**Problem Definition**

Insurance fraud is a serious issue that results in losses of billions of dollars annually for insurance companies. Detecting fraudulent claims is essential to reduce financial losses and maintain fair premium rates for policyholders. Insurance frauds cover a range of improper activities which an individual may commit in order to achieve a favourable outcome from the insurance company. This could range from staging the incident, misrepresenting the situation and the cause of incident and finally the extent of damage caused.

**Problem Statement**: The automobile insurance industry faces significant challenges in identifying fraudulent claims due to the complexity and volume of claims data. Traditional methods are often insufficient to detect sophisticated fraud activities. The goal of this project is to build a machine learning model that can predict auto insurance fraud. The challenge behind this is that frauds are far less common as compared to legitimate insurance claims.

**Data**: This project took advantage of the data which is the largest asset which insurers have in the fight against fraud. The dataset had the details of the insurance policy along with the details of the customer. It also had the details of the accident on the basis of which the claims were made.

**Method**: Historical claims data was analysed to identify patterns indicative of fraud. Then models were built using machine learning algorithms which can predict fraud. The performance of the models were evaluated and the models were tuned for better accuracy.

**Exploratory Data Analysis**

Exploratory data analysis is the process of analyzing and visualizing data to understand its main characteristics before applying any machine learning algorithms. In this project, the dataset consisted of 1000 rows and 39 columns. The column “fraud\_reported” was the dependent variable or the label. It was a categorical variable with only two values “Y” or “N”. So this problem was identified as a binary classification problem.

**Handling missing values and irrelevant features**:

1. Initially, the features "policy\_number", "insured\_zip" and "incident\_location" were found to be irrelevant and not contributing to the prediction of the label. So, they were removed from the dataset.
2. There were no null values or missing values found in the dataset.
3. It was found that some "?" values exist in the columns "collision\_type", "property\_damage" and "police\_report\_available" and were replaced with NaN values.
4. The NaN values in these columns were replaced with the mode of the respective columns as these were categorical variables.
5. The “policy\_bind\_date” and “incident\_date” columns were split into the respective day, month and year columns and these new features were added to the dataset. The “policy\_bind\_date” and “incident\_date” features were deleted from the dataset.
6. The “policy\_csl” feature was split into two new features “csl\_accidental” and “csl\_overall” and these were added to the dataset. “policy\_csl” was deleted from the dataset.
7. There were no duplicate records found in the dataset.
8. The column “incident\_year” had only one unique value, so it was removed from the dataset.

**Separating columns**: The column names were separated into continuous and categorical columns based on the values they contain.

**Statistical summary**: Statistical summary of the numerical columns was derived and count, mean, standard deviation, min, median, max, etc. of the columns were calculated and analysed. During analysis it was found that some columns had skewness and outliers.

**Data visualization**:

**Univariate Analysis**: Upon analysing the distribution of data for the continuous columns using dist plots, it was found that the column “capital\_gains” was right skewed and “capital\_loss” was left skewed. While analysing the distribution of data for the categorical columns using bar plots and pie plots, both the count and the percentage of the category was taken into account and the following insights were derived:

1. 352 or 35.2% of the people have the policy state as Ohio (OH) which is the most frequent policy state.
2. Largest number of people have policy deductable as 1000 with 351 or 35.1% of the people having this amount.
3. Largest number of people have the umbrella limit as 0 with 798 or 79.8% of people having this amount.
4. Female insurance claimants are slightly higher than males with 537 or 53.7% of the claimants being females.
5. Largest number of insurance claimants have the education level as JD with 161 or 16.1% of the claimants having this education level.
6. Largest number of insurance claimants work as machine-op-inspct with 93 or 9.3% of the people falling in this category.
7. Similarly, the count and percentages of other categorical columns were analysed.

**Bivariate analysis**: Upon analysing the correlation between the label “fraud\_reported” and continuous features using bar plots the following observations were recorded:

1. The features "months\_as\_customer", "age" and "policy\_annual\_premium" are unchanged with change in "fraud\_reported". So these features have no correlation with "fraud\_reported".
2. "capital\_gains" and "capital\_loss" change very little with change in "fraud\_reported". So these features have very low correlation with the label "fraud\_reported".
3. Fraud is reported for higher total claim amounts and no fraud is reported for lower total claim amounts. So "total\_claim\_amount" has significant correlation with "fraud\_reported".
4. Fraud is reported for higher injury claim amounts and no fraud is reported for lower injury claim amounts. So "injury\_claim" has significant correlation with "fraud\_reported".
5. Fraud is reported for higher property claim amounts and no fraud is reported for property claim amounts. So "property\_claim" has significant correlation with "fraud\_reported".
6. Fraud is reported for higher vehicle claim amounts and no fraud is reported for lower vehicle claim amounts. So "vehicle\_claim" has significant correlation with "fraud\_reported".

During the analysis of “fraud\_reported” for different categorical columns using bar plots, while the count was already available, the percentages of frauds reported for different categories with respect to the total number of claims received for each category were calculated which is actually the correct way of analysis. The percentages were analysed from which many useful insights were derived which are given as under:

1. Ohio (OH) has reported the highest percentage of frauds in insurance claims at 25.85% of the total number of claims received for this state and Illinois has reported the highest percentage of genuine insurance claims at 77.22% of the total number of claims received for this state.
2. The percentage of frauds reported for different policy deductable amounts is almost similar.
3. The percentage of fraudulent insurance claims reported for an umbrella limit of 2000000 is the highest at 66.67% of the total number of claims received for this category and for an umbrella limit of -1000000 the percentage of legitimate claims is highest at 100% that is, all the claims received for this category were legitimate.
4. The percentage of frauds reported for male insurance claimants is slightly higher than females at 26.13% of the total number of male claimants.
5. The percentage of fraudulent claims reported from claimants with all education levels is almost the same.
6. Claimants having occupation of exec-managerial have reported the highest percentage of fraudulent insurance claims at 36.84% and those having the occupation of other-service or priv-house-serv have reported the highest percentage of legitimate insurance claims at 83.1% each.
7. The claimants who have the hobby of Chess have reported the highest percentage of fraudulent claims at 82.61% and those having the hobby of Camping have reported the highest percentage of genuine claims at 90.91%.
8. Similarly, the percentages of fraudulent and legitimate claims were calculated and analysed other categorical columns.

The relationship between the features incident type and collision type was analysed using crosstab and bar plot and the following observations were recorded:

1. Multi-vehicle collision: 27.45% of multi-vehicle collisions are classified as front collisions, 36.28% as rear collisions and 36.28% as side collisions. Multi-vehicle collisions show a relatively even distribution among front, rear and side collision types.
2. Parked car: 100% of incidents involving parked cars are classified as rear collisions.
3. Single vehicle collision: 34.49% of single vehicle collisions are classified as front collisions, 34.74% as rear collisions and 30.77% as side collisions. Single vehicle collisions show a relatively balanced distribution among front, rear, and side collision types.
4. Vehicle theft: 100% of incidents involving vehicle theft are classified as rear collisions.

**Multivariate analysis**: The label fraud reported was analysed based on the features insured sex and policy annual premium using bar plot and it was found that there is no gender-based imbalance between the amount of policy annual premium and frauds reported. Also, fraud reported was analysed based on insured sex and total claim amount using bar plot and it was observed that the number of fraudulent claims reported is higher when the total claim amount is higher but in this case also there is no gender based imbalance between fraud reported and total claim amount.

**Outlier detection and removal**: Outliers are extreme values in a dataset which can skew statistical analyses and affect the performance of machine learning models. So, they need to be detected and removed. In this dataset, outliers were detected in the columns "age", "policy\_annual\_premium", "total\_claim\_amount" and "property\_claim" using box plot. These outliers were removed using Inter Quartile Range method and the percentage of data loss that occurred due to the removal of outliers was calculated and found to be 2%. This was below the limit of 10% and hence acceptable.

**EDA Concluding Remarks**

**Key findings**:

1. Successfully handled missing values by imputing with mode for categorical columns and irrelevant features by removing them.
2. There were no duplicate records to remove.
3. It was found from statistical summary that some columns had skewness and outliers.

**Data visualization insights**:

1. Distribution plots revealed that capital gains column is right skewed and capital loss column is left skewed.
2. Pie plots revealed that 75.3% of the claims were legitimate and only 24.7% of the claims were fraudulent.
3. Bar plots indicated more fraudulent claims for higher total claim amount which implied that total claim amount has a significant correlation with fraud reported.
4. Bar plots also showed that the percentage of frauds reported for male claimants is higher than females as it is 26.13% for males but only 23.46% for females.

**Feature engineering**: Created new features “policy\_bind\_day”, “policy\_bind\_month”, “policy\_bind\_year”, “incident\_day”, “incident\_month” from the features “policy\_bind\_date” and “incident\_date” and “csl\_accidental', “csl\_overall” from “policy\_csl”.

**Conclusion**: The EDA provided valuable insights into the dataset, highlighting key patterns and relationships between the features and label. These findings informed the subsequent steps in the project, ensuring an effective approach to building robust models to detect insurance claim fraud.

**Pre-Processing Pipeline**

Data preprocessing is a predominant step in a data science project to yield highly accurate and insightful results. Greater the quality of data, the greater is the reliability of the produced results. Following were the step performed during data pre-processing:

**Encoding categorical columns**: In machine learning, the datasets often contain multiple categorical columns. To make the data human readable, the categories in the categorical columns are created with words. But these columns need to be converted into numeric form to make them understandable by machine learning models. This conversion is done through encoding. In our dataset, columns like “policy\_state”, “insured\_sex”, “insured\_occupation”, “incident\_type”, etc. are categorical columns. These columns were converted into numeric form using Ordinal Encoder. It assigned a specific number to each of the categories and converted them into numeric form.

**Checking and removing skewness**: Skewness is a measure of the asymmetry of a data distribution around its mean in a column. It is important to remove skewness from the data before using for building a machine learning model as skewed data can lead to biased predictions and poor performance by the models. In our dataset, the features "total\_claim\_amount" and "vehicle\_claim" were found to have skewness. We used Yeo-Johnson transformation method to remove skewness from these features.

**Correlation heatmap**: We use correlation matrix and heatmap to find out how much every feature is correlated with every other feature or the label. From the analysis, the following findings were derived:

1. The label "fraud\_reported" did not have significant correlation with any of the features.
2. The features like "age" and "months\_as\_customer", "total\_claim\_amount" and "injury\_claim", etc. had high positive correlation between each other.
3. The features "number\_of\_vehicles\_involved" and "incident\_type", "incident\_month" and "incident\_day" had high negative correlation between each other.
4. The features "vehicle\_claim" and "total\_claim\_amount", "csl\_overall" and "csl\_accidental" had perfect positive correlation between each other.

The bar chart derived from the correlation matrix indicated that the features "insured\_education\_level", "age", "insured\_occupation", "policy\_bind\_year", "auto\_model" and "collision\_type" had very low correlation with the label "fraud\_reported" and did not contribute much in its prediction. So, they were removed from the dataset.

**Separating features and label**: The features and the label need to be separated so that the model can predict the label from the features. The features were separated and stored in a separate dataframe and the label was stored in another dataframe.

**Feature scaling**: Feature scaling is important to bring all the features to a comparable scale. Without scaling, features with larger ranges can dominate the learning process leading to biased models. Scaling is particularly important for algorithms that rely on distance calculations such as k-nearest neighbors (KNN) and support vector machines (SVM). As these algorithms have been used in this project, it was important to use scaling. Standard Scaler also known as Z-score Normalization was used which scales the data to have a mean of 0 and a standard deviation of 1 and all the data was brought into a common scale.

**Checking Variance Inflation Factor (VIF)**: Multicollinearity occurs when two or more independent variables in a model are highly correlated. High multicollinearity can inflate the variance of the coefficient estimates and make the model unstable and difficult to interpret. VIF quantifies how much the variance of a regression coefficient is inflated due to multicollinearity. By checking the VIF values of each feature, those features which have a high VIF value can be identified as causing multicollinearity problem and can be removed. In this project, the VIF values of the features were calculated and the features "total\_claim\_amount", "injury\_claim", "property\_claim", "vehicle\_claim", "csl\_accidental" and "csl\_overall" were found to have VIF values greater than 10 which indicated that these features were causing multicollinearity problem. First, the feature "total\_claim\_amount" which had the highest VIF value was removed and the VIF values were checked again. Then, the features "csl\_accidental" and "csl\_overall" had VIF values greater than 10 and "csl\_accidental" was removed. Finally, when the VIF values were checked, none of the features had VIF values greater than 10 and thus the problem of multicollinearity was solved.

**Balancing the label**: In the dataset, it was found that the label “fraud\_reported” is imbalanced as it contained 739 “0” values but only 241 “1” values. If the label is imbalanced the predictions of the model will be biased. So, to correct this, there was a need to balance the label. This was done through SMOTE (Synthetic Minority Oversampling Technique) algorithm. This algorithm works by randomly picking a point from the minority class and computing the k-nearest neighbors of this point. The synthetic points are added between the chosen point and its neighbors. After applying this algorithm, the label was balanced and it contained 1478 values in the label with 739 values each for “0” and “1”. The number of rows in the features dataframe was also increased to 1478.

**Building Machine Learning Models**

Building machine learning models is the most important aspect of a data science project. The training dataset is first passed into the model and the model is trained. Then the testing dataset is passed and the model predicts the label. The Sklearn module was used for building the models. There are several models present inside the Sklearn module. Basically, there are two types of models – classification and regression. As the label in this dataset is a categorical variable, this is a classification problem. So, classification models were used.

**Finding the best random state**: The dataset is split into training and test datasets for training and testing a model. For split the dataset, a random state variable is passed into the train\_test\_split method. To find the best random state for splitting the dataset, the dataset is split with different random states and a model is trained, tested and accuracy is calculated for each split. For this purpose, Decision Tree classifier model was used and the model was trained and tested for all random states between 1 to 200 and the best random state with the highest accuracy was found out.

**Splitting the dataset**: The dataset is split into training and test subsets for training and testing the models. For this purpose, the train\_test\_split method is used. The train\_test\_split is a function in Sklearn module which splits the features and the label into random partitions. It also takes a random state variable as input. It gives four outputs x\_train, x\_test, y\_train and y\_test. The x\_train and x\_test contain the training and testing features while y\_train and y\_test contain the training and testing labels. The features and the label were split using the train\_test\_split method by passing the best random state which was found in the previous step.

**Training and testing the models**: This is an important aspect of a data science project where different models are trained and tested and the metrics like accuracy, precision, etc. are calculated for all the models to find out the best model. The following models were trained and tested in this project:

1. Decision Tree Classifier
2. Logistic Regression
3. KNeighbors Classifier
4. Support Vector Machine Classifier
5. XGB Classifier
6. Random Forest Classifier
7. Bagging Classifier
8. Gradient Boosting Classifier

All the metrics like accuracy, precision, recall, F1-score and confusion matrix were calculated for each and every model.

**Cross-validation**: Cross-validation is a technique used to validate the performance of a machine learning model. It helps in assessing how well the model generalizes to new and unseen data and prevents overfitting. K-Fold cross-validation was used to validate the performance of all the models. This is a technique wherein the dataset is divided into k equal sized folds. The model is trained on k-1 folds and tested on the remaining fold. This is repeated k times with each fold used as the test once. The mean cross-validation score and the difference between the accuracy score achieved in the previous step and the mean cross-validation score for all the models were calculated.

**Selection of the best model**: After training and testing models and performing cross-validation of the models, the accuracies and cross-validation scores of the models were compared and the following models were considered as good performing models:

1. Gradient Boosting Classifier
2. Bagging Classifier
3. Random Forest Classfier
4. XGB Classifier
5. Decision Tree Classifier

Among these models, Random Forest Classifier model had the least difference between accuracy score and cross validation score. So, it was selected as the best model. The metrics of this model are as below:

1. Accuracy – 89.86%
2. Precision – 90%
3. Recall – 90%
4. F1-score – 90%

**Hyper parameter tuning**: Hyperparameters are settings that control the learning process of a machine learning model. Hyper parameter tuning is the process of finding the optimal hyper parameters of a machine learning algorithm that deliver the best performance as measured on a validation set. Hyper parameter tuning for Random Forest classifier model which was selected as the best model was done using RandomizedSearchCV which randomly samples hyperparameter combinations from a specified distribution. The best hyper parameters were found out from this process and the final model was built using these parameters. The accuracy achieved by the final model was 88.74%.

**Computing and plotting ROC curve and AUC for the final model**: ROC stands for Receiver Operating Characteristic. The ROC curve is a graphical representation that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is changed. The True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings are plotted to draw the ROC curve. AUC stands for Area Under the ROC Curve. It tells how much the model is capable of distinguishing between different classes. The ROC curve was plotted for this project and the AUC score was calculated as 89% which is a very good score.

**Saving the model**: The model needs to be saved because only then it can be used in the future and can be deployed in a production environment. The model was saved using the Pickle module.

**Concluding Remarks**

The project has successfully developed a machine learning model to detect fraudulent auto insurance claims. The challenge was that fraudulent claims were far less as compared to legitimate insurance claims. The Exploratory Data Analysis provided valuable insights into the patterns and anomalies in the data. It revealed that the number of fraudulent claims were higher for higher total claim amounts and lower for lower total claim amounts. Many different classification models like Decision Tree classifier, Gradient Boosting classifier, etc. were built during the project. Random Forest classifier was selected as the best and final model. It performed excellently and gave an impressive accuracy of 88.74% and an ROC AUC of 89%. In conclusion, the model was able to correctly distinguish between fraudulent and legitimate insurance claims with high accuracy. By accurately detecting fraudulent claims, the model can help the insurance company save millions of dollars annually and improve the efficiency of the insurance claims processing system.