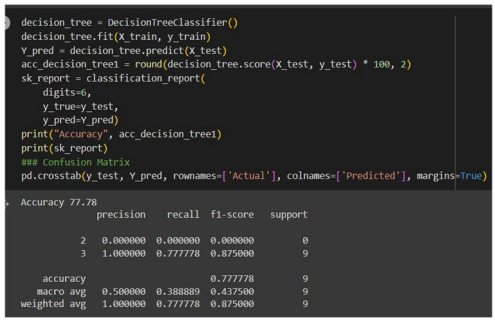


```
Y_pred = decision_tree.predict(X_test)
acc_decision_tree1 = round(decision_tree.score(X_test, y_test) * 100, 2)
sk_report = classification_report(
    digits=6,
    y_true=y_test,
    y_pred=Y_pred)
print("Accuracy", acc_decision_tree1)
print(sk_report)
### Confusion Matrix
pd.crosstab(y_test, Y_pred, rownames=['Actual'], colnames=['Predicted'], margins=True)
```

```
Accuracy 100.0
              precision    recall  f1-score   support

           2   1.000000  1.000000  1.000000         1
           3   1.000000  1.000000  1.000000        24

    accuracy                       1.000000        25
   macro avg   1.000000  1.000000  1.000000        25
weighted avg   1.000000  1.000000  1.000000        25
```

Predicted	2	3	All
Actual			
2	1	0	1
3	0	24	24
All	1	24	25

Figure 5. Accuracy of the Decision Tree Method, when tested against our dataset size of 300



```
decision_tree = DecisionTreeClassifier()
decision_tree.fit(X_train, y_train)
Y_pred = decision_tree.predict(X_test)
acc_decision_tree1 = round(decision_tree.score(X_test, y_test) * 100, 2)
sk_report = classification_report(
    digits=6,
    y_true=y_test,
    y_pred=Y_pred)
print("Accuracy", acc_decision_tree1)
print(sk_report)
### Confusion Matrix
pd.crosstab(y_test, Y_pred, rownames=['Actual'], colnames=['Predicted'], margins=True)
```

```
Accuracy 77.78
              precision    recall  f1-score   support

           2   0.000000  0.000000  0.000000         0
           3   1.000000  0.777778  0.875000         9

    accuracy                       0.777778         9
   macro avg   0.500000  0.388889  0.437500         9
weighted avg   1.000000  0.777778  0.875000         9
```

Figure 6. Accuracy of the Decision Tree Method, when tested against our dataset size of 3000

IV. Hyper parameter Tuning

A Machine Learning model is defined as a mathematical model having a number of parameters that must be learned from data. We fitted our model parameters by training a model using existing data. Hyperparameter tuning is the process of finding the optimal settings for these hyperparameters. This is crucial because the wrong settings can significantly impact the model's performance. Techniques like Grid Search or Random Search are often used for hyperparameter tuning.

However, there is another type of parameter called as Hyperparameters that cannot be learnt directly from the standard training procedure. They are generally repaired before the training process begins. These parameters represent crucial features of the model, such as its complexity or how quickly it should learn.

A. Results

After hyperparameter tuning, the accuracy for each Machine learning Technique was 100%. The results were same every time, irrespective of the size of the dataset. Therefore, the model can be used to predict the feature that is playing a major role in accident. This can somewhere be used to predict the accident, by seeing the road condition and comparing the probability of accident on that road using the model.

Apart from this, the model can also be used to visualize the day and the time, where the probability of the accident increases.

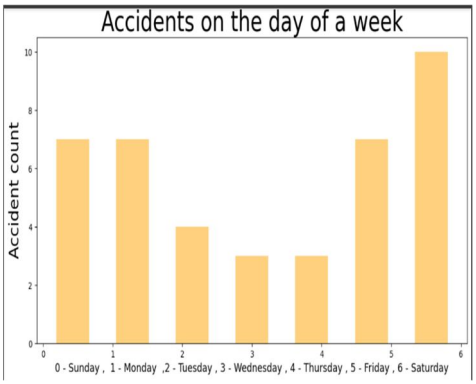


Figure 7. Displays the output of the Accident count according to each day of the week

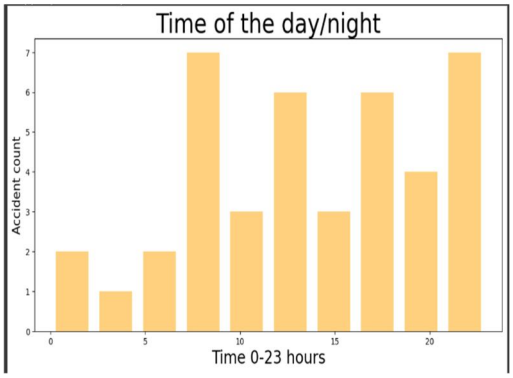


Figure 8. Displays the output of the Accident count according to time of the day

The bar graph can be easily be used to conclude the most of the accident takes place on Saturday, at night between 3:00pm to 5:00pm or after 8pm. Various theories can be made up to determine the realation between the day and the time of accident.

V. Feature Importance

Techniques that compute a score for all of the input characteristics for a particular model are referred to as feature importance techniques; the scores simply express the "importance" of each feature. A higher score indicates that the specific characteristic will have a greater impact on the model used to predict a certain variable.

A. Results

After hyperparameter tuning, the accuracy for each Machine learning Technique was 100%. The results were same every time, irrespective of the size of the dataset. Therefore, the model can be used to predict the feature that is playing a major role in accident. This can somewhere be used to predict the accident, by seeing the road condition and comparing the probability of accident on that road using the model.

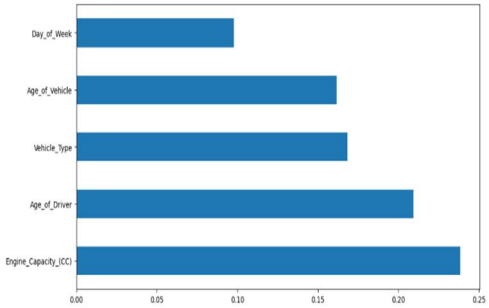


Figure 9. Displays the feature which contributes to accident

Feature importance tells the feature which are affecting or promoting the accident to happen. If we look at the bar chart, we can make out that Engine_Capacity and Age_of_Driver are two most important feature that causes the accident to happen. Basically, we conclude that, if the car is very fast, it is most likely to cause or meet an accident. Also, the age of the driver plays a mojor role in accident.
In case of the smaller dataset, the same two factors are likely to meet with the accident or cause an accident.

VI. Conclusion And Future Work

To sum up, our study was successful in using machine learning to identify the main factors that contribute to accidents. The models were highly accurate and provided useful insights into accident trends. However, more in-depth analysis is needed to understand the model's inner workings. Examining metrics other than accuracy (i.e. accuracy, recall, F 1-score) would provide a more in-depth look at the performance of the model. Finally, it's important to look at how generalizable the findings are. Testing the model on non-visible data and dealing with dataset limitations could have strengthened the overall impact of this study to improve road safety.

Appendix A: Declaration

Funding Statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declaration Of Interest's Statement

The authors declare no conflict of interest.

Additional Information

No additional information is available for this paper.

Acknowledgment

Thanks to our families & colleagues, who supported us morally.

References

[1] Shakil Ahmed, Md Akbar Hossain, Sayan Kumar Ray, Md Mafijul Islam Bhuiyan, Saifur Rahman Sabuj, "A study on road accident prediction and contributing factors using explainable machine learning models: analysis and performance, Transportation Research Interdisciplinary Perspectives", Volume 19, 2023.

[2] George Yannis, Anastasios Dragomanovits, Alexandra Laiou, Thomas Richter, Stephan Ruhl, Francesca La Torre, Lorenzo Domenichini, Daniel Graham, Niovi Karathodorou, Haojie Li, "Use of Accident Prediction Models in Road Safety Management – An International Inquiry Transportation Research Procedia", Volume 14, 2016.

[3] Pourroostaei Ardakani, S.; Liang, X.; Mengistu, K.T.; So, R.S.; Wei, X.; He, B.; Cheshmehzangi, A. Road Car Accident Prediction Using a Machine-Learning-Enabled Data Analysis. Sustainability 2023, 15, 5939.
[4] Feature Importance, "https://machinelearningmastery.com/calculate-feature-importance-with-python/" [Online], Accessed at: 18th September, 2023 at 20:00.
[5] Hyperparamter tununing, "https://www.jeremyjordan.me/hyperparameter-tuning/", [Online], accessed at: 19th September,2023, at 15:00.
[6] El Naqa, I., & Murphy, M. J. (2015). What is machine learning? (pp. 3-11). Springer International Publishing.
[7] Abdulhafedh, A. (2017). Road crash prediction models: different statistical modeling approaches. Journal of transportation technologies, 7(02), 190.
[8] Wright, R. E. (1995). Logistic regression.
[9] Suthaharan, S., & Suthaharan, S. (2016). Decision tree learning. Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning, 237-269.
[10] Dogru, N., & Subasi, A. (2018, February). Traffic accident detection using random forest classifier. In 2018 15th learning and technology conference (L&T) (pp. 40-45). IEEE.