

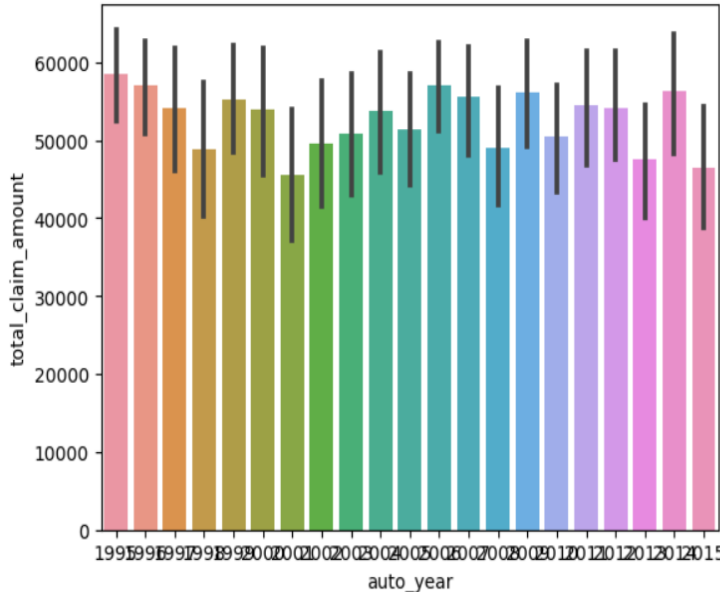
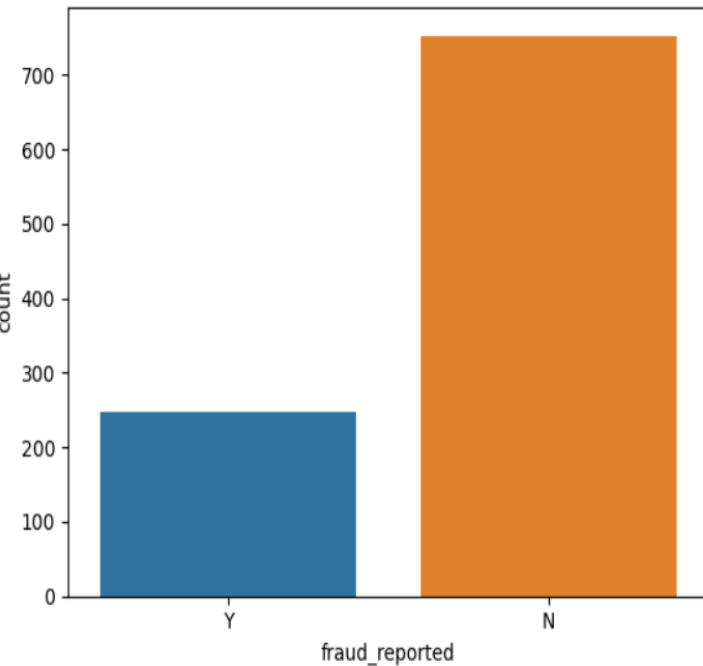
Data Collection and Preprocessing Phase

Date	18 June 2024
Team ID	739990
Project Title	Auto Insurance Fraud Detection
Maximum Marks	6 Marks

Data Exploration and Preprocessing Report

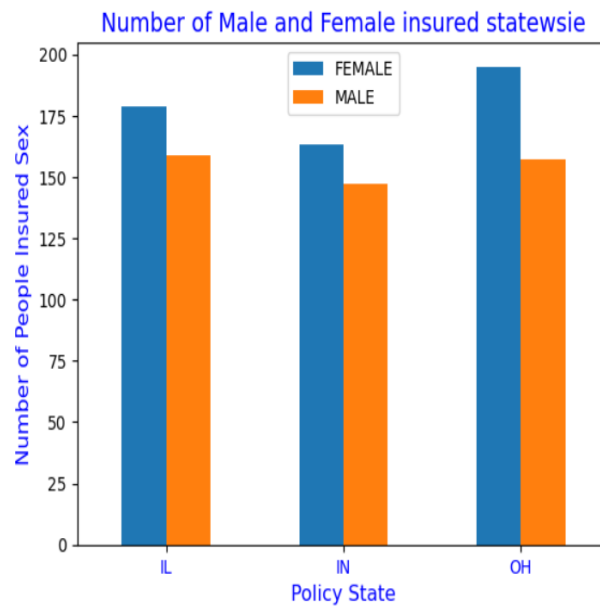
Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions.

Section	Description																																																																																										
Data Overview	<u>Dimension:</u> 1000rows × 40columns <u>Descriptive statistics:</u>																																																																																										
	<table><tr><th></th><th>months_as_customer</th><th>age</th><th>policy_number</th><th>policy_deductable</th><th>policy_annual_premium</th><th>umbrella_limit</th><th>insured_zip</th><th>capital-gains</th><th>capital-loss</th></tr><tr><td>count</td><td>1000.000000</td><td>1000.000000</td><td>1000.000000</td><td>1000.000000</td><td>1000.000000</td><td>1.000000e+03</td><td>1000.000000</td><td>1000.000000</td><td>1000.000000</td></tr><tr><td>mean</td><td>203.954000</td><td>38.948000</td><td>546238.648000</td><td>1136.000000</td><td>1256.406150</td><td>1.101000e+06</td><td>501214.488000</td><td>25126.100000</td><td>-26793.700000</td></tr><tr><td>std</td><td>115.113174</td><td>9.140287</td><td>257063.005276</td><td>611.864673</td><td>244.167395</td><td>2.297407e+06</td><td>71701.610941</td><td>27872.187708</td><td>28104.096686</td></tr><tr><td>min</td><td>0.000000</td><td>19.000000</td><td>100804.000000</td><td>500.000000</td><td>433.330000</td><td>-1.000000e+06</td><td>430104.000000</td><td>0.000000</td><td>-111100.000000</td></tr><tr><td>25%</td><td>115.750000</td><td>32.000000</td><td>335980.250000</td><td>500.000000</td><td>1089.607500</td><td>0.000000e+00</td><td>448404.500000</td><td>0.000000</td><td>-51500.000000</td></tr><tr><td>50%</td><td>199.500000</td><td>38.000000</td><td>533135.000000</td><td>1000.000000</td><td>1257.200000</td><td>0.000000e+00</td><td>466445.500000</td><td>0.000000</td><td>-23250.000000</td></tr><tr><td>75%</td><td>276.250000</td><td>44.000000</td><td>759099.750000</td><td>2000.000000</td><td>1415.695000</td><td>0.000000e+00</td><td>603251.000000</td><td>51025.000000</td><td>0.000000</td></tr><tr><td>max</td><td>479.000000</td><td>64.000000</td><td>999435.000000</td><td>2000.000000</td><td>2047.590000</td><td>1.000000e+07</td><td>620962.000000</td><td>100500.000000</td><td>0.000000</td></tr></table>		months_as_customer	age	policy_number	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	capital-gains	capital-loss	count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1.000000e+03	1000.000000	1000.000000	1000.000000	mean	203.954000	38.948000	546238.648000	1136.000000	1256.406150	1.101000e+06	501214.488000	25126.100000	-26793.700000	std	115.113174	9.140287	257063.005276	611.864673	244.167395	2.297407e+06	71701.610941	27872.187708	28104.096686	min	0.000000	19.000000	100804.000000	500.000000	433.330000	-1.000000e+06	430104.000000	0.000000	-111100.000000	25%	115.750000	32.000000	335980.250000	500.000000	1089.607500	0.000000e+00	448404.500000	0.000000	-51500.000000	50%	199.500000	38.000000	533135.000000	1000.000000	1257.200000	0.000000e+00	466445.500000	0.000000	-23250.000000	75%	276.250000	44.000000	759099.750000	2000.000000	1415.695000	0.000000e+00	603251.000000	51025.000000	0.000000	max	479.000000	64.000000	999435.000000	2000.000000	2047.590000	1.000000e+07	620962.000000	100500.000000	0.000000
		months_as_customer	age	policy_number	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	capital-gains	capital-loss																																																																																	
	count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1.000000e+03	1000.000000	1000.000000	1000.000000																																																																																	
	mean	203.954000	38.948000	546238.648000	1136.000000	1256.406150	1.101000e+06	501214.488000	25126.100000	-26793.700000																																																																																	
	std	115.113174	9.140287	257063.005276	611.864673	244.167395	2.297407e+06	71701.610941	27872.187708	28104.096686																																																																																	
	min	0.000000	19.000000	100804.000000	500.000000	433.330000	-1.000000e+06	430104.000000	0.000000	-111100.000000																																																																																	
	25%	115.750000	32.000000	335980.250000	500.000000	1089.607500	0.000000e+00	448404.500000	0.000000	-51500.000000																																																																																	
	50%	199.500000	38.000000	533135.000000	1000.000000	1257.200000	0.000000e+00	466445.500000	0.000000	-23250.000000																																																																																	
	75%	276.250000	44.000000	759099.750000	2000.000000	1415.695000	0.000000e+00	603251.000000	51025.000000	0.000000																																																																																	
max	479.000000	64.000000	999435.000000	2000.000000	2047.590000	1.000000e+07	620962.000000	100500.000000	0.000000																																																																																		
Univariate Analysis																																																																																											

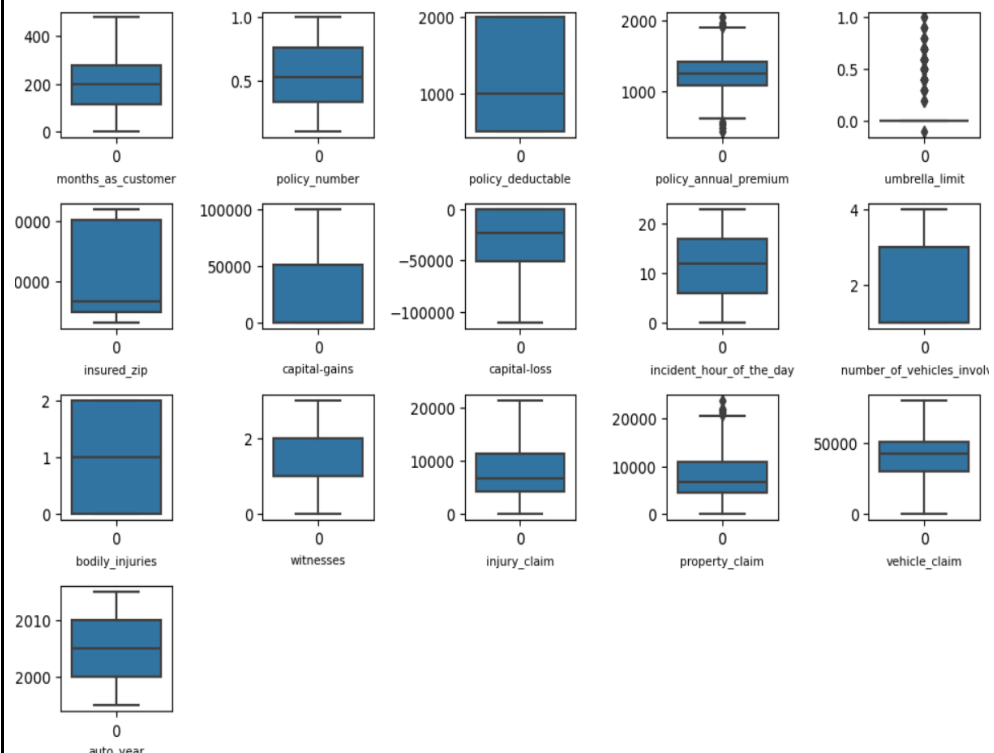
	<pre>sns.barplot(x='auto_year',y='total_claim_amount',data=df)</pre> <p><Axes: xlabel='auto_year', ylabel='total_claim_amount'></p> 
<p>Bivariate Analysis</p>	<pre>sns.countplot(x='fraud_reported',data=df)</pre> <p><Axes: xlabel='fraud_reported', ylabel='count'></p> 

Multivariate Analysis

```
insurance_state=pd.crosstab(data['policy_state'],data['insured_sex'])
insurance_state.plot(kind='bar',grid=False)
plt.xticks(rotation=0,fontsize=10,color='blue')
plt.legend(fontsize=10)
plt.xlabel('Policy State',fontsize=12,color='blue')
plt.ylabel('Number of People Insured Sex',fontsize=12,color='blue')
plt.title('Number of Male and Female insured statesie',fontsize=14,color='blue')
plt.show()
```



Outliers



Data Preprocessing Code Screenshots

Loading Data

```
data=pd.read_csv('insurance_claims.csv')
```

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip
0	328	48	521585	2014-10-17	OH	250/500	1000	1406.91	0	466132
1	228	42	342868	2006-06-27	IN	250/500	2000	1197.22	5000000	468176
2	134	29	687698	2000-09-06	OH	100/300	2000	1413.14	5000000	430632
3	256	41	227811	1990-05-25	IL	250/500	2000	1415.74	6000000	608117
4	228	44	367455	2014-06-06	IL	500/1000	1000	1583.91	6000000	610706
...
995	3	38	941851	1991-07-16	OH	500/1000	1000	1310.80	0	431289
996	285	41	186934	2014-01-05	IL	100/300	1000	1436.79	0	608177
997	130	34	918516	2003-02-17	OH	250/500	500	1383.49	3000000	442797
998	458	62	533940	2011-11-18	IL	500/1000	2000	1356.92	5000000	441714
999	456	60	556080	1996-11-11	OH	250/500	1000	766.19	0	612260

000 rows x 40 columns

Handling Missing Data

```
data['collision_type'] = data['collision_type'].fillna(data['collision_type'].mode()[0])
data['property_damage'] = data['property_damage'].fillna(data['property_damage'].mode()[0])
data['police_report_available'] = data['police_report_available'].fillna(data['police_report_available'].mode()[0])
data['age'] = data['age'].fillna(data['age'].mode()[0])
data['total_claim_amount'] = data['total_claim_amount'].fillna(data['total_claim_amount'].mode()[0])
data['authorities_contacted'] = data['authorities_contacted'].fillna(data['authorities_contacted'].mode()[0])
```

Data Transformation

```
data['Gender']=data['Gender'].map({'Female':1,'Male':0})
data['Property_Area']=data['Property_Area'].map({'Urban':2,'Semiurban': 1,'Rural':0})
data['Married']=data['Married'].map({'Yes':1,'No':0})
data['Education']=data['Education'].map({'Graduate':1,'Not Graduate':0})
data['Loan_Status']=data['Loan_Status'].map({'Y':1,'N':0})

# performing feature Scaling operation using standard scaller on X part of the dataset because
# there different type of values in the columns
sc=StandardScaler()
x_bal=sc.fit_transform(x_bal)
```

Feature Engineering	Attached the codes in final submission.
Save Processed Data	-