# CAR INSURANCE

*A report submitted in partial fulfillment of the requirements for the Award of Degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**INFORMATION TECHNOLOGY**

**By**

**PACHIGOLLA UDAY SRI SAI MANIKANTA**
**Regd. No.: 20B91A12E0**

**Under Supervision of Mr. Gundala Nagaraju**
**Henotic Technology Pvt Ltd, Hyderabad**
**(Duration: 7th July, 2022 to 6th September, 2022)**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**SAGI RAMA KRISHNAM RAJUENGINEERING COLLEGE**

(An Autonomous Institution)

Approved by AICTE, NEW DELHI and Affiliated to JNTUK, Kakinada

CHINNA AMIRAM, BHIMAVARAM,
ANDHRA PRADESH

# Table of Contents

# Abstract

The insurance companies are tremendously interested in the prediction of the future. Accurate prediction gives a probability to decrease financial loss for the company. The insurers use rather complex methodologies for this purpose.

The major models are a decision tree, a random forest, a binary logistic regression, and a support vector machine. A great number of different variables are under analysis in this case.

The algorithms involve detection of relations between claims, implementation of high dimensionality to reach all the levels, detection of the missing observations, etc. In this way, the individual customer’s portfolio is made.

Forecasting the upcoming claims helps to charge competitive premiums that are not too high and not too low. It also contributes to the improvement of the pricing models. This helps the insurance company to be one step ahead of its competitor.

# 1.0 Introduction

With the increasing power of computer technology, companies and institutions can nowadays store large amounts of data at reduced cost. The amount of available data is increasing exponentially and cheap disk storage makes it easy to store data that previously was thrown away. There is a huge amount of information locked up in databases that is potentially important but has not yet been explored. The growing size and complexity of the databases makes it hard to analyse the data manually, so it is important to have automated systems to support the process. Hence there is the need of computational tools able to treat these large amounts of data and extract valuable information.

In this context, Data Mining provides automated systems capable of processing large amounts of data that are already present in databases. Data Mining is used to automatically extract important patterns and trends from databases seeking regular patterns that can reveal the structure of the data and answer business problems. Data Mining includes learning techniques that fall into the field of Machine learning. The growth of databases in recent years brings data mining at the forefront of new business technologies.

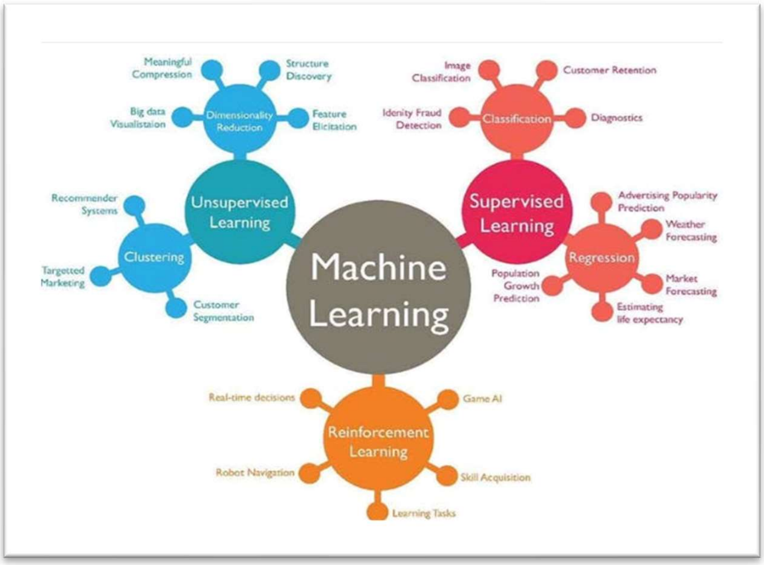
A key challenge for the insurance industry is to charge each customer an appropriate price for the risk they represent. Risk varies widely from customer to customer and a deep understanding of different risk factors helps predict the likelihood and cost of insurance claims. The goal of this program is to see how well various statistical methods perform in predicting auto Insurance claims based on the characteristics of the driver, vehicle and driver / vehicle coverage details.

A number of factors will determine claims prediction among them a age, past accident history, and domicile, etc. However, this contest focused on the relationship between claims and vehicle characteristics well as other characteristics associated with the auto insurance policies.

## 1.1. What are the different types of Machine Learning?

Different types of Machine Learning :-

1. Supervised Learning.
2. Unsupervised Learning.
3. Reinforcement Learning.



## 1. Supervised Learning :-

In supervised learning, we are given a data set and **already know what our correct output** should look like, having the idea that there is a relationship between the input and output.

### Two types of Supervised Learning :-

1. **Regression**—Estimate continuous values (Real valued output)
2. **Classification**—Identify a unique class (Discrete values, Boolean or Categories)

## 1.1 Regression :-

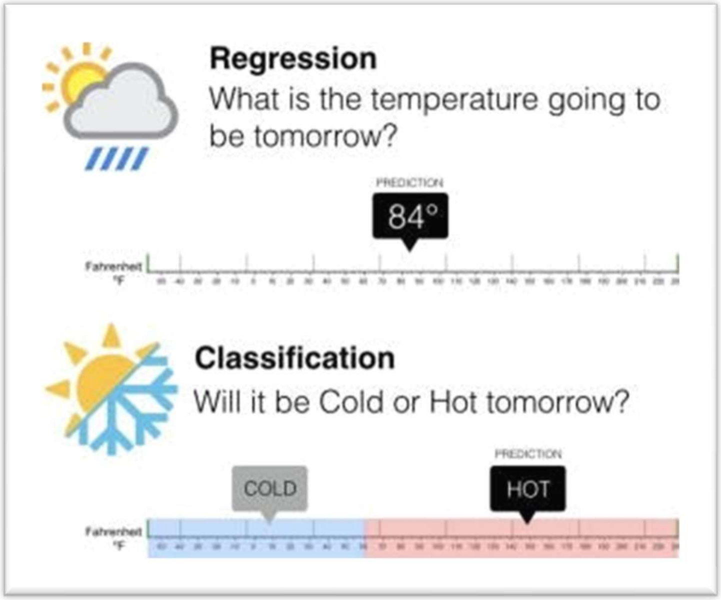
**Regression** models a target prediction value based on independent variables. It is mostly used for finding out the **relationship between variables** and **forecasting**. Regression can be used to estimate/ predict **continuous values** (Real valued output).

For example **:** Given a picture of a person, we have to predict the age on the basis of the given picture .

## 1.2 Classification :-

**Classification** means to **group** the output into a class. If the data set is **discrete** or **categorical** then it is a classification problem.

For example **:** Given data about the sizes of houses in the real estate market, making our output about whether the house "sells for **more** or **less** than the asking price" i.e. Classifying houses into two discrete categories.



## 2. Unsupervised Learning :-

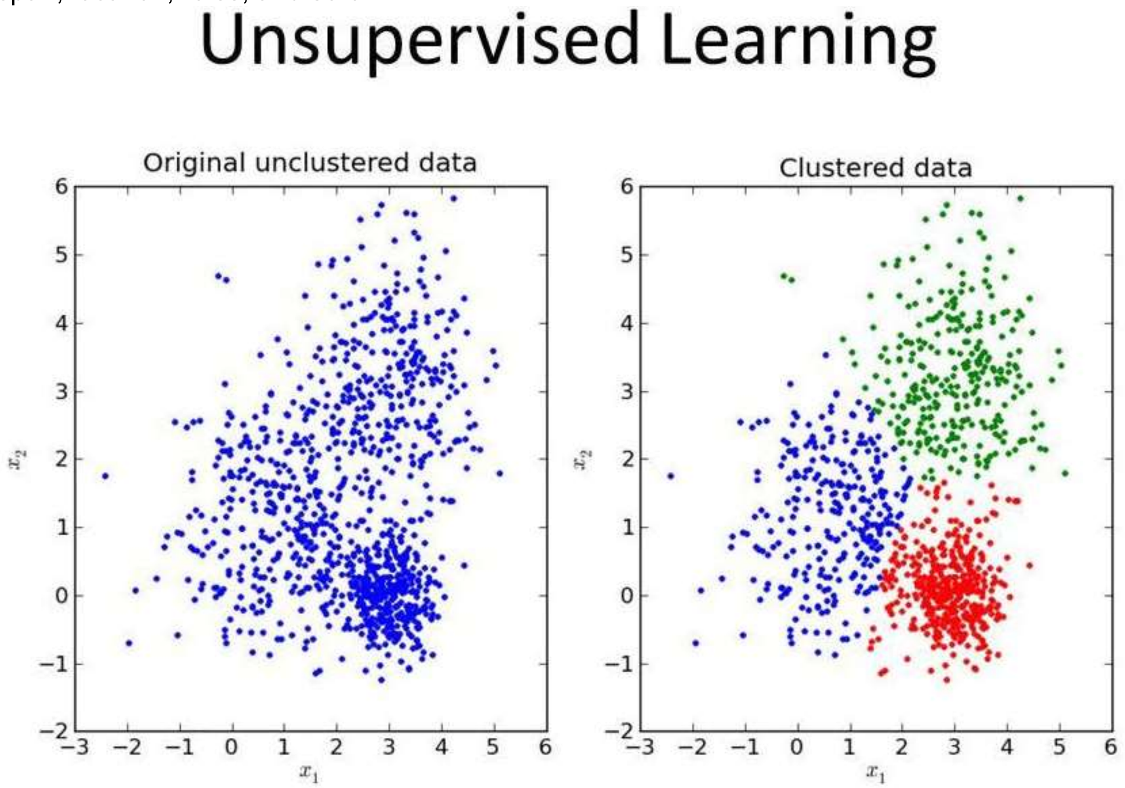
It allows us to approach problems with little or no idea about what our results look like. We can **derive structure** from data where we don't necessarily know the effect of the variables.

We can derive this structure by **clustering** the data based on relationships among the variables in the data.

## 2.1. Clustering :-

C**lustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (clusters).

For example : Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into **groups that are somehow similar** or related by different variables, such as lifespan, location, roles, and so on.



## 3. Reinforcement Learning :-

Reinforcement Learning is about taking suitable actions to **maximize reward** in a particular situation. It is employed by various software and machines to find the **best possible behavior** or path to take in a specific situation.

Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it, so the model is trained with the correct answer itself whereas in reinforcement learning, **there is no answer** and the **reinforcement agent** decides what to do in order to perform the given task. In the absence of training data set, it is bound to **learn from its experience**.

## 1.2. Benefits of Using Machine Learning

The resistance to machine learning surrounded by the industry is multi-faceted –starting from the underwriters and the information technology teams to operational management. A variety of reasons contain the following steps:

- Loss of jobs
- Focus on current issues like building the scalable foundation for their current journey
- Non-availability of resources and dearth of required skill-set
- Financial constraints in form of funding the development/POC/research stages
- Insurers want clear evidence of success before adoption
- Regulatory restrictions and privacy norms
- Large diversity of training for real-world operation and simulation of various scenarios
- Initiation challenges
- Infrastructural challenges faced by a neural network designer include filling millions of database rows for its connections.

**Figure:** This figure illustrates the improvement of the machine learning market in diverse geological regions over 10 years. It shows the accelerating approval of Artificial intelligence (AI) and the critical significance of technology trend. Global AI market, by geography 2017–2024 (in US$ M).

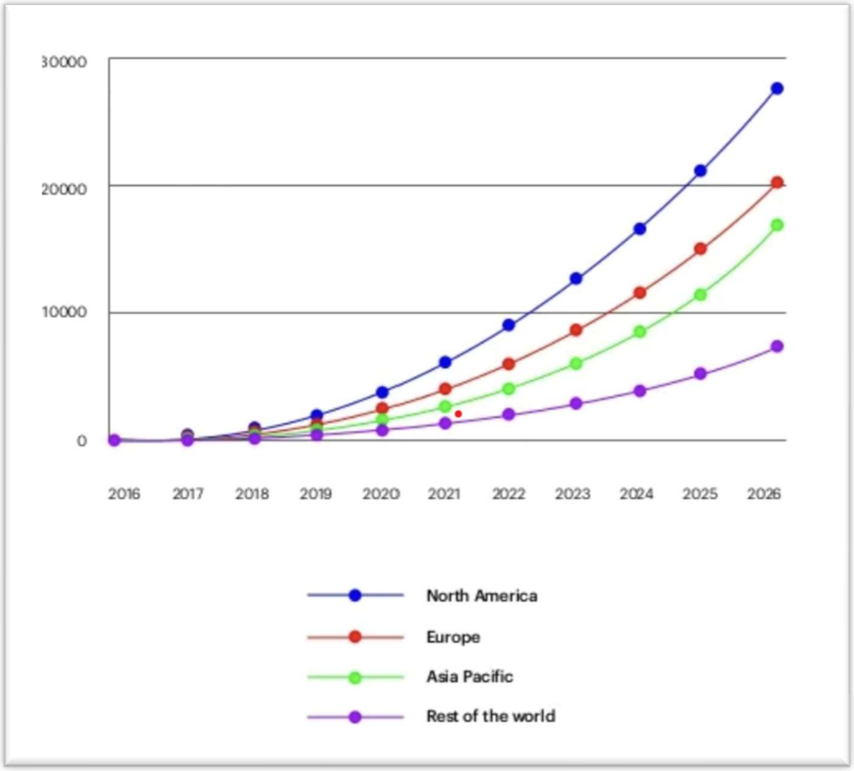


Fig1: Machine learning for insurance business Areas

## 1.3. About Industry (example car insurance)

The vehicle insurance industry has been going through massive transformations in the past couple of years. With more emphasis on customized insurance plans and the increasing level of market competition. As of 2018, the vehicle insurance industry has reached the $200 billion mark. And this means that now there is no room for those auto insurance organizations that are not serious about their revered clientele. With a new competent organization budding every other week, the market competition is getting more cut-throat than ever.

### 1.3.1 AI / ML Role in Car Insurance

Machine Learning is a sub-set of artificial intelligence where computer algorithms are used to autonomously learn from data. Machine learning (ML) is getting more and more attention and is becoming increasingly popular in many other industries. Within the insurance industry, there is more application of ML regarding the claims.

## 2.0 Car Insurance Claims

Car insurance premiums have historically been priced on underwriting and rating. Underwriting is a process where the insurer assesses the applicant’s risk. They do this by incorporating personal information and internal claims data into weighted algorithms. Insurers then look at rating factors to predict the likelihood of a claim’s submission. The rating assigns a price based on the projected cost to the insurer of assuming financial responsibility of potential claims. Car insurance premiums fluctuate with the projected risk to the insurer. A policyholder can lower their premium by taking on more risk. For instance, the policyholder can choose to drop optional coverages or increase the deductible. A deductible is the out-of-pocket portion of the claim for which the driver is responsible.
The main factors for car insurance claims are Location, Age, Gender, Marital status, driving experience, driving record, Claims history, Credit history, Previous insurance coverage, Vehicle type, Vehicle use, Miles driven ,etc.

### 2.1. Main Drivers for AI Auto Quote Analysis

Predictive modelling allows for simultaneous consideration of many variables and quantification of their overall effect. When a large number of claims are analysed, patterns regarding the characteristics of the claims that drive loss development begin to emerge.

The following are the main drivers which influencing the Claims Analytics:

- **Policy Characteristics**
    - ✓ Exposures
    - ✓ Limits and Deductibles
    - ✓ Coverages and Perils
- **Insured Characteristics**
    - ✓ Credit Information
    - ✓ Prior loss experience
    - ✓ Payment history
- **Geography based on insured locations**
    - ✓ Auto Repair Costs
    - ✓ Jurisdictional Orientation
    - ✓ Demographics
    - ✓ Crime
- **Agency Characteristics**
    - ✓ Exclusive Agents
    - ✓ Independent Agents
- **Claim information**
    - ✓ FNOL
    - ✓ Claimant data (Credit info, geography, social data, etc.)
    - ✓ Other participants (insured, doctors, lawyers, witnesses, etc.)
    - ✓ Cause, type of Injury/Damage
    - ✓ Injury or damaged object
    - ✓ Coverage
    - ✓ Loss Location
    - ✓ Date and time of Loss and Report
    - ✓ Weather at time & location of loss
- **Details from Prior Claims**
    - ✓ from same insured
    - ✓ from same claimant
    - ✓ from same location
- **Household Characteristics**

### 2.2. Internship Project - Data Link

The internship project data has taken from Kaggle and the link is https://www.kaggle.com/code/yugandharshilawane/car-insurance-data

# 3.0 AI / ML Modelling and Results

## 3.1. Your Problem of Statement

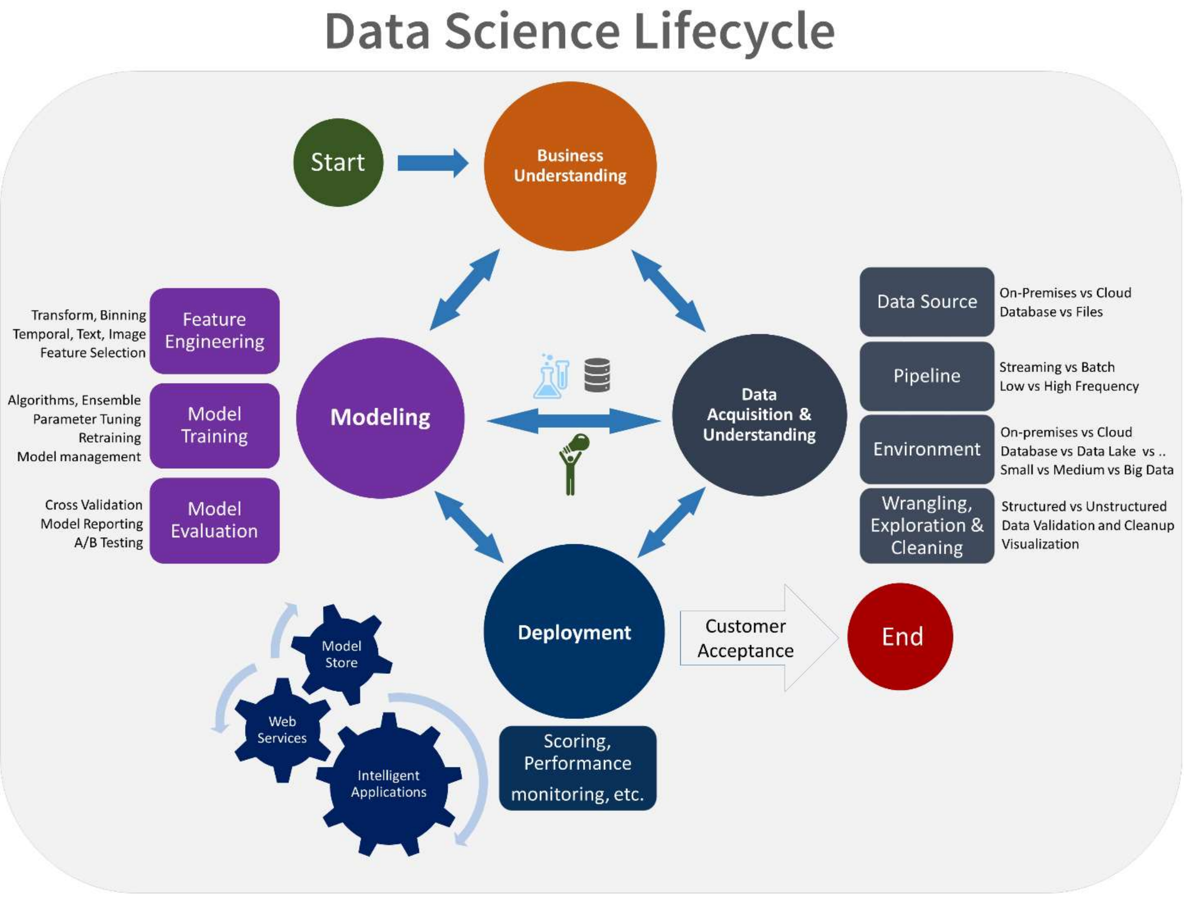
Predictive models are most effective when they are constructed using a company's own historical claims data since this allows the model to recognize the specific nature of a company's exposure as well as its claims practices. The construction of the model also involves input from the company throughout the process, as well as consideration of industry leading claims practices and benchmarks.

Predictive modelling can be used to quantify the impact to the claims department resulting from the failure to meet or exceed claim service leading practices. It can also be used to identify the root cause of claim leakage. Proper use of predictive modelling will allow for potential savings across two dimensions:

- Early identification of claims with the potential for high leakage, thereby allowing for the proactive management of the claim
- Recognition of practices that are unnecessarily increasing claims settlement payments

## 3.2. Data Science Project Life Cycle

Data Science is a multidisciplinary field of study that combines programming skills, domain expertise and knowledge of statistics and mathematics to extract useful insights and knowledge from data.



### 3.2.1 Data Exploratory Analysis

Exploratory data analysis has been done on the data to look for relationship and correlation between different variables and to understand how they impact or target variable.
The exploratory analysis is done for Auto Quote / Policy Conversion with different parameters and all the charts are presented in **Appendices 5.1 - List of charts (5.1.1 to 5.1.2)**

### 3.2.2 Data Pre-processing

We removed variables which does not affect our target variable (Outcome) as they may add noise and also increase our computation time ,we checked the data for anomalous data points and outliers .We did principal component analysis on the data set to filter out unnecessary variables and to select only the important variables which have greater correlation with our target variable.

#### 3.2.2.1. Check the Duplicate and low variation data

- Duplicate observations occur **when two or more rows have the same values or nearly the same values**.
- A duplicate value is **one in which all values in at least one row are identical to all of the values in another row**. A comparison of duplicate values depends on the what appears in the cell not the underlying value stored in the cell.
- Use the **duplicated().any()** function to know whether the dataset contains duplicates or not.
- It returns **True** if the dataset contains duplicates otherwise False.
- If it is True then drop all duplicate records from the dataset by using the function drop_duplicates(). Then we can see that the number of records has dropped

#### 3.2.2.2. Identify and address the missing variables

The cause of the presence of missing values in the dataset can be **loss of information, disagreement in uploading the data**, and many more. Missing values need to be imputed to proceed to the next step of the model development pipeline.

The missing data model was similar to other features in the data set, but beyond that, the missing data values are not random. If the data is missing in the variable considered then it is said to be missing not at random(MNAR).

We handle missing data in a dataset by
**Imputing the Missing Value**

1. Replacing With Arbitrary Value.
2. Replacing With Mode.
3. Replacing With Median.
4. Replacing with previous value – Forward fill.
5. Replacing with next value – Backward fill.
6. Interpolation.
7. Impute the Most Frequent Value.

### 3.2.2.3. Handling of Outliers

An outlier may occur due to the variability in the data, or due to experimental error/human error.
They may indicate an experimental error or heavy skewness in the data (heavy-tailed distribution).

If our dataset is small, we can detect the outlier by just looking at the dataset. But what if we have a huge dataset, how do we identify the outliers then? We need to use visualization and mathematical techniques.

Below are some of the techniques of detecting outliers

- Boxplots
- Z-score
- Inter Quantile Range (IQR)

### 3.2.2.4. Categorical data and Encoding Techniques

Machine learning models require all input and output variables to be numeric .This means that if your data contains categorical data, you must encode it to numbers before you can fit and evaluate a model.

The two most popular techniques are an **Ordinal Encoding** and a **One-Hot Encoding**.

- Encoding is a required pre-processing step when working with categorical data for machine learning algorithms.
- How to use ordinal encoding for categorical variables that have a natural rank ordering.
- How to use one-hot encoding for categorical variables that do not have a natural rank ordering.

### 3.2.2.5. Feature Scaling

If the data in any conditions has data points far from each other, scaling is **a technique to make them closer to each other** or in simpler words, we can say that the scaling is used for making data points generalized so that the distance between them will be lower.

We used minmax scaling technique here Min-Max scaling is **a normalization technique that enables us to scale data in a dataset to a specific range using each feature's minimum and maximum value**.

A Min-Max scaling is typically done via the following equation: **Xsc =X−XminXmax−Xmin**. One family of algorithms that is scale-invariant encompasses tree-based learning algorithms.

### 3.2.3 Selection of Dependent and Independent variables

The dependent or target variable here is Outcome Target which tells us a particular policy holder has filed a claim or not the target variable is selected based on our business problem and what we are trying to predict.

The independent variables are selected after doing exploratory data analysis and we used Boruta to select which variables are most affecting our target variable.

### 3.2.4 Data Sampling Methods

The data we have is highly unbalanced data so we used some sampling methods which are used to balance the target variable so we our model will be developed with good accuracy and precision. We used three Sampling methods

#### 3.2.4.1. Stratified sampling

Stratified sampling randomly selects data points from majority class so they will be equal to the data points in the minority class. So, after the sampling both the class will have same no of observations.
It can be performed using strata function from the library sampling.

#### 3.2.4.2. Simple random sampling

Simple random sampling is a sampling technique where a set percentage of the data is selected randomly. It is generally done to reduce bias in the dataset which can occur if data is selected manually without randomizing the dataset.
We used this method to split the dataset into train dataset which contains 70% of the total data and test dataset with the remaining 30% of the data.

### 3.2.5 Models Used for Development

We built our predictive models by using the following ten algorithms

#### 3.2.5.1. Model 01-LOGISTIC REGRESSION

Logistic uses logit link function to convert the likelihood values to probabilities so we can get a good estimate on the probability of a particular observation to be positive class or negative class .The also gives us p-value of the variables which tells us about significance of each independent variable.

#### 3.2.5.2. Model 02-DECISION TREE

- Decision Tree is a supervised learningDecision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**

- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.

#### 3.2.5.3. Model 03-RANDOM FOREST

Random forest is an algorithm that consists of many decision trees. It was first developed by Leo Breiman and Adele Cutler. The idea behind it is to build several trees, to have the instance classified by each tree, and to give a "vote" at each class .The model uses a "bagging" approach and the random selection of features to build a collection of decision trees with controlled variance. The instance's class is to the class with the highest number of votes, the class that occurs the most within the leaf in which the instance is placed.
The error of the forest depends on:

- Trees correlation: the higher the correlation, the higher the forest error rate.
- The strength of each tree in the forest. A strong tree is a tree with low error. By using trees that classify the instances with low error the error rate of the forest decreases.

#### 3.2.5.4. Model 04-EXTRA TREES CLASSIFIER

**Extremely Randomized Trees Classifier(Extra Trees Classifier)** is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a “forest” to output it’s classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest.

Each Decision Tree in the Extra Trees Forest is constructed from the original training sample. Then, at each test node, Each tree is provided with a random sample of k features from the feature-set from which each decision tree must select the best feature to split the data based on some mathematical criteria (typically the Gini Index). This random sample of features leads to the creation of multiple de-correlated decision trees

#### 3.2.5.5. Model 05-KNN CLASSIFIER

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a **non-parametric, supervised learning classifier**, which uses proximity to make classifications or predictions about the grouping of an individual data point.

KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

**KNN is a supervised learning algorithm used for classification**. KNN is a classification algorithm which falls under the greedy techniques however k-means is a clustering algorithm (unsupervised machine learning technique).

#### 3.2.5.6. Model 06-GAUSSIAN NAÏVE BAYES

Gaussian Naive Bayes is **a probabilistic classification algorithm based on applying Bayes' theorem with strong independence assumptions**.

Gaussian Naive Bayes **supports continuous valued features and models each as conforming to a Gaussian (normal) distribution**. An approach to create a simple model is to assume that the data is described by a Gaussian distribution with no co-variance (independent dimensions) between dimensions.

#### 3.2.5.7. Model 07-XGB CLASSIFIER

XGBoost, which stands for Extreme Gradient Boosting, is **a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library**. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.

**It uses more accurate approximations to find the best tree model**. Boosting: N new training data sets are formed by random sampling with replacement from the original dataset, during which some observations may be repeated in each new training data set.

#### 3.2.5.8. Model 08-SVM

A support vector machine (SVM) is **a type of deep learning algorithm that performs supervised learning for classification or regression of data groups**. In AI and machine learning, supervised learning systems provide both input and desired output data, which are labeled for classification.

Support vector machines (SVMs) are a set of supervised learning methods used for **classification, regression and outliers detection**. The advantages of support vector machines are: Effective in high dimensional spaces. Still effective in cases where number of dimensions is greater than the number of samples.

#### 3.2.5.9. Model 09-LIGHT GBM

Light GBM, short for Light Gradient Boosting Machine, is **a free and open source distributed gradient boosting framework for machine learning originally developed by Microsoft**. It is based on decision tree algorithms and used for ranking, classification and other machine learning tasks.

Light GBM is **a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm, used for ranking, classification and many other machine learning tasks**

## 3.3. AI / ML Models Analysis and Final Results

We used our train dataset to build the above models and used our test data to check the accuracy and performance of our models.

We used confusion matrix to check accuracy, Precision, Recall and F1 score of our models and compare and select the best model for given dataset of size ~11000 policies.

### 3.3.1 Different Model codes

- The Python code for models with simple random sampling technique as follows:

```
# Build the Calssification models and compare the results

from sklearn.linear_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import ExtraTreesClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.naive_bayes import GaussianNB

from sklearn.svm import SVC

from sklearn.ensemble import BaggingClassifier

from sklearn.ensemble import GradientBoostingClassifier

import lightgbm as lgb

# Create objects of classification algorithm with default hyper-parameters

ModelLR = LogisticRegression()

ModelDC = DecisionTreeClassifier()

ModelRF = RandomForestClassifier()

ModelET = ExtraTreesClassifier()
ModelKNN = KNeighborsClassifier(n_neighbors=5)

ModelSVM = SVC(probability=True)

modelBAG = BaggingClassifier(base_estimator=None, n_estimators=100,
max_samples=1.0, max_features=1.0, bootstrap=True, bootstrap_features=False,
oob_score=False, warm_start=False, n_jobs=None, random_state=None,
verbose=0)


ModelGB = GradientBoostingClassifier(loss='deviance',
learning_rate=0.1,n_estimators=100, subsample=1.0, criterion='friedman_mse',
min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0,
max_depth=3, min_impurity_decrease=0.0, init=None,random_state=None,
max_features=None, verbose=0, max_leaf_nodes=None, warm_start=False,
validation_fraction=0.1, n_iter_no_change=None, tol=0.0001, ccp_alpha=0.0)
```

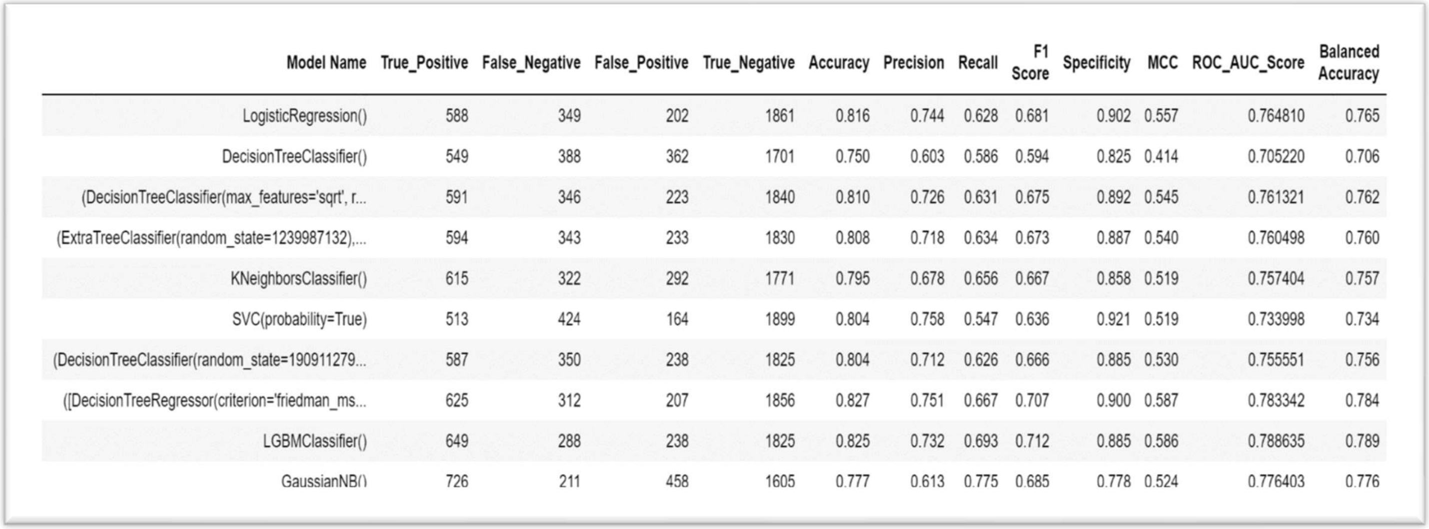
```
ModelLGB = lgb.LGBMClassifier()

ModelGNB = GaussianNB()

# Evalution matrix for all the algorithms

MM = [ModelLR, ModelDC, ModelRF, ModelET, ModelKNN, ModelSVM,
modelBAG, ModelGB, ModelLGB, ModelGNB]

for models in MM:
    # Fit the model
    models.fit(x_train, y_train)
    # Prediction

    y_pred = models.predict(x_test)
    y_pred_prob = models.predict_proba(x_test)

    # Print the model name

    print('Model Name: ', models)

    # confusion matrix in sklearn
    from sklearn.metrics import confusion_matrix
    from sklearn.metrics import classification_report
    # actual values
    actual = y_test
    # predicted values
    predicted = y_pred
    # confusion matrix
    matrix = confusion_matrix(actual,predicted, labels=[1,0],sample_weight=None,
normalize=None)
    print('Confusion matrix : \n', matrix)
     # outcome values order in sklearn

    tp, fn, fp, tn = confusion_matrix(actual,predicted,labels=[1,0]).reshape(-1)
    print('Outcome values : \n', tp, fn, fp, tn)
    # classification report for precision, recall f1-score and accuracy
    C_Report = classification_report(actual,predicted,labels=[1,0])
    print('Classification report : \n', C_Report)
     # calculating the metrics
    sensitivity = round(tp/(tp+fn), 3);
    specificity = round(tn/(tn+fp), 3);
    accuracy = round((tp+tn)/(tp+fp+tn+fn), 3);
    balanced_accuracy = round((sensitivity+specificity)/2, 3);

    precision = round(tp/(tp+fp), 3);
    f1Score = round((2*tp/(2*tp + fp + fn)), 3);
     # Matthews Correlation Coefficient (MCC). Range of values of MCC lie between
-1 to +1.
     # A model with a score of +1 is a perfect model and -1 is a poor model
    from math import sqrt
    mx = (tp+fp) * (tp+fn) * (tn+fp) * (tn+fn)
    MCC = round(((tp * tn) - (fp * fn)) / sqrt(mx), 3)
    print('Accuracy :', round(accuracy*100, 2),'%')
    print('Precision :', round(precision*100, 2),'%')
```

```
  print('Recall :', round(sensitivity*100,2), '%')
  print('F1 Score :', f1Score)
  print('Specificity or True Negative Rate :', round(specificity*100,2), '%' )
  print('Balanced Accuracy :', round(balanced_accuracy*100, 2),'%')
  print('MCC :', MCC)
   # Area under ROC curve
  from sklearn.metrics import roc_curve, roc_auc_score
  print('roc_auc_score:', round(roc_auc_score(actual, predicted), 3))

  # ROC Curve

  from sklearn.metrics import roc_auc_score
  from sklearn.metrics import roc_curve
  logit_roc_auc = roc_auc_score(actual, predicted)
  fpr, tpr, thresholds = roc_curve(actual, models.predict_proba(x_test)[:,1])
  plt.figure()
  # plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc)
  plt.plot(fpr, tpr, label= 'Classification Model' % logit_roc_auc)
  plt.plot([0, 1], [0, 1],'r--')
  plt.xlim([0.0, 1.0])
  plt.ylim([0.0, 1.05])
  plt.xlabel('False Positive Rate')
  plt.ylabel('True Positive Rate')
  plt.title('Receiver operating characteristic')
  plt.legend(loc="lower right")
  plt.savefig('Log_ROC')
  plt.show()
  print('------------------------------------------------------------------------------------------------')
  #--------------------------------------------------------------------------------------------------------


  new_row = {'Model Name' : models,
          'True_Positive' : tp,
          'False_Negative' : fn,
          'False_Positive' : fp,
          'True_Negative' : tn,
          'Accuracy' : accuracy,
          'Precision' : precision,
          'Recall' : sensitivity,
          'F1 Score' : f1Score,
          'Specificity' : specificity,
          'MCC':MCC,
          'ROC_AUC_Score':roc_auc_score(actual, predicted),
          'Balanced Accuracy':balanced_accuracy}
  EMResults = EMResults.append(new_row, ignore_index=True)
#----------------------------------------------------------------------------------------------------------------
```

# 4.0 Conclusions and Future work

The model results in the following order by considering the model accuracy, F1 score and RoC AUC score.

1) **LIGHT GBM** with Random Sampling
2) **GRADIENT BOOSTING** with Simple Random Sampling
3) **RANDOM FOREST** with Simple Random Sampling



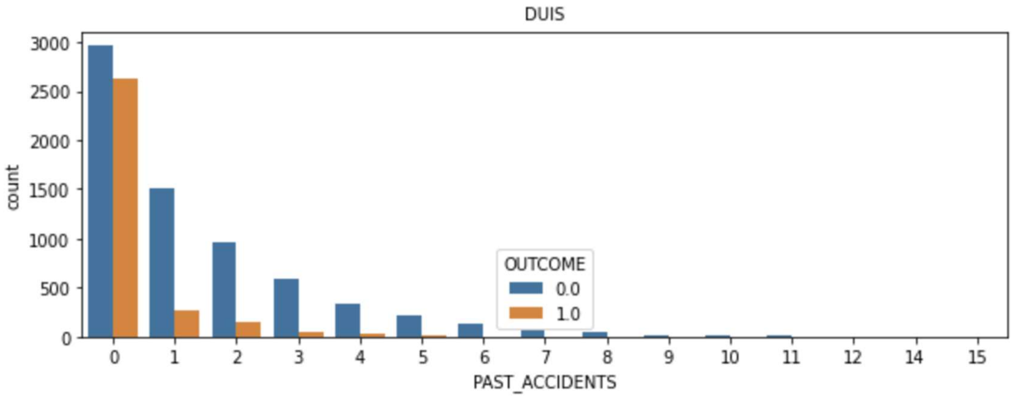
| Model Name | True_Positive | False_Negative | False_Positive | True_Negative | Accuracy | Precision | Recall | F1 Score | Specificity | MCC | ROC_AUC_Score | Balanced Accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LogisticRegression() | 588 | 349 | 202 | 1861 | 0.816 | 0.744 | 0.628 | 0.681 | 0.902 | 0.557 | 0.764810 | 0.765 |
| DecisionTreeClassifier() | 549 | 388 | 362 | 1701 | 0.750 | 0.603 | 0.586 | 0.594 | 0.825 | 0.414 | 0.705220 | 0.706 |
| (DecisionTreeClassifier(max_features='sqrt', r... | 591 | 346 | 223 | 1840 | 0.810 | 0.726 | 0.631 | 0.675 | 0.892 | 0.545 | 0.761321 | 0.762 |
| (ExtraTreeClassifier(random_state=1239987132),... | 594 | 343 | 233 | 1830 | 0.808 | 0.718 | 0.634 | 0.673 | 0.887 | 0.540 | 0.760498 | 0.760 |
| KNeighborsClassifier() | 615 | 322 | 292 | 1771 | 0.795 | 0.678 | 0.656 | 0.667 | 0.858 | 0.519 | 0.757404 | 0.757 |
| SVC(probability=True) | 513 | 424 | 164 | 1899 | 0.804 | 0.758 | 0.547 | 0.636 | 0.921 | 0.519 | 0.733998 | 0.734 |
| (DecisionTreeClassifier(random_state=190911279... | 587 | 350 | 238 | 1825 | 0.804 | 0.712 | 0.626 | 0.666 | 0.885 | 0.530 | 0.755551 | 0.756 |
| ([DecisionTreeRegressor(criterion='friedman_ms... | 625 | 312 | 207 | 1856 | 0.827 | 0.751 | 0.667 | 0.707 | 0.900 | 0.587 | 0.783342 | 0.784 |
| LGBMClassifier() | 649 | 288 | 238 | 1825 | 0.825 | 0.732 | 0.693 | 0.712 | 0.885 | 0.586 | 0.788635 | 0.789 |
| GaussianNB() | 726 | 211 | 458 | 1605 | 0.777 | 0.613 | 0.775 | 0.685 | 0.778 | 0.524 | 0.776403 | 0.776 |

We recommend model – **LIGHT GBM** with Random Sampling technique as a best fit for the given dataset.

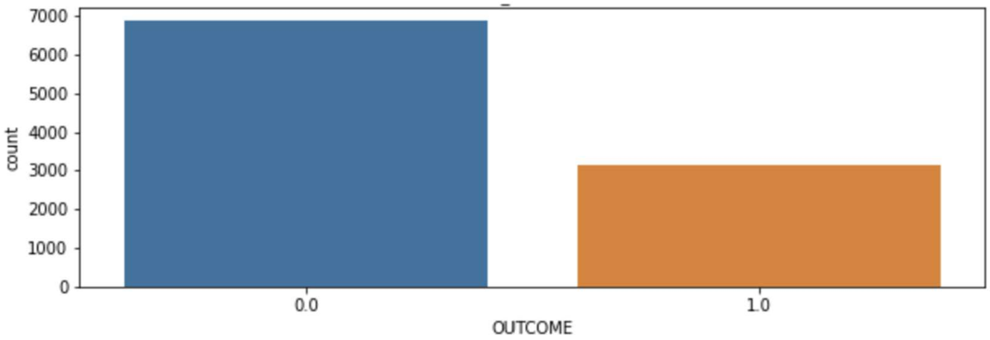
# 5.0 Appendices

## 5.1. List of Charts

### 5.1.1 Chart 01: Total Past Accidents



### 5.1.2 Chart 02: Total Outcome


count
OUTCOME