exp4

August 16, 2024

[ ]:

```
import os
import pandas as pd
import numpy as np
from datetime import datetime
```

[ ]:

```
df = pd.read_csv('Housing.csv')
```

[ ]:

```
# 1: Handling Missing Values
numerical_cols = ['price', 'area', 'bedrooms', 'bathrooms', 'stories',␣
 ↪'parking']
df[numerical_cols] = df[numerical_cols].fillna(df[numerical_cols].mean())
```

[ ]:

```
# For categorical columns: Fill missing values with the mode
categorical_cols = ['mainroad', 'guestroom', 'basement', 'hotwaterheating',
                    'airconditioning', 'prefarea', 'furnishingstatus']
df[categorical_cols] = df[categorical_cols].fillna(df[categorical_cols].mode().
 ↪iloc[0])
```

[ ]:

```
# 2: Encoding Categorical Variables with one-hot encoding
for col in categorical_cols:
    dummies = pd.get_dummies(df[col], prefix=col, drop_first=True)
    df = pd.concat([df, dummies], axis=1)
    df.drop(col, axis=1, inplace=True)
```

[ ]:

```
# 3: Scaling Numerical Features
numerical_cols = ['price', 'area', 'bedrooms', 'bathrooms', 'stories',␣
 ↪'parking']
df[numerical_cols] = (df[numerical_cols] - df[numerical_cols].mean()) /␣
 ↪df[numerical_cols].std()
```

[ ]:

```
# 4: Feature Engineering
df['total_rooms'] = df['bedrooms'] + df['bathrooms']
```

[ ]:

```
# 5: Removing Duplicates
df.drop_duplicates(inplace=True)
```

[ ]:

```
# 6: Handling Outliers
for col in numerical_cols:
    percentile_95 = df[col].quantile(0.95)
    df[col] = np.where(df[col] > percentile_95, percentile_95, df[col])
```

[ ]:

```
# 7: Normalization
df[numerical_cols] = (df[numerical_cols] - df[numerical_cols].min()) /␣
 ↪(df[numerical_cols].max() - df[numerical_cols].min())
```

[ ]:

```
# 8: Binning
df['area_binned'] = pd.cut(df['area'], bins=3, labels=["small", "medium",␣
 ↪"large"])
```

[ ]:

```
# 9: Feature Selection
selected_features = ['area', 'bathrooms', 'stories', 'total_rooms',␣
 ↪'area_binned']
df_selected = df[selected_features]
```

[ ]:

```
# Final check of the preprocessed data
print(df_selected.head())
```

```
       area  bathrooms   stories  total_rooms area_binned
0  0.785034        1.0  0.666667     2.822638       large
1  0.994558        1.0  1.000000     6.802978       large
2  1.000000        1.0  0.333333     1.467742       large
3  0.795918        1.0  0.333333     2.822638       large
4  0.785034        0.0  0.333333     0.832468       large
```

[ ]:

```
# Step 11: Saving to a New CSV File
base_filename = "preprocessed_Housing"
file_extension = ".csv"
timestamp = datetime.now().strftime("%Y%m%d_%H%M%S")
full_filename = f"{base_filename}_{timestamp}{file_extension}"

df_selected.to_csv(full_filename, index=False)

print(f"Preprocessed data saved to {full_filename}")
```

```
Preprocessed data saved to preprocessed_Housing_20240816_210117.csv
```