

PROMPT 1 (Copilot): Visualize histogram subplot for numeric columns

Modify the following code snippet to generate subplots with 3 columns:

```
for col in num_cols:
```

```
    sns.histplot(  
        data= df_copy[col],
```

```
        kde= True,
```

```
        bins= 30
```

```
    )
```

```
plt.show()
```

PROMPT (ChatGPT): Statistical Test for Bimodality

2a : You are a data scientist with a degree in applied statistics. While working on a project we observed that one of the features shows bimodal distribution. Suggest an easy statistical test to support this claim via python.

2b : Now Suggest technique to identify the exact point to split the data into two separate modes.

PROMPT 3 (Copilot): Visualize count subplot for categorical columns

Modify the following code snippet to generate subplots with 3 columns:

```
for col in discrete_num_cols:
```

```
    sns.countplot(  
        data= df_copy,
```

```
        x= col
```

```
        color= "skyblue",
```

```
        edgecolor= "black"
```

```
    )
```

```
plt.show()
```

PROMPT 4 (Copilot): Add data labels in horizontal bar

Modify the following code snippet to add data labels slightly above the bars. Also, increase the y limit by max + 50

```
plt.barh(

    y= df_copy['fraud_reported'].value_counts().values
    width= df_copy['fraud_reported'].value_counts().index
    color= "skyblue",
    edgecolor= "black"
)

plt.title('Imbalance in Fraud Reported')

plt.show()
```

PROMPT 5 (Copilot): Generate subplot with one row

Modify the following code snippet to generate subplot of two columns and one row:

```
# First Mode

sns.histplot(

    data=df_copy[df_copy['total_claim_amount'] <= split_point]['total_claim_amount'],
    kde=True,
    bins=30,
)

plt.title("First Mode Distribution")

# Second Mode

sns.histplot(

    data=df_copy[df_copy['total_claim_amount'] > split_point]['total_claim_amount'],
    kde=True,
    bins=30,
)

plt.title("Second Mode Distribution")

plt.tight_layout()

plt.show()
```

PROMPT 6 (ChatGPT): Filter and return dataframe based on one value

6a: Generate code to return rows with the value "?" from the dataframe using pandas.

6b: modify code this code line "rows_with_question_mark = df[df.isin(['?']).any(axis=1)]" to also return columns using .loc

PROMPT 7 (Copilot): Hyperparameter fine tuning

Generate python code to fine tune hyperparameters of Random Forest using Grid Search CV.

PROMPT 8 (ChatGPT): Input Data Processing Script

You are a data scientist with 10+ years of experience and proficiency in Python.

- Your task is to develop a preprocessing script
- To transform input data features into processed features using IF and CASE WHEN statements, except for numeric features.
- We have imported a scalar object
- Assign numeric values to the respective processed features and use scaler object for feature scaling using cols to scale

Code:``python

```
processed_features = ['policy_deductable', 'policy_annual_premium', 'umbrella_limit', 'capital-gains',  
'capital-loss', 'incident_severity', 'incident_hour_of_the_day', 'total_claim_amount', 'auto_year',  
'fraud_reported', 'day', 'insured_occupation_armed-forces', 'insured_occupation_craft-repair',  
'insured_occupation_exec-managerial', 'insured_occupation_farming-fishing',  
'insured_occupation_handlers-cleaners', 'insured_occupation_machine-op-inspct',  
'insured_occupation_other-service', 'insured_occupation_priv-house-serv',  
'insured_occupation_prof-specialty', 'insured_occupation_protective-serv',  
'insured_occupation_sales', 'insured_occupation_tech-support', 'insured_occupation_transport-  
moving', 'insured_hobbies_basketball', 'insured_hobbies_board-games', 'insured_hobbies_bungie-  
jumping', 'insured_hobbies_camping', 'insured_hobbies_chess', 'insured_hobbies_cross-fit',  
'insured_hobbies_dancing', 'insured_hobbies_exercise', 'insured_hobbies_golf',  
'insured_hobbies_hiking', 'insured_hobbies_kayaking', 'insured_hobbies_movies',  
'insured_hobbies_paintball', 'insured_hobbies_polo', 'insured_hobbies_reading',  
'insured_hobbies_skydiving', 'insured_hobbies_sleeping', 'insured_hobbies_video-games',  
'insured_hobbies_yachting', 'auto_model_92x', 'auto_model_93', 'auto_model_95',  
'auto_model_A3', 'auto_model_A5', 'auto_model_Accord', 'auto_model_C300', 'auto_model_CRV',  
'auto_model_Camry', 'auto_model_Civic', 'auto_model_Corolla', 'auto_model_E400',  
'auto_model_Escape', 'auto_model_F150', 'auto_model_Forrestor', 'auto_model_Fusion',  
'auto_model_Grand Cherokee', 'auto_model_Highlander', 'auto_model_Impreza',  
'auto_model_Jetta', 'auto_model_Legacy', 'auto_model_M5', 'auto_model_MDX',  
'auto_model_ML350', 'auto_model_Malibu', 'auto_model_Maxima', 'auto_model_Neon',  
'auto_model_Passat', 'auto_model_Pathfinder', 'auto_model_RAM', 'auto_model_RSX',
```

```
'auto_model_Silverado', 'auto_model_TL', 'auto_model_Tahoe', 'auto_model_Ultima',  
'auto_model_Wrangler', 'auto_model_X5', 'auto_model_X6']
```

```
df = pd.DataFrame(0, columns= processed_features, index= [0])
```

```
df
```

```
scaler = scaler_obj['normal_scaler']
```

```
col_to_scale = scaler_obj['cols_to_scale']
```

```
input_data = {
```

```
    'policy_deductable': 1000,
```

```
    'policy_annual_premium' : 1406.91,
```

```
    'umbrella_limit' : 0,
```

```
    'insured_occupation' : 'craft-repair',
```

```
    'insured_hobbies' : 'sleeping',
```

```
    'capital-gains' : 53300,
```

```
    'capital-loss' : 0,
```

```
    'incident_severity' : 'Major Damage',
```

```
    'incident_hour_of_the_day' : 5,
```

```
    'total_claim_amount' : 71610,
```

```
    'auto_model' : '92x',
```

```
    'auto_year' : 2004,
```

```
    'fraud_reported' : 1,
```

```
    'day': 25
```

```
}
```

```
...
```

PROMPT 8 (Copilot): Streamlit Application

Generate a Streamlit Script for the Most Viable Product:

- Title: "Insurance Claim Risk Predictor"
- Input Variables: 'policy_deductible', 'policy_annual_premium', 'umbrella_limit', 'insured_occupation', 'insured_hobbies', 'capital-gains', 'capital-loss', 'incident_severity', 'incident_hour_of_the_day', 'total_claim_amount', 'auto_model', 'auto_year', 'day'
- options for 'insured_occupation': craft-repair, sales, armed-forces, tech-support, prof-specialty, other-service, priv-house-serv, exec-managerial, protective-serv, machine-op-inspct, transport-moving, handlers-cleaners, adm-clerical, farming-fishing. Ensure options are sorted.
- options for 'insured_hobbies': sleeping, board-games, bungee-jumping, base-jumping, golf, camping, dancing, skydiving, reading, movies, hiking, yachting, paintball, chess, kayaking, polo, basketball, video-games, cross-fit, exercise. Ensure options are sorted.
- options for 'incident_severity': Minor Damage, Major Damage, Total Loss
- options for 'incident_hour_of_the_day': 1 to 24 as dropdown
- options for 'auto_model': 92x, RAM, Tahoe, 95, Pathfinder, A5, Camry, F150, A3, Neon, MDX, Maxima, Legacy, TL, Impreza, RSX, Forrester, Escape, Corolla, 3 Series, C300, Wrangler, M5, X5, E400, Highlander, Civic, Silverado, CRV, 93, Accord, X6, Malibu, Fusion, ML350, Passat, Ultima, Jetta, Grand Cherokee. Ensure options are sorted.
- options for 'auto_year': 1995 to 2015 as dropdown
- options for 'day': 0 to 6 as dropdown, use Mon to Sun drop down and transform it using CASE to store in variable
- assign the variables to the respective keys in dictionary of input data
- design input in a grid
- "Calculate Risk" button to execute prediction script once the input data is passed
- display Fraud Claim Probability: {probability:.2%}