regression_tree

November 29, 2023

[ ]:
```
from sklearn.ensemble import RandomForestRegressor
import pandas as pd
import numpy as np
import sqlite3
import logging
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import RidgeCV, LassoCV
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
import itertools
import statsmodels.api as sm
from multiprocessing import Pool
import seaborn as sns
import matplotlib.pyplot as plt
import random
```

[ ]:
```
logging.basicConfig(filename='logs.log', level=logging.DEBUG,
                    format='%(asctime)s:%(levelname)s:%(message)s')

conn = sqlite3.connect('../data_eng/HOUSING.db')
cursor = conn.cursor()
df = pd.read_sql("select * from Realtor_Final_Merged", conn)


cursor.close()
conn.close()

df = df.select_dtypes(include=[np.number])

df.head(2)
```

[ ]:

	month_date_yyyymm	median_listing_price	active_listing_count	\
0	202309	635000.0	53.0	
1	202309	399999.0	2.0	

```
   median_days_on_market  new_listing_count  price_increased_count  \
0                   39.0               36.0                    0.0
1                  195.0                0.0                    0.0

   price_reduced_count  pending_listing_count  \
0                 20.0                   50.0
1                  0.0                    2.0

   median_listing_price_per_square_foot  median_square_feet  …  \
0                                 235.0              2708.0  …
1                                 113.0              3564.0  …

   pending_ratio  quality_flag  year  CPIAUCSL_value  FEDFUNDS_value  \
0         0.9434           0.0  2023         307.481            5.33
1         1.0000           1.0  2023         307.481            5.33

   UMCSENT_value  RSXFS_value  BOPGSTB_value  HOUST_value  PI_value
0           67.9     613076.0       -61542.0       1358.0   23166.1
1           67.9     613076.0       -61542.0       1358.0   23166.1

[2 rows x 22 columns]
```

[ ]:
```
df['target_bins'] = pd.qcut(df['median_days_on_market'], q=10,␣
 ↪duplicates='drop')

train_set, temp_set = train_test_split(df, test_size=0.4,␣
 ↪stratify=df['target_bins'], random_state=42)

validation_set, test_set = train_test_split(temp_set, test_size=0.5,␣
 ↪random_state=42)

train_set = train_set.drop(columns=['target_bins','month_date_yyyymm','year'])
validation_set = validation_set.
 ↪drop(columns=['target_bins','month_date_yyyymm','year'])
test_set = test_set.drop(columns=['target_bins','month_date_yyyymm','year'])

logging.info(f"Created Train Validate and Test sets")
```

[ ]:
```
train_set_x = train_set.drop('median_days_on_market', axis=1)
train_set_y = train_set['median_days_on_market']


val_set_x = validation_set.drop('median_days_on_market', axis=1)
val_set_y = validation_set['median_days_on_market']

test_set_x = test_set.drop('median_days_on_market', axis=1)
test_set_y = test_set['median_days_on_market']
```

[ ]:
```
# Training
rf_model = RandomForestRegressor(n_estimators=100, random_state=random.
 ↪randint(1,100))

rf_model.fit(train_set_x, train_set_y)
```

[ ]:
```
RandomForestRegressor(random_state=72)
```

[ ]:
```
# param_grid = {
#     'n_estimators': [50, 75, 125],
#     'max_depth': [5, 8, 12, 15],
#     'min_samples_split': [2, 5, 10],
#     'min_samples_leaf': [1, 2, 3]
# }

# rf = RandomForestRegressor(random_state=random.randint(1,100))
# grid_search = GridSearchCV(estimator=rf, param_grid=param_grid,
#                            cv=3, n_jobs=-1, verbose=2,␣
 ↪scoring='neg_mean_squared_error')


# grid_search.fit(train_set_x, train_set_y)


# print("Best Parameters:", grid_search.best_params_)

# best_model = grid_search.best_estimator_

# val_predictions = best_model.predict(train_set_x)

# val_mse = mean_squared_error(train_set_y, val_predictions)
# val_r2 = r2_score(train_set_y, val_predictions)

# print("Validation MSE with Best Model:", val_mse)
# print("Validation R-squared with Best Model:", val_r2)
```