

POST 2 - GEN AI

MASTERING CHATGPT FOR EVERYDAY PYTHON CODING IN DATA SCIENCE AND ADVANCED DATA ANALYSIS TASKS

10 Real World Use Cases

**GENERATIVE AI
FOR ALL - PROMPT
ENGINEERING**

 **DINESHLAL**



DINESH LAL
(DIRECTOR, DATA SCIENCE)



1

“WRITE A PYTHON PROGRAM TO IMPORT CSV DATA SET AS A PANDAS DATAFRAME. ALSO PRINT SOME SUMMARIES POST IMPORTING THE FILE.”

```
import pandas as pd

# Replace 'your_dataset.csv' with the actual path to your CSV file
csv_file_path = 'your_dataset.csv'

# Import CSV data into a pandas DataFrame
df = pd.read_csv(csv_file_path)

# Display the first few rows of the DataFrame
print("First 5 rows of the DataFrame:")
print(df.head())

# Display basic information about the DataFrame
print("\nDataFrame information:")
print(df.info())

# Display summary statistics of the numeric columns
print("\nSummary statistics:")
print(df.describe())

# Display the column names in the DataFrame
print("\nColumn names:")
print(df.columns)
```

2

“WRITE A PYTHON PROGRAM TO GENERATE DERIVED FEATURES ON A DATAFRAME NAME TABLE1 HAVING 20 INDEPENDENT VARIABLES. THE DEPENDENT VARIABLE NAME IS "TARGET". PROGRAM SHOULD CATER CATEGORICAL AND CONTINUOUS VARIABLE SEPARATELY”

```
python Copy code  
  
import pandas as pd  
from sklearn.preprocessing import PolynomialFeatures  
  
# Assuming you have a DataFrame named TABLE1 with 20 independent variab  
# Replace the following code with the actual columns in your DataFrame  
  
# Example DataFrame  
data = {  
    'Var1': [1, 2, 3, 4, 5],  
    'Var2': ['A', 'B', 'A', 'B', 'A'],  
    'Var3': [10.1, 20.2, 30.3, 40.4, 50.5],  
    # Add other variables...  
    'Target': [0, 1, 0, 1, 0]  
}  
  
TABLE1 = pd.DataFrame(data)  
  
# Separate categorical and continuous variables
```

CLICK TO SEE FULL CODE -
[HTTPS://CHAT.OPENAI.COM/SHARE/04D55CB7-0452-4896-972B-7E1515D019D3](https://chat.openai.com/share/04d55cb7-0452-4896-972b-7e1515d019d3)

3

“CREATE A PYTHON CODE TO IMPUTE MISSING VALUE IN ALL THE CONTINUOUS VARIABLES. THE DATASET HAS DEPENDENT VARIABLE NAMED AS “TARGET” REST ALL ARE INDEPENDENT VARIABLE.”

```
python Copy code  
  
import pandas as pd  
from sklearn.impute import SimpleImputer  
  
# Assuming your dataset is stored in a DataFrame called 'df'  
# Replace 'your_dataset.csv' with the actual file name or provide your  
  
# If reading from a CSV file:  
# df = pd.read_csv('your_dataset.csv')  
  
# Sample DataFrame  
data = {'Target': [1, 0, 1, 0, 1],  
        'Var1': [23, 35, 18, None, 42],  
        'Var2': [54, 27, None, 39, 62],  
        'Var3': [None, 15, 28, 21, 32]}  
df = pd.DataFrame(data)  
  
# Separate the target variable and independent variables  
X = df.drop('Target', axis=1)  
y = df['Target']
```

CLICK TO SEE FULL CODE -
[HTTPS://CHAT.OPENAI.COM/SHARE/6D55](https://chat.openai.com/share/6d5533e5-ca23-49fc-b90e-7ffa5acfd17c)
[33E5-CA23-49FC-B90E-7FFA5ACFD17C](https://chat.openai.com/share/6d5533e5-ca23-49fc-b90e-7ffa5acfd17c)

4

“I HAVE A DATAFRAME WITH 200 VARIABLES. CAN YOU PLEASE WRITE A CODE TO DESCRIBE THE FEATURES AND VISUALIZE SOME UNIVARIATE DISTRIBUTION. I WANT TO SPECIFICALLY SEE, NUMBER OF MISSING VALUES, MEAN, MEDIAN, STANDARD DEVIATION, AND QUARTILE VALUES FOR EACH VARIABLE. I WANT EXPORT THE GRAPHS IN A PPT FILE.

”

```
python Copy code  
  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
from pptx import Presentation  
from pptx.util import Inches  
import numpy as np  
  
# Load your DataFrame  
# Assuming your DataFrame is named 'df'  
# Replace 'your_data.csv' with the actual path or URL of your data file  
df = pd.read_csv('your_data.csv')  
  
# Create a PowerPoint presentation  
presentation = Presentation()  
  
# Function to create a slide with a title and content  
def add_slide(prs, title, content):  
    slide_layout = prs.slide_layouts[1] # Use the layout with title and content  
    slide = prs.slides.add_slide(slide_layout)  
    title_placeholder = slide.placeholders[0]  
    content_placeholder = slide.placeholders[1]  
    title_placeholder.text = title
```

CLICK TO SEE FULL CODE -
[HTTPS://CHAT.OPENAI.COM/SHARE/78E08928-A79F-47CE-B99C-3C78F60F1B69](https://chat.openai.com/share/78e08928-a79f-47ce-b99c-3c78f60f1b69)

5

"

WRITE A PYTHON CODE TO CALCULATE CORRELATION MATRIX AMONG ALL VARIABLES WITH EACH OTHER. PLEASE NOTE THAT THERE ARE SOME CATEGORICAL VARIABLES AND SOME CONTINUOUS VARIABLES.

"

python

Copy code

```
import pandas as pd

# Sample DataFrame with both categorical and continuous variables
data = {
    'Category1': ['A', 'B', 'A', 'B', 'A'],
    'Category2': ['X', 'Y', 'Y', 'X', 'X'],
    'Numeric1': [10, 20, 15, 25, 30],
    'Numeric2': [5, 15, 10, 20, 25]
}

df = pd.DataFrame(data)

# Convert categorical variables to numerical using one-hot encoding
df_encoded = pd.get_dummies(df, columns=['Category1', 'Category2'])

# Concatenate the encoded DataFrame with the original DataFrame
df_final = pd.concat([df, df_encoded], axis=1)

# Calculate the correlation matrix
correlation_matrix = df_final.corr()

# Display the correlation matrix
print(correlation_matrix)
```

6

“I HAVE 100 INDEPENDENT VARIABLES IN A DATAFRAME AND A DEPENDENT VARIABLE NAMED "DEPENDENT_VAR". CAN YOU PLEASE WRITE A PYTHON CODE TO CREATE BIVARIATE DISTRIBUTION BETWEEN DEPENDENT VARIABLE AND EACH INDEPENDENT VARIABLE (SEPARATE SECTION FOR CONTINUOUS, AND CATEGORICAL VARIABLE). ALSO I WANT A VARIABLE IMPORTANCE TABLE.”

```
python Copy code  
  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.ensemble import RandomForestRegressor  
from sklearn.feature_selection import mutual_info_regression  
  
# Assuming your DataFrame is named df  
# Replace 'Dependent_var' with the actual column name of your dependent  
dependent_var = 'Dependent_var'  
  
# Separate continuous and categorical variables  
continuous_vars = df.select_dtypes(include=['float64', 'int64']).columns  
categorical_vars = df.select_dtypes(include=['object']).columns  
  
# Bivariate distribution for continuous variables  
for var in continuous_vars:  
    plt.scatter(df[var], df[dependent_var])  
    plt.xlabel(var)  
    plt.ylabel(dependent_var)  
    plt.title(f'Bivariate Distribution: {var} vs {dependent_var}')  
    plt.show()
```

CLICK TO SEE FULL CODE -
[HTTPS://CHAT.OPENAI.COM/SHARE/D4A40E2D-8DAB-401A-B4A4-4545809101E0](https://chat.openai.com/share/d4a40e2d-8dab-401a-b4a4-4545809101e0)

7

“CREATE A PYTHON CODE TO FIT LOGISTIC REGRESSION MODEL ON A DATAFRAME WITH DEPENDENT VARIABLE NAMED AS "DEPENDENT_VAR" AND INDEPENDENT VARIABLES ['SPEND', 'AGE', 'TOTAL_SALES', 'DUMMY1', 'DUMMY2']. ALSO GENERATE SOME METRICS TO EVALUATE MODEL PERFORMANCE, PREFERABLY ON CHARTS.”

```
python Copy code  
  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.model_selection import train_test_split  
from sklearn.linear_model import LogisticRegression  
from sklearn.metrics import confusion_matrix, classification_report, roc_auc_score  
  
# Assuming your dataframe is named df  
# Make sure to replace 'YourDataset.csv' with your actual dataset  
# Also, make sure the dependent variable is named "Dependent_Var"  
# and independent variables are ['Spend', 'Age', 'Total_sales', 'Dummy1', 'Dummy2']  
  
# Example DataFrame creation (replace this with your actual DataFrame)  
data = {  
    'Spend': [100, 200, 150, 300, 250],  
    'Age': [25, 30, 28, 35, 32],  
    'Total_sales': [500, 700, 600, 800, 750],  
    'Dummy1': [0, 1, 1, 0, 1],  
    'Dummy2': [1, 0, 1, 0, 1],  
    'Dependent_Var': [0, 1, 1, 0, 1]  
}
```

CLICK TO SEE FULL CODE -
[HTTPS://CHAT.OPENAI.COM/SHARE/28E4861A-3D49-410F-8ED1-7F7DAE698916](https://chat.openai.com/share/28E4861A-3D49-410F-8ED1-7F7DAE698916)

8

"

CREATE A PYTHON PROGRAM TO CREATE ENSEMBLE OF MULTIPLE MACHINE LEARNING MODELS. ALSO ADD SECTION TO VALIDATE THE MODEL PERFORMANCE ON TEST DATA, WHEN COMPARING INDIVIDUAL MODELS WITH ENSEMBLE MODELS.

"

```
python Copy code

import numpy as np
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Load the Iris dataset as an example
data = load_iris()
X = data.data
y = data.target

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Define individual machine learning models
model1 = RandomForestClassifier(random_state=42)
model2 = GradientBoostingClassifier(random_state=42)
model3 = LogisticRegression(random_state=42)
```

CLICK TO SEE FULL CODE -
[HTTPS://CHAT.OPENAI.COM/SHARE/42166916-1606-4328-8FB8-7C50AD187FD4](https://chat.openai.com/share/42166916-1606-4328-8fb8-7c50ad187fd4)

9

“
WRITE A PYTHON SCRIPT TO
EXPORT A LOGISTIC
REGRESSION MODEL, AND A
CODE TO SCORE A NEW DATA.
”

```
python Copy code  
  
# Import necessary libraries  
import pandas as pd  
from sklearn.model_selection import train_test_split  
from sklearn.linear_model import LogisticRegression  
from sklearn.metrics import accuracy_score  
from sklearn.externals import joblib # For scikit-learn version 0.22 a  
# If you're using scikit-learn version 0.23 and above, use:  
# from joblib import dump, load  
  
# Sample data for demonstration  
data = {'feature1': [1, 2, 3, 4, 5],  
        'feature2': [2, 3, 4, 5, 6],  
        'target': [0, 0, 1, 1, 1]}  
df = pd.DataFrame(data)  
  
# Split data into features (X) and target variable (y)  
X = df[['feature1', 'feature2']]  
y = df['target']  
  
# Split the data into training and testing sets  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2
```

CLICK TO SEE FULL CODE -
[HTTPS://CHAT.OPENAI.COM/SHARE/DC46](https://chat.openai.com/share/DC46OCF1-C75D-49BC-B161-C92C571B5832)
[OCF1-C75D-49BC-B161-C92C571B5832](https://chat.openai.com/share/DC46OCF1-C75D-49BC-B161-C92C571B5832)

10

“WRITE A DETAILED PYTHON CODE FOR A NEW DATA SCIENTIST TO CREATE A LOGISTIC REGRESSION MODEL. KEEP ALL THE IMPORTANT SECTION LIKE DATA IMPORT, DATA EXPLORATION, VARIABLE IMPORTANCE, VARIABLE REDUCTION, MODEL BUILDING, MODEL SCORING, AND EXPORTING THE RESULTS. ADD ADDITIONAL SECTIONS IF I MISSED. THERE SHOULD BE DETAILED COMMENTS IN EACH SECTION EXPLAINING WHAT THE CODE IS DOING.”

```
python Copy code  
  
# Import necessary libraries  
import pandas as pd  
import numpy as np  
from sklearn.model_selection import train_test_split  
from sklearn.preprocessing import StandardScaler  
from sklearn.linear_model import LogisticRegression  
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report  
import matplotlib.pyplot as plt  
  
# Set random seed for reproducibility  
np.random.seed(42)  
  
# Data Import  
# Assuming the data is in a CSV file named 'your_data.csv'  
data = pd.read_csv('your_data.csv')  
  
# Data Exploration  
# Display the first few rows of the dataset  
print("First few rows of the dataset:")  
print(data.head())  
  
↓
```

CLICK TO SEE FULL CODE -
[HTTPS://CHAT.OPENAI.COM/SHARE/011E99FB-DB3C-48DB-817C-F7C77E02C2E9](https://chat.openai.com/share/011e99fb-db3c-48db-817c-f7c77e02c2e9)

Thank You