CMT307-Applied Machine Learning

Implementation and Evaluation of a Case Study Using Machine Learning Techniques

INTRODUCTION: -

Machine learning is a subfield of computer science that deals with the development of algorithms and techniques that allow computers to learn and make predictions based on data. It is used in a variety of applications, including pattern recognition, artificial intelligence, and data analytics. In machine learning, the goal is often to learn a function from past data to make predictions or decisions. This is called as supervised learning or classification. There are many different methods to implement machine learning like decision making, clustering, classification, forecasting, deep learning, support vector machines, reinforcement learning etc. As the size of data grows, the need for computational power to process it also increases, making machine learning a key tool in today's data-driven world.

The goal of our assignment is that "A health insurance provider is looking to expand into car insurance". our task is to create machine learning model to predict **result** whether a customer will buy car insurance based on the variables.

The following are the steps involved in the topic we'll be discussing:

1. Data Pre-processing.

2. Exploratory Data Analysis.

3. Model Implementation.

4. Performance Evaluation.

5. Results and Discussions.

DATA PRE-PROCESSING AND EXPLORATORY DATA ANALYSIS

The data set has been provided to us and based on the information from the column names and associated data values, the dataset contains categorical and numerical features as follows.

Categorical features

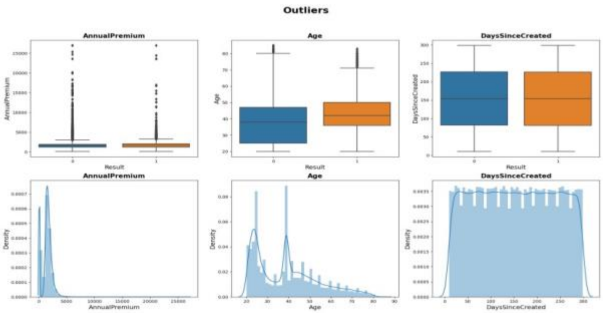
- Gender
- HasDrivingLicense
- RegionID
- Switch
- VehicleAge
- PastAccident
- SalesChannelID

Numerical features

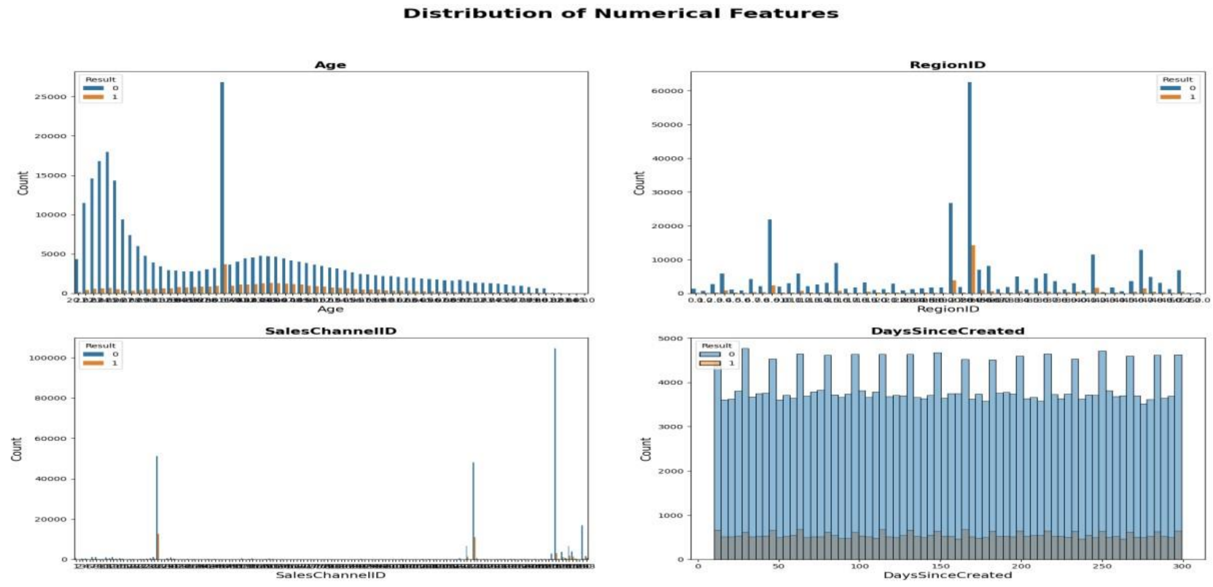
- Age
- AnnualPremium
- DaysSinceCreated

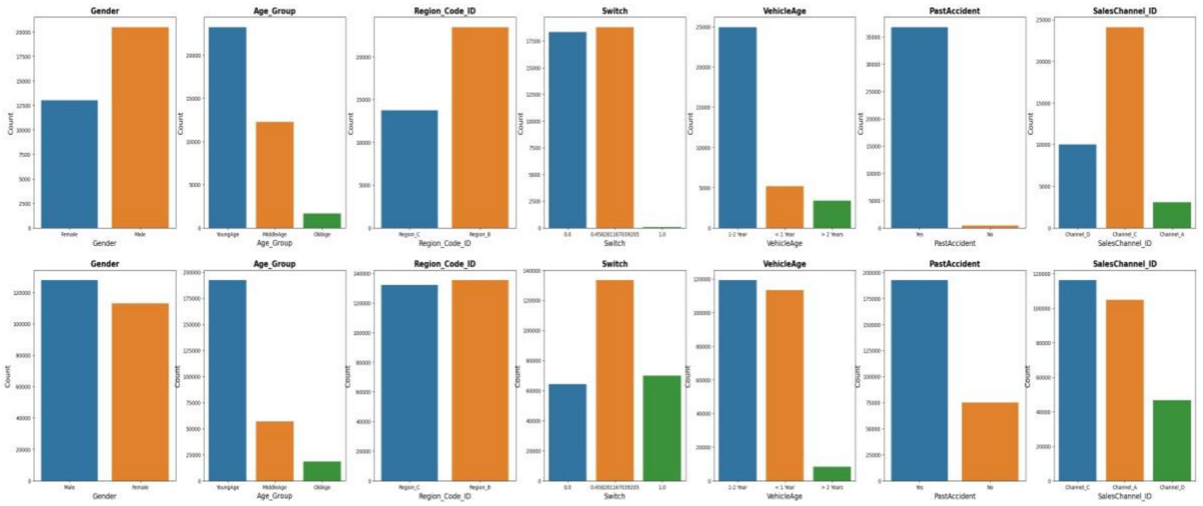
As the preliminary step we import all the required libraries to carry out the project and start out basic analyses of the data set. This step gives us a clear idea of the kind of dataset we are working with. After importing all the required modules, we find out the basic insights regarding the dataset viz., shape, size, description of the data-frame, data types of the columns used, dropping Null/NA then we store it in a Pandas DataFrame called data_df. Next, the code checks for duplicated rows using the duplicated() method and stores the resulting Boolean values in a new DataFrame. Then the code converts all the columns into required format and then handles missing values in the 'Switch', 'PastAccident', 'Age', 'HasDrivingLicense', and 'RegionID' columns. For each column, it fills the missing values with the mean value for that column using the fillna() method. Finally, the code uses the isna().sum() method to sum the number of null values in each column and returns the result. Then we check the outliers and plot it as below.

- From the plot it can be implied that Annual Premium has a positively skewed distribution. we can also depict that DaysSinceCreated has a approximately uniform distribution. Age columns has some outliers, but we are not going to treat them because it won't be affecting our result.

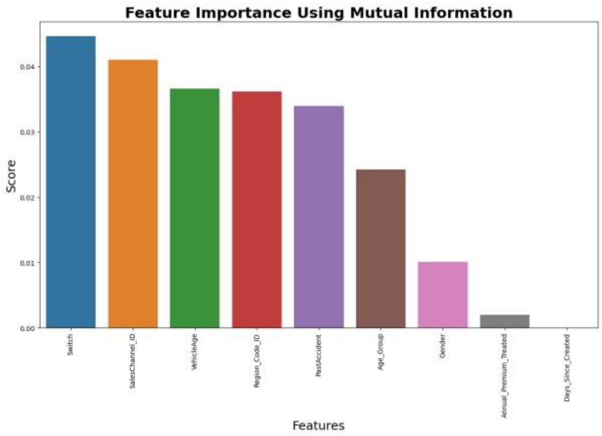


We also see the distribution of numerical features and categorical features.





Then we build a function to visualize the important features using the mutual information available.



switch is the most important feature.

<u>Model Implementation and Performance evaluation.</u>

The data has 50% missing values in switch and PastAccident which was handled earlier in the data pre-processing stage, MICE imputation was used for Past Accident and switch because of the huge number of missing values. Then we use SMOTE (Synthetic Minority Oversample Technique) which will help us in balancing the data.

We applied different Machine Learning Models to our data set and see how each of them performs. Firstly, we will tune the hyper-parameters of those models and then we will compare and choose the best model among them, based on Elapsed Time and Evaluation Metrics of the best parameters.

List of Machine Learning Models we used to train and evaluate our data set on:

- Decision Tree
- Gaussian Naive Bayes
- AdaBoost Classifier
- Bagging Classifier
- Logistic Regression
- KNN
- Random Forest
- XGBoost

Hyper-Parameter Tuning Methods:

We have tried different hyper-parameter tuning methods. Every method gave the same result but GridSearchCV and RandomizedSearchCV took a huge amount of time to train the models. HalvingRandomizedSearchCV took the least time to train the models and predict the output. That's why we use the Tuning_Method as Halving_Randomized_Search_CV

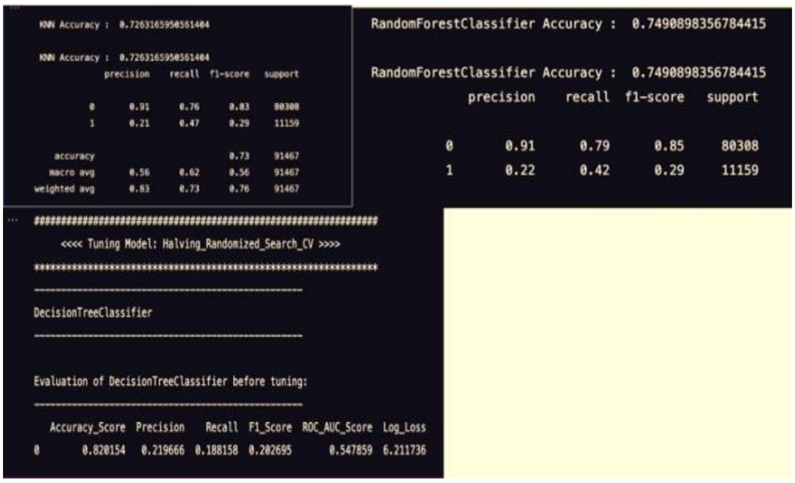
Tuning Methods:

- HalvingRandomizedSearchCV

Evaluation Metrics:

- Accuracy Score
- Precision
- Recall
- F1 Score
- ROC AUC Score

KNN has better results



```
KNN Accuracy :  0.7263165950561404

KNN Accuracy :  0.7263165950561404
              precision    recall  f1-score   support

           0       0.91      0.76      0.83     80308
           1       0.21      0.47      0.29     11159

    accuracy                           0.73     91467
   macro avg       0.56      0.62      0.56     91467
weighted avg       0.83      0.73      0.76     91467
```

```
RandomForestClassifier Accuracy :  0.7490898356784415

RandomForestClassifier Accuracy :  0.7490898356784415
              precision    recall  f1-score   support

           0       0.91      0.79      0.85     80308
           1       0.22      0.42      0.29     11159
```

```
##########################################################################
    <<<< Tuning Model: Halving_Randomized_Search_CV >>>>
**************************************************************************
------------------------------------------------------------
DecisionTreeClassifier
------------------------------------------------------------


Evaluation of DecisionTreeClassifier before tuning:
------------------------------------------------------------
   Accuracy_Score  Precision    Recall  F1_Score  ROC_AUC_Score  Log_Loss
0        0.820154   0.219666  0.188158  0.202695       0.547859  6.211736
```

<u>RESULTS AND DISCUSSION</u>

Starting from loading our dataset, we initially checked for null values and duplicates. There were 50% null values but no duplicates, so treatment of missing values was done and after data processing, we applied feature scaling techniques to normalize our data to bring all features on the same scale and make it easier to process by ML algorithms.

Through Exploratory Data Analysis, we categorized Age, Region_Code ,Policy Sales_Channel etc.further, we observed that customers belonging to young Age are more interested in vehicle response. We observed that customers having vehicles older than 2 years are more likely to be interested in vehicle insurance. Similarly, customers having damaged vehicles are more likely to be interested in vehicle insurance.

For Feature Selection, we used Kendall's rank correlation coefficient for numerical features and for categorical features, we applied the Mutual Information technique. Here we observed that Switch is the most important feature and has the highest impact on the dependent feature and there is no correlation between the two numeric features.

We applied multiple algorithms with hyperparameter tuning and got different results, The top 3 were KNN , decision tree and Random Forest. we selected KNN as our best model considering precision and recall as we have an unequal number of observations in each class in our dataset, so accuracy alone can be misleading.

Although we were asked to predict if the customer is willing to buy the car insurance, seeing the metrics, the above ML algorithm will likely be more efficient in predicting how **likely the customer is not willing to purchase the car insurance**, because of reasons like highly imbalanced data set or maybe huge missing values. There is a scope for improvement in many areas.