Vehicle Insurance Claim Fraud Detection

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

By

D. SAI VARSHITH 2203A51812

P. NAVEEN KUMAR 2203A51800

E. PRATHUSH KUMAR 2203A51661

K. RAJU 2203A51424

Under the guidance of

Mr. Dr. E.L.N. Kiran Kumar

Associate Professor, CS & AI.

SR UNIVERSITY

Ananthasagar, Warangal.

CERTIFICATE

This is to certify that this project entitled “Vehicle Insurance Claim Fraud Detection" is the project work carried out by D.SAI VARSHITH, P. NAVEEN KUMAR , E. PRATHUSH KUMAR, K. RAJU as a project work for the course Artificial intelligence and machine learning to award the degree BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE &

ENGINEERING during the academic year 2023-2024 under our guidance and Supervision.

Mr.E.L.N. Kiran Kumar	Dr.M.Sheshikala
Assoc. Prof. CS & AI	Assoc. Prof. & HOD(CSE),
SR University ,	SR University,
Ananthasagar, Warangal.	Ananthasagar, Warangal.

External Examiner

ACKNOWLEDGEMENT

We owe an enormous debt of gratitude to our project guide Mr. Dr. E.L.N. KiranKumar, Assoc. Prof. CS and AI as well as Head of the CSE Department Dr.M.Sheshikala, Associate Professor for guiding us from the beginning through the end of the Capstone Phase-II project with their intellectual advices and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction. We express our thanks to project co-ordinators Mr.Sallauddin Md, Asst. Prof, Y.Chanti Asst. Prof. for their encouragement and support. Finally, we express our thanks to all the teaching and non-teaching staff of the department for their suggestions and timely support

TABLE OF CONTENTS

1. ABSTRACT

The rise of fraudulent activities in vehicle insurance claims poses significant challenges to insurance companies, leading to financial losses and compromised customer trust. This abstract outline a comprehensive approach to enhance the detection of fraudulent vehicle insurance claims through advanced analytical techniques.

This research delves into the current landscape of vehicle insurance fraud, highlighting its detrimental impact on the industry and the need for robust detection mechanisms. Leveraging advanced data analytics, including machine learning algorithms, predictive modelling, and anomaly detection, this study proposes a multifaceted framework to identify suspicious patterns and behaviours indicative of fraudulent activities. This can be done by using Decision tree and KNN algorithms.

2. INTRODUCTION:

Insurance fraud happens when people try to cheat the system by making false claims to get money they don't deserve. This not only costs insurance companies a lot of money but also affects honest customers who end up paying higher premiums. Our goal is to develop a smart computer program that can automatically spot these fraudulent claims, helping insurance companies save money and keep costs fair for everyone.

Using a special type of computer learning called machine learning, we're teaching our program to recognize patterns in data. By showing it lots of examples of both real and fake insurance claims, it learns to identify the signs of fraud. Once trained, our program can analyse new claims quickly and accurately, flagging suspicious ones for human review. With this tool, insurance companies can better protect themselves and their customers from fraudsters. So, let's dive in and see how we can make a difference in the world of insurance!

3. PROBLEM STATEMENT:

Developing a computer program to identify fraudulent vehicle insurance claims, using data on accidents and policy history to distinguish between genuine and fake claims, ultimately helping insurance companies save money and maintain fair premiums?

In the realm of vehicle insurance, combating fraudulent claims is critical to preserving financial stability and customer trust. This project aims to develop a robust machine learning model that analyses various claim attributes to accurately identify fraudulent behaviour. By creating a predictive system capable of real-time fraud detection, insurance companies can streamline claim processing, minimize losses, and ensure fair premiums for honest policyholders. Leveraging advanced machine learning techniques, including feature engineering and model optimization, this initiative seeks to empower insurers with a proactive defense against fraud, safeguarding their financial sustainability and upholding industry integrity.

4. LITERATIVE REVIEW

4.1 Related Work

REF NO. DATASET ALGORITHM ACCURACY

REF NO.	DATASET	ALGORITHM	ACCURACY	
1.	Kaggle(Fraud_oracle.csv)	Decision Tree.	92.4%	
2.	Kaggle(Fraud_oracle.csv)	K-Nearest Neighbors	93.8%	:

5. DATASET

Dataset is taken from Kaggle.

A1 | Month

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	Month	WeekOfM	DayOfWee	Make	AccidentA	DayOfWee	MonthClai	WeekOfM	Sex	MaritalSta	Age	Fault	PolicyType	VehicleCat	VehiclePri	FraudFoun	PolicyNum	RepNumb	Deductible	DriverRati	Days_Polic	Days_Polic	PastNumb	Ag
2	Dec	5	Wednesda	Honda	Urban	Tuesday	Jan	1	Female	Single	21	Policy Hol	Sport - Lia	Sport	more than	0	1	12	300	1	more than	more than	none	3 y
3	Jan	3	Wednesda	Honda	Urban	Monday	Jan	4	Male	Single	34	Policy Hol	Sport - Col	Sport	more than	0	2	15	400	4	more than	more than	none	6 y
4	Oct	5	Friday	Honda	Urban	Thursday	Nov	2	Male	Married	47	Policy Hol	Sport - Col	Sport	more than	0	3	7	400	3	more than	more than	1	7 y
5	Jun	2	Saturday	Toyota	Rural	Friday	Jul	1	Male	Married	65	Third Party	Sedan - Lia	Sport	20000 to 2	0	4	4	400	2	more than	more than	1	m
6	Jan	5	Monday	Honda	Urban	Tuesday	Feb	2	Female	Single	27	Third Party	Sport - Col	Sport	more than	0	5	3	400	1	more than	more than	none	5 y
7	Oct	4	Friday	Honda	Urban	Wednesda	Nov	1	Male	Single	20	Third Party	Sport - Col	Sport	more than	0	6	12	400	3	more than	more than	none	5 y
8	Feb	1	Saturday	Honda	Urban	Monday	Feb	3	Male	Married	36	Third Party	Sport - Col	Sport	more than	0	7	14	400	1	more than	more than	1	7 y
9	Nov	1	Friday	Honda	Urban	Tuesday	Mar	4	Male	Single	0	Policy Hol	Sport - Col	Sport	more than	0	8	1	400	4	more than	more than	1	ne
10	Dec	4	Saturday	Honda	Urban	Wednesda	Dec	5	Male	Single	30	Policy Hol	Sport - Col	Sport	more than	0	9	7	400	4	more than	more than	none	6 y
11	Apr	3	Tuesday	Ford	Urban	Wednesda	Apr	3	Male	Married	42	Policy Hol	Utility - All	Utility	more than	0	10	7	400	1	more than	more than	2 to 4	m
12	Mar	2	Sunday	Mazda	Urban	Wednesda	Mar	3	Male	Single	71	Policy Hol	Sedan - All	Sedan	more than	0	11	7	400	3	more than	more than	none	m
13	Mar	5	Monday	Honda	Urban	Monday	Mar	5	Male	Married	52	Policy Hol	Sedan - Lia	Sport	20000 to 2	0	12	13	400	1	more than	more than	2 to 4	m
14	Jan	3	Friday	Ford	Urban	Friday	Jan	3	Male	Married	28	Policy Hol	Sedan - Lia	Sport	more than	0	13	11	400	1	more than	more than	1	7 y
15	Jan	5	Friday	Honda	Rural	Wednesda	Feb	1	Male	Single	0	Third Party	Sedan - Co	Sedan	more than	0	14	12	400	3	more than	more than	none	ne
16	Jan	5	Monday	Ford	Urban	Thursday	Feb	1	Male	Married	61	Policy Hol	Sedan - Lia	Sport	more than	0	15	3	400	1	more than	more than	none	m
17	Aug	4	Tuesday	Ford	Urban	Monday	Aug	5	Male	Single	38	Policy Hol	Sedan - Lia	Sport	more than	0	16	16	400	1	more than	more than	none	6 y
18	Apr	4	Thursday	Ford	Urban	Wednesda	May	1	Male	Married	41	Policy Hol	Sedan - All	Sedan	more than	0	17	15	400	4	more than	more than	none	7 y
19	Jul	5	Sunday	Chevrolet	Urban	Wednesda	Aug	1	Female	Married	28	Third Party	Sedan - Co	Sedan	20000 to 2	0	18	6	400	1	more than	more than	none	7 y
20	May	4	Thursday	Pontiac	Urban	Monday	May	5	Male	Single	32	Policy Hol	Sedan - Lia	Sport	20000 to 2	0	19	6	400	1	more than	more than	1	7 y
21	Apr	4	Monday	Honda	Urban	Tuesday	May	1	Male	Married	30	Third Party	Sedan - Lia	Sport	more than	0	20	2	400	2	more than	more than	2 to 4	6 y
22	Apr	2	Friday	Mazda	Urban	Tuesday	May	1	Male	Married	40	Policy Hol	Sedan - Lia	Sport	20000 to 2	0	21	3	400	1	more than	more than	1	m
23	Jan	2	Saturday	Chevrolet	Urban	Monday	Jan	2	Male	Married	47	Policy Hol	Sedan - Co	Sedan	20000 to 2	0	22	13	400	2	more than	more than	1	m
24	Aug	3	Sunday	Mazda	Urban	Thursday	Aug	5	Male	Married	63	Policy Hol	Sedan - Lia	Sport	20000 to 2	0	23	8	400	3	more than	more than	1	m
25	Jun	3	Saturday	Pontiac	Urban	Tuesday	Jun	3	Male	Single	31	Third Party	Sedan - Lia	Sport	30000 to 3	0	24	5	400	3	more than	more than	none	6 y
26	Sep	3	Friday	Mazda	Urban	Friday	Sep	3	Male	Married	45	Policy Hol	Sedan - All	Sedan	more than	0	25	12	400	3	more than	more than	more than	m
27	Mar	3	Monday	Pontiac	Urban	Tuesday	Apr	1	Male	Married	60	Policy Hol	Sedan - Lia	Sport	20000 to 2	0	26	16	400	4	more than	more than	more than	m

fraud_oracle

H6 | 2

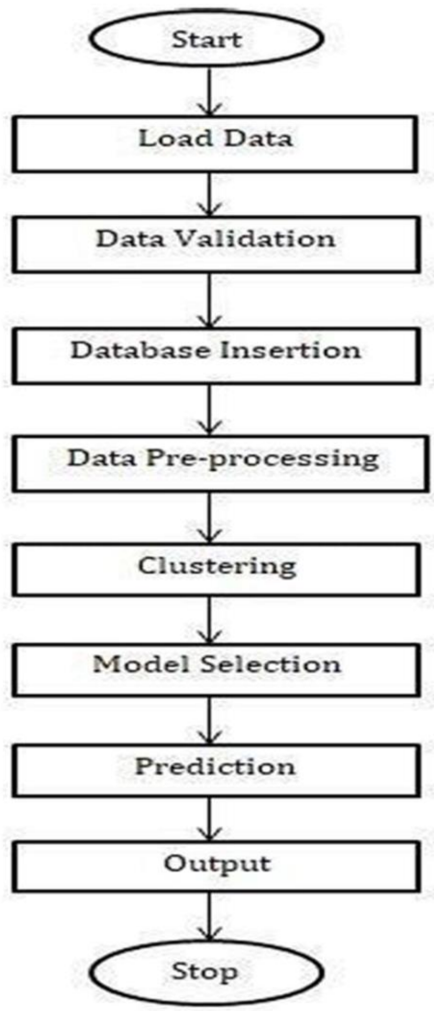
	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG
1	Days_Polic	Days_Polic	PastNumb	AgeOfVehi	AgeOfPolic	PoliceRep	WitnessPr	AgentType	NumberOf	AddressCh	NumberOf	Year	BasePolicy
2	more than	more than	none	3 years	26 to 30	No	No	External	none	1 year	3 to 4	1994	Liability
3	more than	more than	none	6 years	31 to 35	Yes	No	External	none	no change	1 vehicle	1994	Collision
4	more than	more than	1	7 years	41 to 50	No	No	External	none	no change	1 vehicle	1994	Collision
5	more than	more than	1	more than	51 to 65	Yes	No	External	more than	no change	1 vehicle	1994	Liability
6	more than	more than	none	5 years	31 to 35	No	No	External	none	no change	1 vehicle	1994	Collision
7	more than	more than	none	5 years	21 to 25	No	No	External	3 to 5	no change	1 vehicle	1994	Collision
8	more than	more than	1	7 years	36 to 40	No	No	External	1 to 2	no change	1 vehicle	1994	Collision
9	more than	more than	1	new	16 to 17	No	No	External	none	no change	1 vehicle	1994	Collision
10	more than	more than	none	6 years	31 to 35	No	Yes	External	3 to 5	no change	1 vehicle	1994	Collision
11	more than	more than	2 to 4	more than	36 to 40	No	No	External	3 to 5	no change	1 vehicle	1994	All Perils
12	more than	more than	none	more than	over 65	No	No	External	none	no change	1 vehicle	1994	All Perils
13	more than	more than	2 to 4	more than	41 to 50	No	No	External	none	no change	1 vehicle	1994	Liability
14	more than	more than	1	7 years	31 to 35	No	No	External	none	no change	1 vehicle	1994	Liability
15	more than	more than	none	new	16 to 17	No	No	External	none	no change	1 vehicle	1994	Collision
16	more than	more than	none	more than	51 to 65	No	No	External	none	no change	1 vehicle	1994	Liability
17	more than	more than	none	6 years	36 to 40	No	No	External	none	no change	1 vehicle	1994	Liability
18	more than	more than	none	7 years	36 to 40	No	No	External	none	no change	1 vehicle	1994	All Perils
19	more than	more than	none	7 years	31 to 35	No	No	External	1 to 2	no change	1 vehicle	1994	Collision
20	more than	more than	1	7 years	31 to 35	No	No	External	none	no change	1 vehicle	1994	Liability
21	more than	more than	2 to 4	6 years	31 to 35	No	No	External	more than	no change	1 vehicle	1994	Liability
22	more than	more than	1	more than	36 to 40	No	No	External	more than	no change	1 vehicle	1994	Liability
23	more than	more than	1	more than	41 to 50	No	No	External	none	4 to 8 year	2 vehicles	1994	Collision
24	more than	more than	1	more than	51 to 65	No	No	External	1 to 2	no change	1 vehicle	1994	Liability
25	more than	more than	none	6 years	31 to 35	No	No	External	3 to 5	no change	1 vehicle	1994	Liability
26	more than	more than	more than	more than	36 to 40	Yes	No	External	none	no change	1 vehicle	1994	All Perils
27	more than	more than	more than	more than	51 to 65	No	No	External	none	no change	1 vehicle	1994	Liability

6. PROPOSED METHODOLOGY

The methodology of decision trees involves recursively partitioning the dataset based on the features that best split the data into homogeneous subsets with respect to the target variable. At each step, the algorithm selects the feature and corresponding split point that maximizes the purity of the resulting subsets, typically measured by metrics such as Gini impurity or information gain. This process continues until a stopping criterion is met, such as reaching a maximum tree depth or no further improvement in purity. Decision trees are versatile and interpretable models that can handle both numerical and categorical data, making them widely used in various domains for classification and regression tasks.

The methodology of K-nearest neighbours (KNN) involves classifying or regressing data points based on the majority vote or averaging of their nearest neighbours in the feature space. Given a new data point, KNN calculates the distance to all other points in the dataset, typically using Euclidean distance or other distance metrics. It then identifies the K nearest neighbours to the new point and assigns a class label or regression value based on the most common class or average value among these neighbours. KNN is a non-parametric and lazy learning algorithm, meaning it does not require training before making predictions and instead relies on the entire training dataset during inference.

6.1 FLOW CHART:



6.2 COMPARED ALGORITHMS

6.2.1 DECISION TREE

Decision trees are the most common way to say something. They are strong on sound data and learn divisive sayings. Decision tree is a k-array tree where each internal node displays an experiment in a few elements from a set of input element that communicates with the data. Every branch from a node is related to the unimaginable feature values determined for that node. Also, all test results in branches, refer to changed test results. The basic algorithm for decision tree imports algorithm is the decision-making tree algorithm in the form of repeated downward divisions and conquests.

The Decision Tree algorithm is a powerful tool used in our vehicle insurance fraud detection project. It helps in making decisions by mapping out various outcomes and consequences in a tree-like model. This algorithm analyses different characteristics like policyholder details and vehicle specifics to identify potential fraud. It's proven effective in detecting fraudulent activities, thereby assisting insurance companies in mitigating losses and enhancing operational efficiency.

6.2.2 K-Nearest Neighbours

The K-Nearest Neighbours (KNN) algorithm is a key component in our vehicle insurance fraud detection system. It works by classifying new data points based on their similarity to known data. In the context of vehicle insurance, it uses features like policyholder details and vehicle information to predict whether a claim could be fraudulent. This algorithm has shown its effectiveness in identifying potential fraud, thereby helping insurance companies prevent losses and enhance their fraud detection capabilities.

Twitter Data Sensory Analysis using KNN Editing. Emotional analysis refers to the use of natural language processing, text analysis, and computer languages to systematically identify, extract, evaluate, and learn practical situations and independent information. Sentiment Analysis is the most widely used method of quoting a text. Twitter Sentiment Analysis, therefore, means using advanced text-cutting techniques to analyse text emotions (here, tweet) in a positive, negative and neutral way.

7. RESULTS & DISCUSSION

Code:

Importing Libraries

```
[1] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    %matplotlib inline
```

Fig: Libraries Importing

Reading the csv file using pandas library

```
df = pd.read_csv('fraud_oracle.csv')
df.head()
```

	Month	WeekOfMonth	DayOfWeek	Make	AccidentArea	DayOfWeekClaimed	MonthClaimed	WeekOfMonthClaimed	Sex	MaritalStatus	...	AgeOfVehicle	AgeOfPolicyHolder	PoliceReportFile
0	Dec	5	Wednesday	Honda	Urban	Tuesday	Jan	1	Female	Single	...	3 years	26 to 30	N
1	Jan	3	Wednesday	Honda	Urban	Monday	Jan	4	Male	Single	...	6 years	31 to 35	Ye
2	Oct	5	Friday	Honda	Urban	Thursday	Nov	2	Male	Married	...	7 years	41 to 50	N
3	Jun	2	Saturday	Toyota	Rural	Friday	Jul	1	Male	Married	...	more than 7	51 to 65	Ye
4	Jan	5	Monday	Honda	Urban	Tuesday	Feb	2	Female	Single	...	5 years	31 to 35	N

5 rows × 33 columns

Preprocessing analysis

```
df.info(verbose='True')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15420 entries, 0 to 15419
Data columns (total 33 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   Month                 15420 non-null  object
 1   WeekOfMonth           15420 non-null  int64 
 2   DayOfWeek             15420 non-null  object
 3   Make                  15420 non-null  object
 4   AccidentArea          15420 non-null  object
 5   DayOfWeekClaimed      15420 non-null  object
 6   MonthClaimed          15420 non-null  object
 7   WeekOfMonthClaimed    15420 non-null  int64 
 8   Sex                   15420 non-null  object
 9   MaritalStatus         15420 non-null  object
 10  Age                   15420 non-null  int64 
 11  Fault                 15420 non-null  object
 12  PolicyType            15420 non-null  object
 13  VehicleCategory       15420 non-null  object
 14  VehiclePrice          15420 non-null  object
 15  FraudFound_P          15420 non-null  int64 
 16  PolicyNumber          15420 non-null  int64 
 17  RepNumber             15420 non-null  int64 
 18  Deductible            15420 non-null  int64 
 19  DriverRating          15420 non-null  int64 
 20  Days_Policy_Accident  15420 non-null  object
 21  Days_Policy_Claim     15420 non-null  object
 22  PastNumberOfClaims    15420 non-null  object
 23  AgeOfVehicle          15420 non-null  object
 24  AgeOfPolicyHolder     15420 non-null  object
 25  PoliceReportFiled     15420 non-null  object
 26  WitnessPresent        15420 non-null  object
 27  AgentType             15420 non-null  object
 28  NumberOfSuppliments   15420 non-null  object
 29  AddressChange_Claim   15420 non-null  object
 30  NumberOfCars          15420 non-null  object
 31  Year                  15420 non-null  int64 
 32  BasePolicy            15420 non-null  object
dtypes: int64(9), object(24)
memory usage: 3.9+ MB
```

```
df.describe()
```

	WeekOfMonth	WeekOfMonthClaimed	Age	FraudFound_P	PolicyNumber	RepNumber	Deductible	DriverRating	Year
count	15420.000000	15420.000000	15420.000000	15420.000000	15420.000000	15420.000000	15420.000000	15420.000000	15420.000000
mean	2.788586	2.693969	39.855707	0.059857	7710.500000	8.483268	407.704280	2.487808	1994.866472
std	1.287585	1.259115	13.492377	0.237230	4451.514911	4.599948	43.950998	1.119453	0.803313
min	1.000000	1.000000	0.000000	0.000000	1.000000	1.000000	300.000000	1.000000	1994.000000
25%	2.000000	2.000000	31.000000	0.000000	3855.750000	5.000000	400.000000	1.000000	1994.000000
50%	3.000000	3.000000	38.000000	0.000000	7710.500000	8.000000	400.000000	2.000000	1995.000000
75%	4.000000	4.000000	48.000000	0.000000	11565.250000	12.000000	400.000000	3.000000	1996.000000
max	5.000000	5.000000	80.000000	1.000000	15420.000000	16.000000	700.000000	4.000000	1996.000000

describe() is used to know about several statistical measures such as mean,median,standard deviation and many more.

```
[5] df.isnull().sum()
```

```
Month                    0
WeekOfMonth              0
DayOfWeek                0
Make                     0
AccidentArea             0
DayOfWeekClaimed         0
MonthClaimed             0
WeekOfMonthClaimed       0
Sex                      0
MaritalStatus            0
Age                      0
Fault                    0
PolicyType               0
VehicleCategory          0
VehiclePrice             0
FraudFound_P             0
PolicyNumber             0
RepNumber                0
Deductible               0
DriverRating             0
Days_Policy_Accident     0
Days_Policy_Claim        0
PastNumberOfClaims       0
AgeOfVehicle             0
AgeOfPolicyHolder        0
PoliceReportFiled        0
WitnessPresent           0
AgentType                0
NumberOfSuppliments      0
AddressChange_Claim      0
NumberOfCars             0
Year                     0
BasePolicy               0
dtype: int64
```

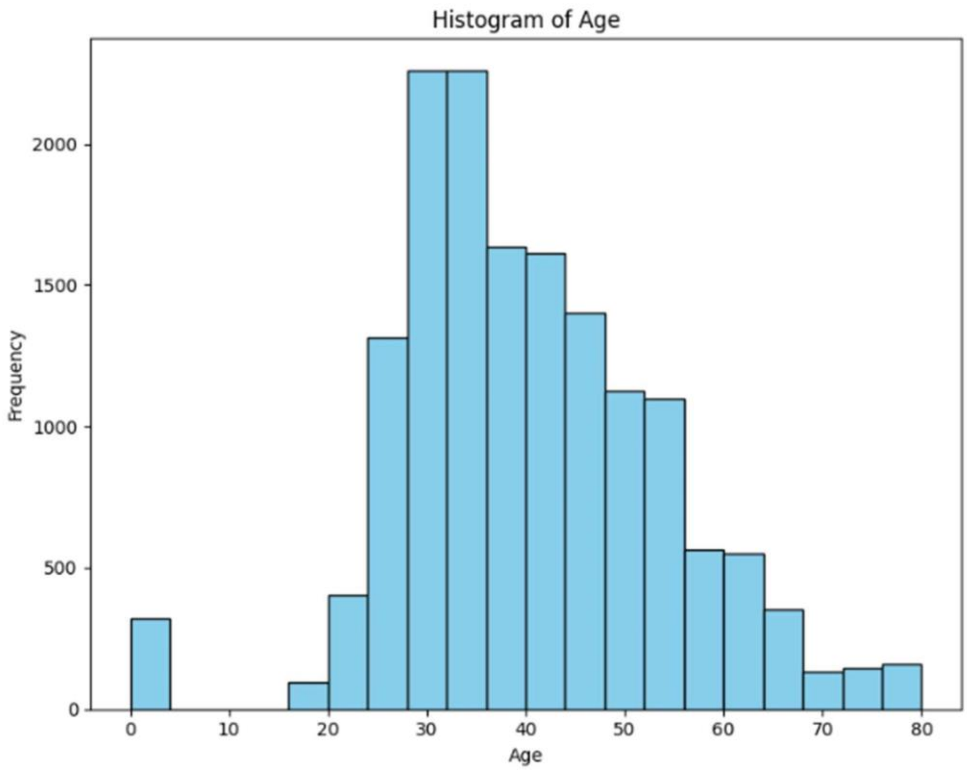
isnull().sum() is used to count the total number of null values.

```
[6] df.shape
```

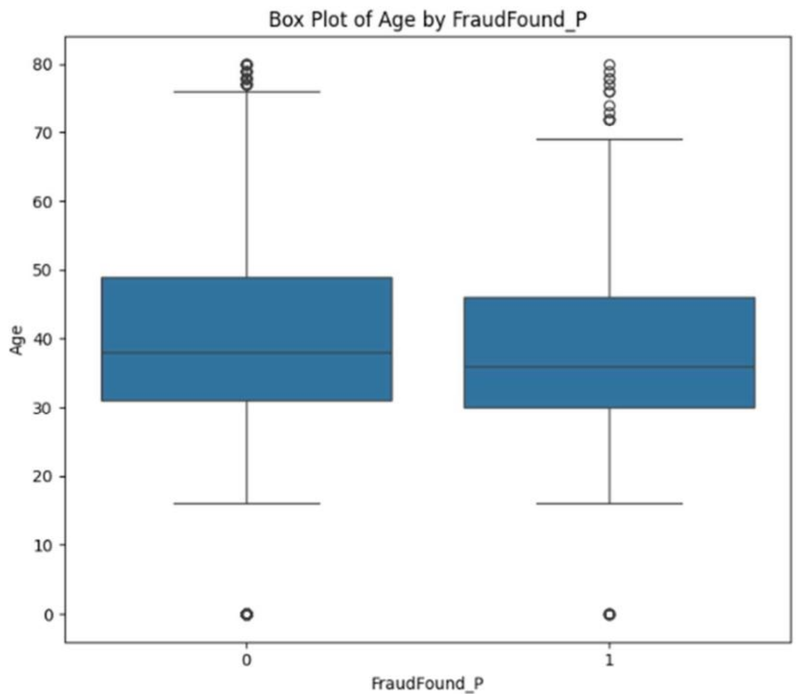
```
(15420, 33)
```

Data Visualization:

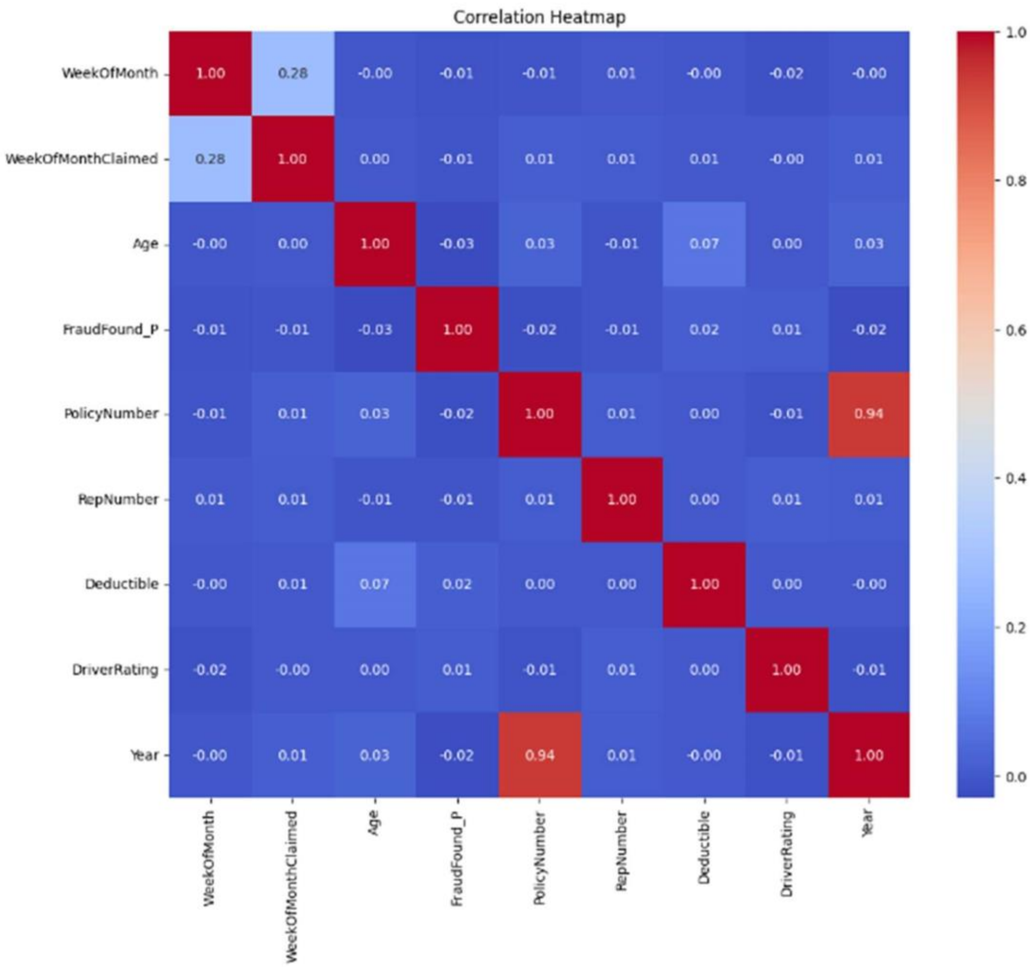
```python
# Features to create histograms for
features = ['Age', 'VehiclePrice', 'RepNumber', 'Deductible', 'DriverRating']

# Create subplots
fig, axes = plt.subplots(nrows=len(features), ncols=1, figsize=(8, 6*len(features)))

# Plot histograms for each feature
for i, feature in enumerate(features):
    axes[i].hist(df[feature], bins=20, color='skyblue', edgecolor='black')
    axes[i].set_title(f'Histogram of {feature}')
    axes[i].set_xlabel(feature)
    axes[i].set_ylabel('Frequency')

# Adjust layout and display plots
plt.tight_layout()
plt.show()
```



```python
#Box Plots for Numerical Features vs. Target Variable
# Features and target variable
numerical_features = ['Age', 'VehiclePrice', 'RepNumber', 'Deductible', 'DriverRating']
target_variable = 'FraudFound_P'  # Replace 'FraudFound_P' with your actual target variable

# Create subplots
fig, axes = plt.subplots(nrows=len(numerical_features), ncols=1, figsize=(8, 6*len(numerical_features)))

# Plot box plots for each feature against the target variable
for i, feature in enumerate(numerical_features):
    sns.boxplot(x=target_variable, y=feature, data=df, ax=axes[i])
    axes[i].set_title(f'Box Plot of {feature} by {target_variable}')
    axes[i].set_xlabel(target_variable)
    axes[i].set_ylabel(feature)

# Adjust layout and display plots
plt.tight_layout()
plt.show()
```

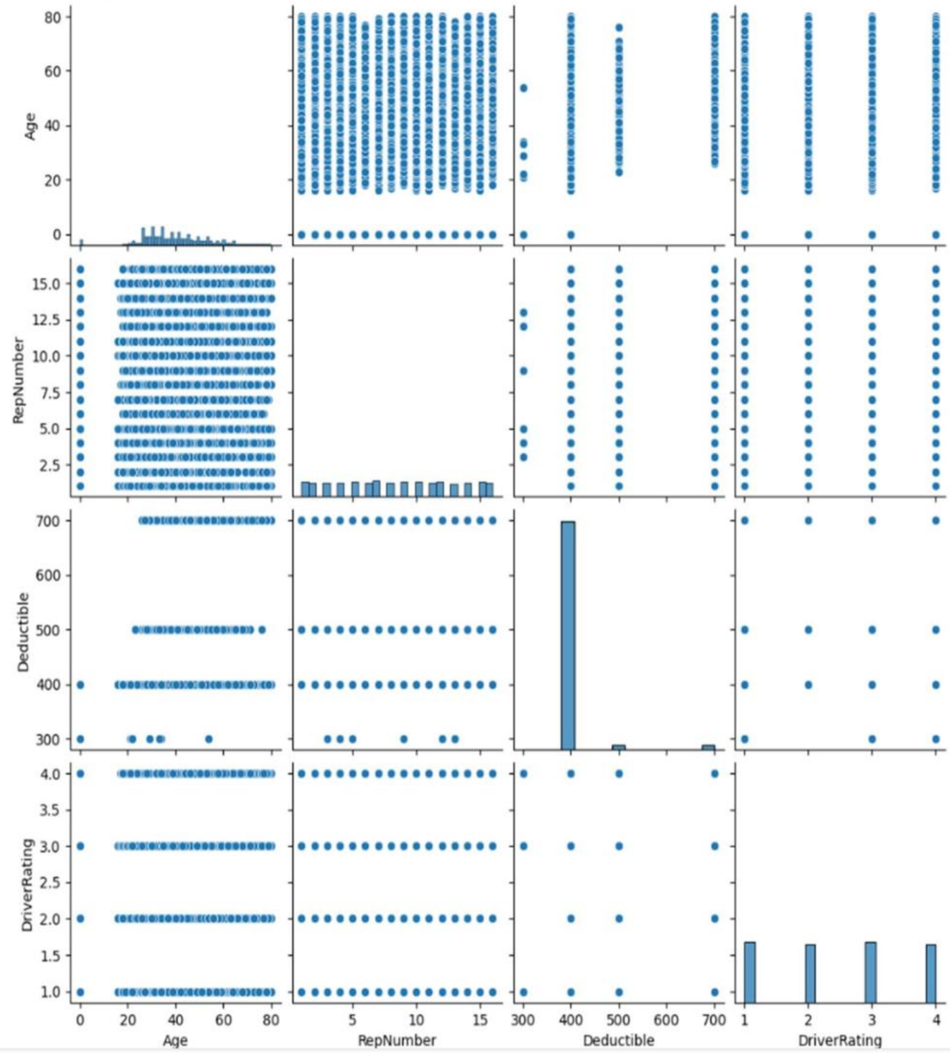


```python
#Correlation Heatmap
# Select only numeric columns
numeric_columns = df.select_dtypes(include=['float64', 'int64'])

# Calculate the correlation matrix
corr = numeric_columns.corr()

# Create a heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f", annot_kws={"size": 10})
plt.title('Correlation Heatmap')
plt.show()
```



```
sns.pairplot(new_data)
```

```
<seaborn.axisgrid.PairGrid at 0x7b6d3b71b490>
```



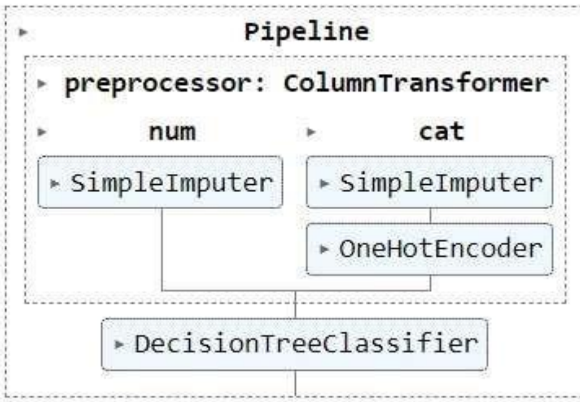
Training the Model using Decision Tree

```
[34] # Split data into features and target variable
     X = df.drop(columns=['FraudFound_P'])  # Features
     y = df['FraudFound_P']  # Target variable
```

```
[35] from sklearn.model_selection import train_test_split
     # Split the data into training and testing sets
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
from sklearn.compose import make_column_selector as selector
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.tree import DecisionTreeClassifier

# Define numerical and categorical columns
numerical_columns = selector(dtype_exclude=object)(X_train)
categorical_columns = selector(dtype_include=object)(X_train)

# Define preprocessing steps
numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median'))
])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

# Bundle preprocessing for numerical and categorical data
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numerical_columns),
        ('cat', categorical_transformer, categorical_columns)
])
```

```python
])

# Append classifier to preprocessing pipeline
clf = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', DecisionTreeClassifier())
])

# Train the model
clf.fit(X_train, y_train)
```




```python
from sklearn.metrics import accuracy_score,classification_report
# Drop rows with missing values in both X_test and y_test
X_test = X_test.dropna()
y_test = y_test.dropna()

# Predict on the testing set
y_pred = clf.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
misclassification_rate = 1 - accuracy
print("Accuracy:", accuracy)
print("Misclassification Rate:", misclassification_rate)
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.9267185473411155
Misclassification Rate: 0.07328145265888453
Classification Report:
               precision    recall  f1-score   support

           0       0.96      0.96      0.96      4341
           1       0.40      0.38      0.39       285

    accuracy                           0.93      4626
   macro avg       0.68      0.67      0.68      4626
weighted avg       0.93      0.93      0.93      4626
```

```
from sklearn.model_selection import cross_val_score
# Combine features and target variable for dropping rows with missing values
combined_data = pd.concat([X, y], axis=1)

# Drop rows with missing values in both features and target variable
combined_data.dropna(inplace=True)

# Separate features and target variable after handling missing values
X_processed = combined_data.drop(columns=['FraudFound_P'])
y_processed = combined_data['FraudFound_P']

# Cross-validation
cv_scores = cross_val_score(clf, X_processed, y_processed, cv=5)
print("Cross-validation Mean Accuracy:", cv_scores.mean())
```

```
Cross-validation Mean Accuracy: 0.7162127107652401
```

[39]

```
# Create a KNN classifier pipeline
clf = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', KNeighborsClassifier())])

# Train the KNN classifier
clf.fit(X_train, y_train)

# Predict on the testing set
y_pred = clf.predict(X_test)

# Evaluate the model
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.9383916990920882
Classification Report:
               precision    recall  f1-score   support

           0       0.94      1.00      0.97      4341
           1       0.00      0.00      0.00       285

    accuracy                           0.94      4626
   macro avg       0.47      0.50      0.48      4626
weighted avg       0.88      0.94      0.91      4626
```

```
# Calculate accuracy and misclassification rate
accuracy = accuracy_score(y_test, y_pred)
misclassification_rate = 1 - accuracy

print("Accuracy:", accuracy)
print("Misclassification Rate:", misclassification_rate)
```

```
Accuracy: 0.9383916990920882
Misclassification Rate: 0.061608300907911806
```

8.Conclusion:

So, I hereby conclude that the machine learning project for detecting vehicle insurance fraud has been successful. The model, trained on past data, effectively identifies fraud patterns by analysing policyholder and vehicle details. This aids insurance companies in reducing losses. The Decision tree algorithm used has proven to be effective, achieving high accuracy and providing insights into the main factors in fraud detection. Regular updates are necessary to keep the model effective. This project has demonstrated the potential of machine learning in enhancing fraud detection capabilities in the vehicle insurance domain.

In the context of fraud detection, precision and recall are often more important metrics than overall accuracy. Precision measures the proportion of true positive predictions among all positive predictions, while recall measures the proportion of true positives that were correctly identified. We got K-Nearest Neighbours (KNN) has a slightly higher accuracy (93.8%) compared to Decision Trees (92.6%).For fraud detection, interpretability might be crucial, as understanding why a certain decision was made is important for investigation purposes. In that case, Decision Trees might be preferred due to their transparent nature.