**Automobile\_insurance\_fraud**

**The article should contain the following sub-topics:**

1. Problem Definition
2. Data Analysis
3. EDA Concluding Remarks
4. Pre-processing Pipeline
5. Building Machine Learning Models
6. Concluding remark

**Problem definition**

The insurance industry is a complex and multifaceted one with mint different types of polices & coverage option like car insurance health insurance life insurance there are wide verities of insurance product for the customers insurance fraud can happen it any stage of the process from the sale of the policy to the filing of the claim fraud can occur in the insurance of industry in a variety of ways including false claim, fake polices, identity theft insurers use a variety of tool and techniques including data analytics machine learning to prevent and detect the fraudulent activities it can also involve individuals or group and can be by polices holders insurers or third party given the complexity and scale of the problem it is clear that we need a sophisticated tool and technique to help and detect and prevent insurance fraud insurance fraud is a serious problem that can cost insurers and policy holders billions of dollar each year it can take many forms such as false claim fake policy identity theft as explained earlier that's why it is so important to have effective tools and techniques for detecting and preventing insurance frauds

​

so question come how we detect or prevent from insurance frauds, so min this project our goal is to develop a machine learning model to detect insurance fraud by analyzing the data on past claims and policy holder we hope ton identify the pattern and trends that might be inductive of the fraudulent activity.

​

​

so, problem statement of the project is with background that in mind our goal is to develop a machine learning model to detect insurance fraud we wanted to build a models that could analyze data on the past claims and policy holders and predict likelihood of claim being fraudulent. This would allow to insurer to identify and investigate suspicious claims they are paid out potentially saving millions of dollars in the process

1. **Data Analysis**

Data collection & preparation

here we collect the data and prepare the data we gather the large dataset of insurer claims along with information of policy holder and the claim themselves this include the data on the type policy the policyholder age, location and details of the claim itself. once we had the data then we clean it we had to preprocess it and get it ready for the further analysis this involve the task such as removing the missing value or indirect values, transforming the data into the format that could be used by our machine learning mode.

**EDA Concluding Remarks**

# Then after I call csv dataset after that i have to check the data type should have correct as well so I can check the null value. Then we see we have null value as well. those columns are mostly categorical columns and we can take out the percentage of the missing values.

# In this data set i got the percentage of missing value in those columns, here, I see that the most number of null value in the property are damage, so in this data set this column not required any more

# Now In this dataset here check the correlation to check the multi-colonity problems so here we can see that there are like some column who have the multi-colonity in there so have to like drop some of these columns as to reduce the problem of multi-colonity.

# Then after we check the unique value in this dataset. here we check the unique value in each of categorical columns because if columns have more unique value then the variance of that column is so much high.so we have remove those kind of dataset for further analysis.

# After removing column:- (policy\_number','policy\_bind\_date','policy\_state','insured\_zip','incident\_date','incident\_location','incident\_state','incident\_city','insured\_hobbies','auto\_make','auto\_model','auto\_year','\_c39)

# Then after we got this columns:-

# Col-Name, months\_as\_customer, age, policy\_csl, policy\_deductable, policy\_annual\_premium umbrella\_limit, insured\_sex, insured\_education\_level, insured\_occupation, insured\_relationship, number\_of\_vehicles\_involved, property\_damage, bodily\_injuries witnesses, police\_report\_available, total\_claim\_amount, injury\_claim, property\_claim vehicle\_claim, fraud\_reported

# Now we check with the problem of multi-colonity

# As we can see the age column and the month as the customer column have the very high correlation along with the correlation of the like a total game amount with a vehicle claim property claim and reclaim these columns are also having the very high multicolonity so we have remove those coloumns.now can drop the column of age and total claim amt. then we dropped the age and total claim column.

# After that Here get the target variable single variable and the independent variables or other variables and then we standardizations and the one hot encoding as well so all them category papers that I used I can make them into the numerical by using the one hot encoding then i check my numerical columns as well and then i check and concatenate both them to make a complete independent variable data set

# outlier check

# then after we check outlier as well for checking outlier I use train test and split. I split my data set into test and train samples and then we have to do the scaling process for scaling process always remember you can do the scaling only on the training data as fit transform but on test you can only do the transform otherwise it will do the problem of data leakage. that's why we can all the standardization as well.

**Building Machine Learning Models**

# Modeling support vector classifier

# Here we call the svc from the sqln library add my machine learning models and then after I can check the score of my training and my testing as well then we see our dataset is under fitting and over fitting then you can check the score of confusion Metrix and also the classification report as well as you see it gave testing accuracy of 0.7 that mean it is a 0.77

# K - nearest neighborhood

# Then after we call knn model. we use knn model by using same approch and it also give .77accuracy in the testing phase.

# Decision Tree classifier

# Then after we we use Decision tree classifier as well it has also gave traing 1.0 and testing accuracy 0.72.

# Random Forest Classification

# Here we use random forest classification as well.

# Model comparison

# Now here i compare all models and got score.

# Model score

# 0 svc\_model 0.772

# 1 knn\_model 0.772

# 2 Decision Tree 0.752

# 3 Random Forest 0.772