DOPP_preparation_for_ml

January 22, 2025

[13]:
```
import pandas as pd
import os
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_selection import SelectKBest, f_regression
from sklearn.model_selection import train_test_split
```

[25]:
```
cwd = os.getcwd()
df = pd.read_csv(os.getcwd()+'/preprossed data/preprocessed_data.csv')
df['days_rented'] = (pd.to_datetime(df['end_date']) - pd.
 ↪to_datetime(df['start_date'])).dt.days
df
```

[25]:

	listing_id	price	min_nights	max_nights	host_since	\
0	40625	164.0	1	365	2010-07-20	
1	40625	164.0	1	365	2010-07-20	
2	40625	164.0	1	365	2010-07-20	
3	40625	164.0	1	365	2010-07-20	
4	40625	164.0	1	365	2010-07-20	
…	…	…	…	…		
534014	1242111585356854473	999.0	5	179	2022-03-11	
534015	1242111585356854473	999.0	5	179	2022-03-11	
534016	1242111585356854473	999.0	5	179	2022-03-11	
534017	1242111585356854473	999.0	5	179	2022-03-11	
534018	1242111585356854473	999.0	5	179	2022-03-11	

	host_response_rate	host_acceptance_rate	accommodates	bathrooms	\
0	1.0	0.0	4	1.0	
1	1.0	0.0	4	1.0	
2	1.0	0.0	4	1.0	
3	1.0	0.0	4	1.0	
4	1.0	0.0	4	1.0	
…	…	…	…	…	
534014	0.0	0.0	3	1.0	
534015	0.0	0.0	3	1.0	
534016	0.0	0.0	3	1.0	
534017	0.0	0.0	3	1.0	

```
534018                0.0                   0.0             3        1.0

        bedrooms  … other  park  shops_and_retail  supermarket  \
0            2.0  …     1     1                 1            1
1            2.0  …     1     1                 1            1
2            2.0  …     1     1                 1            1
3            2.0  …     1     1                 1            1
4            2.0  …     1     1                 1            1
…          …  …   …   …               …          …
534014       1.0  …     1     1                 1            1
534015       1.0  …     1     1                 1            1
534016       1.0  …     1     1                 1            1
534017       1.0  …     1     1                 1            1
534018       1.0  …     1     1                 1            1

        transport_and_infrastructure  Essentials  Hair_dryer  Kitchen  Wifi  \
0                                  1           1           1        1     1
1                                  1           1           1        1     1
2                                  1           1           1        1     1
3                                  1           1           1        1     1
4                                  1           1           1        1     1
…                                …         …         …       …   …
534014                             1           1           1        1     1
534015                             1           1           1        1     1
534016                             1           1           1        1     1
534017                             1           1           1        1     1
534018                             1           1           1        1     1

        days_rented
0                 6
1                 6
2                 6
3                 6
4                 6
…               …
534014            6
534015            6
534016            6
534017            6
534018            6

[534019 rows x 69 columns]
```

[26]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 534019 entries, 0 to 534018
```

```
Data columns (total 69 columns):
 #   Column                                Non-Null Count   Dtype
---  ------                                --------------   -----
 0   listing_id                            534019 non-null  int64
 1   price                                 534019 non-null  float64
 2   min_nights                            534019 non-null  int64
 3   max_nights                            534019 non-null  int64
 4   host_since                            534019 non-null  object
 5   host_response_rate                    534019 non-null  float64
 6   host_acceptance_rate                  534019 non-null  float64
 7   accommodates                          534019 non-null  int64
 8   bathrooms                             534019 non-null  float64
 9   bedrooms                              534019 non-null  float64
 10  amenities                             532005 non-null  object
 11  number_of_reviews                     534019 non-null  int64
 12  review_scores_rating                  534019 non-null  float64
 13  review_scores_accuracy                534019 non-null  float64
 14  review_scores_cleanliness             534019 non-null  float64
 15  review_scores_checkin                 534019 non-null  float64
 16  review_scores_communication           534019 non-null  float64
 17  review_scores_location                534019 non-null  float64
 18  review_scores_value                   534019 non-null  float64
 19  host_has_listings                     534019 non-null  int64
 20  reviews_per_month                     534019 non-null  float64
 21  nearest_stop_distance_km              534019 non-null  float64
 22  start_date                            534019 non-null  object
 23  end_date                              534019 non-null  object
 24  host_response_time_within_a_day       534019 non-null  bool
 25  host_response_time_within_a_few_hours 534019 non-null  bool
 26  host_response_time_within_an_hour     534019 non-null  bool
 27  is_superhost                          534019 non-null  bool
 28  host_has_profile_pic                  534019 non-null  bool
 29  host_identity_verified                534019 non-null  bool
 30  neighbourhood_Brigittenau             534019 non-null  bool
 31  neighbourhood_Dobling                 534019 non-null  bool
 32  neighbourhood_Donaustadt              534019 non-null  bool
 33  neighbourhood_Favoriten               534019 non-null  bool
 34  neighbourhood_Floridsdorf             534019 non-null  bool
 35  neighbourhood_Hernals                 534019 non-null  bool
 36  neighbourhood_Hietzing                534019 non-null  bool
 37  neighbourhood_Innere_Stadt            534019 non-null  bool
 38  neighbourhood_Josefstadt              534019 non-null  bool
 39  neighbourhood_Landstra§e              534019 non-null  bool
 40  neighbourhood_Leopoldstadt            534019 non-null  bool
 41  neighbourhood_Liesing                 534019 non-null  bool
 42  neighbourhood_Margareten              534019 non-null  bool
 43  neighbourhood_Mariahilf               534019 non-null  bool
 44  neighbourhood_Meidling                534019 non-null  bool
```

```
 45  neighbourhood_Neubau                       534019 non-null  bool
 46  neighbourhood_Ottakring                    534019 non-null  bool
 47  neighbourhood_Penzing                      534019 non-null  bool
 48  neighbourhood_Rudolfsheim-Funfhaus         534019 non-null  bool
 49  neighbourhood_Simmering                    534019 non-null  bool
 50  neighbourhood_Wahring                      534019 non-null  bool
 51  neighbourhood_Wieden                       534019 non-null  bool
 52  property_type_Other                        534019 non-null  bool
 53  property_type_Private_room                 534019 non-null  bool
 54  property_type_Room_in_hotel_or_similar     534019 non-null  bool
 55  property_type_Shared_room                  534019 non-null  bool
 56  cafe_restaurant                            534019 non-null  int64
 57  education                                  534019 non-null  int64
 58  entertainment_leisure                      534019 non-null  int64
 59  other                                      534019 non-null  int64
 60  park                                       534019 non-null  int64
 61  shops_and_retail                           534019 non-null  int64
 62  supermarket                                534019 non-null  int64
 63  transport_and_infrastructure               534019 non-null  int64
 64  Essentials                                 534019 non-null  int64
 65  Hair_dryer                                 534019 non-null  int64
 66  Kitchen                                    534019 non-null  int64
 67  Wifi                                       534019 non-null  int64
 68  days_rented                                534019 non-null  int64
dtypes: bool(32), float64(14), int64(19), object(4)
memory usage: 167.0+ MB
```

[27]:
```
df = df.drop(columns=['start_date', 'end_date',
                                    'host_since',
                                    'amenities'])
```

[28]:
```
# Columns to exclude
exclude_columns = ['price', 'days_rented']
# Get all columns except the excluded ones
df['income_per_period_rented'] = df['price'] * df['days_rented']
df = df.drop(exclude_columns, axis=1)
aggregated_df = df.groupby('listing_id').
 ↪agg(total_income=('income_per_period_rented', 'sum')).reset_index()

# Merge the aggregated result back with the original dataframe to retain other␣
 ↪columns
df = pd.merge(df, aggregated_df, on='listing_id', how='left').
 ↪drop_duplicates(subset=['listing_id'])
df
```

[28]:

	listing_id	min_nights	max_nights	host_response_rate	\
0	40625	1	365	1.0	

```
53                    51287          25        1125                 1.0
106                   78416           1        1125                 0.0
159                   90247           1        1125                 1.0
212                  109679           1         365                 1.0
…                       …          …          …                   …
533754  1241959973134398458           3          90                 1.0
533807  1241979933051239847           1         179                 0.0
533860  1241983222351205737           1         179                 0.0
533913  1242077281163770800           1         365                 0.0
533966  1242111585356854473           1         179                 0.0

        host_acceptance_rate  accommodates  bathrooms  bedrooms  \
0                        0.0             4        1.0       2.0
53                       1.0             2        1.0       0.0
106                      0.0             4        1.0       1.0
159                      1.0             4        1.0       1.0
212                      0.0             4        1.0       0.0
…                        …             …          …         …
533754                   0.0             4        1.0       1.0
533807                   0.0             4        1.5       1.0
533860                   0.0             2        1.0       1.0
533913                   0.0             4        1.0       1.0
533966                   0.0             3        1.0       1.0

        number_of_reviews  review_scores_rating  …  park  shops_and_retail  \
0                     217                  4.86  …     1                 1
53                    380                  4.67  …     0                 1
106                   178                  4.35  …     1                 1
159                   773                  4.85  …     1                 1
212                   143                  4.87  …     1                 1
…                       …                     …  …   …                 …
533754                  0                  0.00  …     0                 1
533807                  0                  0.00  …     1                 1
533860                  0                  0.00  …     1                 1
533913                  0                  0.00  …     0                 1
533966                  0                  0.00  …     1                 1

        supermarket  transport_and_infrastructure  Essentials  Hair_dryer  \
0                 1                             1           1           1
53                1                             1           1           1
106               1                             1           1           1
159               1                             1           1           1
212               1                             1           1           1
…               …                             …           …           …
533754            1                             1           0           0
533807            1                             1           1           1
533860            1                             1           1           1
```

```
533913            1                              1           0           0
533966            1                              1           1           1

        Kitchen  Wifi  income_per_period_rented  total_income
0             1     1                     984.0       52152.0
53            1     1                     480.0       25440.0
106           1     1                     360.0       19080.0
159           1     1                     768.0       40704.0
212           1     1                     636.0       33708.0
…         …    …                         …           …
533754        1     1                     870.0       46110.0
533807        1     0                    5994.0      317682.0
533860        1     0                    5994.0      317682.0
533913        1     1                     540.0       28620.0
533966        1     1                    5994.0      317682.0

[10076 rows x 65 columns]
```

[ ]:

[29]:
```
df.corr()['total_income'].sort_values(ascending = False)
```

[29]:
```
total_income                        1.000000
income_per_period_rented            1.000000
listing_id                          0.171549
host_has_listings                   0.131129
host_response_time_within_an_hour   0.123338
                                        …
host_identity_verified             -0.059474
is_superhost                       -0.060068
property_type_Private_room         -0.063742
Kitchen                            -0.095938
other                                    NaN
Name: total_income, Length: 65, dtype: float64
```

[30]:
```
X = df.drop(['listing_id','total_income'], axis=1)
y = df['total_income']

X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5,␣
 ↪random_state=42)
```

[31]:
```
# Selecting K-Best features

selector = SelectKBest(score_func=f_regression, k=15)
X_selected = selector.fit_transform(X_train, y_train)
```

```
# Get selected feature names
selected_features = X_train.columns[selector.get_support()]
print("Selected features:", selected_features)
```

```
Selected features: Index(['host_response_rate', 'accommodates', 'bathrooms',
'bedrooms',
       'number_of_reviews', 'host_has_listings', 'reviews_per_month',
       'host_response_time_within_an_hour', 'is_superhost',
       'host_identity_verified', 'neighbourhood_Margareten',
       'property_type_Private_room', 'Hair_dryer', 'Kitchen',
       'income_per_period_rented'],
      dtype='object')
```

[32]:
```
# leaving features with the highest score
X_train = X_train[list(selected_features)]
X_val = X_val[list(selected_features)]
X_test = X_test[list(selected_features)]
```

[33]:
```
from sklearn.model_selection import cross_val_score
from sklearn.metrics import mean_squared_error, r2_score
from xgboost import XGBRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor

# Define models
models = {
    'Random Forest Regressor': RandomForestRegressor(),
    'Gradient Boosting Regressor': GradientBoostingRegressor(),
    'XGBoost Regressor': XGBRegressor()
}


for name, model in models.items():
    # Cross-validation
    cv_scores = cross_val_score(model, X_train, y_train, cv=5,␣
 ↪scoring='neg_mean_squared_error')
    mean_mse = -cv_scores.mean()
    std_mse = cv_scores.std()

    # Fit the model
    model.fit(X_train, y_train)

    # Predict on the validation set
    y_val_pred = model.predict(X_val)

    # Predict on the test set
    y_test_pred = model.predict(X_test)
```

```
    # Calculate metrics for validation set
    val_mse = mean_squared_error(y_val, y_val_pred)
    val_r2 = r2_score(y_val, y_val_pred)

    # Calculate metrics for test set
    test_mse = mean_squared_error(y_test, y_test_pred)
    test_r2 = r2_score(y_test, y_test_pred)

    # Print results
    print(f"{name}:")
    print(f"Cross-Validation Mean MSE: {mean_mse:.4f} ± {std_mse:.4f}")
    print(f"Validation Set MSE: {val_mse:.4f}")
    print(f"Validation Set R-squared: {val_r2:.4f}")
    print(f"Test Set MSE: {test_mse:.4f}")
    print(f"Test Set R-squared: {test_r2:.4f}\n")
```

```
Random Forest Regressor:
Cross-Validation Mean MSE: 1983943987.7967 ± 3858476066.6825
Validation Set MSE: 210633899.2733
Validation Set R-squared: 0.9956
Test Set MSE: 2646241571.0167
Test Set R-squared: 0.9770

Gradient Boosting Regressor:
Cross-Validation Mean MSE: 1952824223.5472 ± 3763417025.1298
Validation Set MSE: 1744937370.4412
Validation Set R-squared: 0.9638
Test Set MSE: 4198166025.4762
Test Set R-squared: 0.9636

XGBoost Regressor:
Cross-Validation Mean MSE: 3294130320.0824 ± 4369658406.4894
Validation Set MSE: 1955629254.3117
Validation Set R-squared: 0.9594
Test Set MSE: 9256070715.6216
Test Set R-squared: 0.9197
```

[ ]:

[ ]:

[ ]: