Assignment No-4

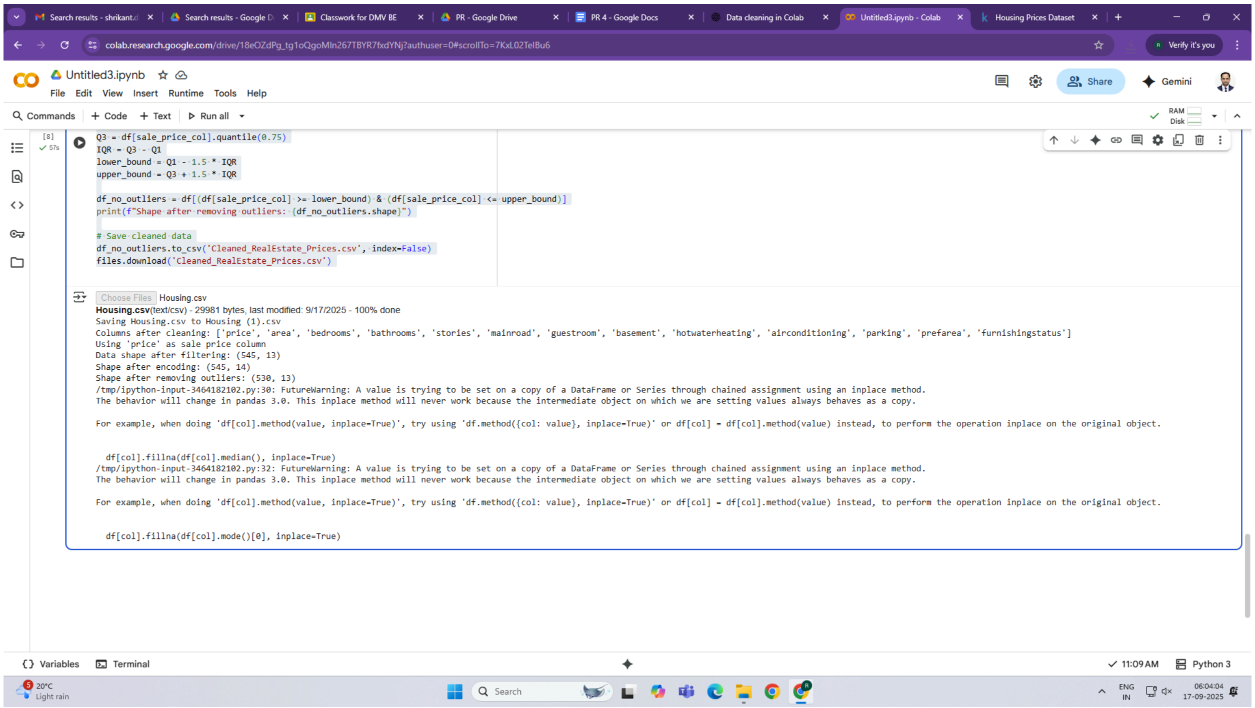
CODE:-

```
import pandas as pd
import numpy as np
from google.colab import files

# Upload file
uploaded = files.upload()
filename = next(iter(uploaded))

# Load dataset
df = pd.read_csv(filename)

# Clean columns
df.columns = df.columns.str.strip().str.lower().str.replace(' ',
'_').str.replace(r'[^\w\s]', '', regex=True)
print("Columns after cleaning:", df.columns.tolist())

# Find sale price column
sale_price_col = None
for col in df.columns:
    if 'price' in col:
        sale_price_col = col
        print(f"Using '{col}' as sale price column")
        break

if sale_price_col is None:
    raise ValueError("No column related to 'price' found in the dataset!")

# Handle missing values
for col in df.columns:
    if df[col].dtype in ['float64', 'int64']:
        df[col].fillna(df[col].median(), inplace=True)
    else:
        df[col].fillna(df[col].mode()[0], inplace=True)
```

```
# Filter example
if 'sale_date' in df.columns:
    df['sale_date'] = pd.to_datetime(df['sale_date'], errors='coerce')
    df = df[(df['sale_date'].dt.year >= 2018) & (df['sale_date'].dt.year
<= 2023)]

if 'property_type' in df.columns:
    df = df[df['property_type'].str.lower() == 'single family']

print(f"Data shape after filtering: {df.shape}")

# Encode categoricals
cat_cols = df.select_dtypes(include=['object']).columns.tolist()
df_encoded = pd.get_dummies(df, columns=cat_cols, drop_first=True)

print(f"Shape after encoding: {df_encoded.shape}")

# Aggregate avg sale price by neighborhood
if 'neighborhood' in df.columns:
    avg_price_by_neighborhood =
df.groupby('neighborhood')[sale_price_col].mean().reset_index().rename(col
umns={sale_price_col: 'avg_sale_price'})
    print(avg_price_by_neighborhood.head())

# Aggregate avg sale price by property type
if 'property_type' in df.columns:
    avg_price_by_property_type =
df.groupby('property_type')[sale_price_col].mean().reset_index().rename(co
lumns={sale_price_col: 'avg_sale_price'})
    print(avg_price_by_property_type.head())

# Outlier handling
Q1 = df[sale_price_col].quantile(0.25)
Q3 = df[sale_price_col].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
```

```
df_no_outliers = df[(df[sale_price_col] >= lower_bound) &
(df[sale_price_col] <= upper_bound)]
print(f"Shape after removing outliers: {df_no_outliers.shape}")

# Save cleaned data
df_no_outliers.to_csv('Cleaned_RealEstate_Prices.csv', index=False)
files.download('Cleaned_RealEstate_Prices.csv')
```

OUTPUT:-


colab.research.google.com/drive/18eOZdPg_tg1oQgoMln267TBYR7fxdYNj?authuser=0#scrollTo=7KxL02TelBu6
Untitled3.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
Gemini
Variables Terminal
11:09 AM Python 3

```
Q3 = df[sale_price_col].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

df_no_outliers = df[(df[sale_price_col] >= lower_bound) & (df[sale_price_col] <= upper_bound)]
print(f"Shape after removing outliers: {df_no_outliers.shape}")

# Save cleaned data
df_no_outliers.to_csv('Cleaned_RealEstate_Prices.csv', index=False)
files.download('Cleaned_RealEstate_Prices.csv')
```

```
Choose Files Housing.csv
Housing.csv(text/csv) - 29981 bytes, last modified: 9/17/2025 - 100% done
Saving Housing.csv to Housing (1).csv
Columns after cleaning: ['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad', 'guestroom', 'basement', 'hotwaterheating', 'airconditioning', 'parking', 'prefarea', 'furnishingstatus']
Using 'price' as sale price column
Data shape after filtering: (545, 13)
Shape after encoding: (545, 14)
Shape after removing outliers: (530, 13)
/tmp/ipython-input-3464182102.py:30: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df[col].fillna(df[col].median(), inplace=True)
/tmp/ipython-input-3464182102.py:32: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df[col].fillna(df[col].mode()[0], inplace=True)
```