Data Collection and Preprocessing Phase

Date	June 2024
Team ID	Team-739856
Project Title	Auto Insurance Fraud Detection Using Machine Learning
Maximum Marks	6 Marks

Preparation Template

The images will be preprocessed by resizing, normalizing, augmenting, de noising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	Description
Data Overview	There are many popular open sources for collecting the data. E.g.: kaggle.com, UCI repository, etc. In this project we have used .csv data.
Data Preparation	These are the general steps of pre-processing the data before using it for machine learning
Handling missing values	We use Handling missing values For checking the null values
Handling categorical data	As we can see our dataset has categorical data we must convert the categorical data to integer encoding or binary encoding
Handling Outliers in Data	With the help of boxplot, outliers are visualized. And here we are going to find upper bound and lower bound of numerical features with some mathematical formula.

Data Preparation

Collect the dataset	Please refer to the link given below to download the dataset. Link: https://www.kaggle.com/datasets/buntyshah/auto-insurance-claims-data
| Importing the libraries | ```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier,GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import f1_score
from sklearn.metrics import classification_report,confusion_matrix
import warnings
import pickle
from scipy import stats
warnings.filterwarnings('ignore')
``` |
| Loading Data | We use the code
`df=pd.read_csv("/content/Train.csv")`
For reading the dataset |

Handling missing values	```df.isna().any()``` months_as_customer False age False policy_number False policy_bind_date False policy_state False policy_csl False policy_deductable False policy_annual_premium False umbrella_limit False insured_zip False insured_sex False insured_education_level False insured_occupation False insured_hobbies False insured_relationship False capital-gains False capital-loss False incident_date False incident_type False collision_type False incident_severity False authorities_contacted True incident_state False incident_city False incident_location False incident_hour_of_the_day False number_of_vehicles_involved False property_damage False bodily_injuries False witnesses False police_report_available False total_claim_amount False injury_claim False property_claim False vehicle_claim False auto_make False auto_model False auto_year False fraud_reported False _c39 True dtype: bool

Handling Categorical values	```python
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd # Import pandas for handling dataframes

for k in df.columns:
  if df[k].dtype in ['int64', 'float64']: # Check if column is numeric
    print(k)
    # Handle potential missing values
    sns.boxplot(df[k].dropna()) # Drop missing values before plotting
    plt.show()
  else:
    print(f"Skipping column '{k}' as it is non-numeric.") # Inform user about skipped columns
```
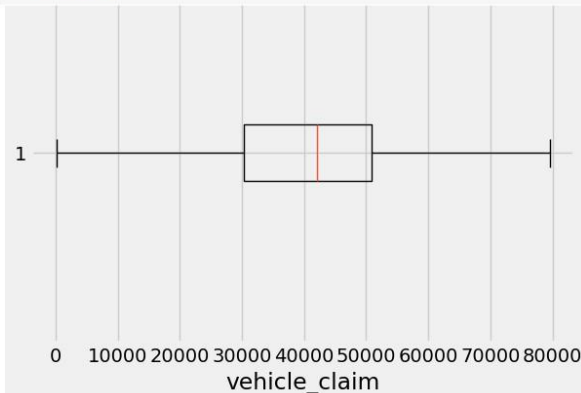
−100000

```
Skipping column 'incident_date' as it is non-numeric.
Skipping column 'incident_type' as it is non-numeric.
Skipping column 'collision_type' as it is non-numeric.
Skipping column 'incident_severity' as it is non-numeric.
Skipping column 'authorities_contacted' as it is non-numeric.
Skipping column 'incident_state' as it is non-numeric.
Skipping column 'incident_city' as it is non-numeric.
Skipping column 'incident_location' as it is non-numeric.
incident_hour_of_the_day
``` |
| Handling Outliers | ```python
#outlier handling

import matplotlib.pyplot as plt
df=pd.read_csv('/content/insurance_claims.csv')
for i in dict(df.dtypes):
  if dict(df.dtypes)[i] == 'int64' or dict(df.dtypes)[i] == 'float64':
    plt.boxplot(df[i], vert = False)
    plt.title("Detecting outlier using Boxplot")
    plt.xlabel(i)
    plt.show()
```



Detecting outlier using Boxplot |