

swiggy-regression

April 19, 2023

```
[1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[2]: from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.tree import DecisionTreeRegressor, plot_tree
from sklearn.ensemble import RandomForestRegressor
from sklearn.compose import ColumnTransformer
from sklearn.metrics import r2_score, mean_squared_error
```

```
[3]: df = pd.read_csv('swiggy-preprocessed.csv')
```

```
[4]: df.head()
```

```
[4]:
```

	id	name	city	rating	rating_count	cost	\
0	567335	AB FOODS POINT	Abohar	0.0	Too Few Ratings	200.0	
1	531342	Janta Sweet House	Abohar	4.4	50+ ratings	200.0	
2	158203	theka coffee desi	Abohar	3.8	100+ ratings	100.0	
3	187912	Singh Hut	Abohar	3.7	20+ ratings	250.0	
4	543530	GRILL MASTERS	Abohar	0.0	Too Few Ratings	250.0	

	lic_no	link	\
0	22122652000138	https://www.swiggy.com/restaurants/ab-foods-po...	
1	12117201000112	https://www.swiggy.com/restaurants/janta-sweet...	
2	22121652000190	https://www.swiggy.com/restaurants/theka-coffe...	
3	22119652000167	https://www.swiggy.com/restaurants/singh-hut-n...	
4	12122201000053	https://www.swiggy.com/restaurants/grill-maste...	

	address	menu	\
0	AB FOODS POINT, NEAR RISHI NARANG DENTAL CLINI...	Menu/567335.json	
1	Janta Sweet House, Bazar No.9, Circullar Road,...	Menu/531342.json	
2	theka coffee desi, sahtiya sadan road city	Menu/158203.json	
3	Singh Hut, CIRCULAR ROAD NEAR NEHRU PARK ABOHAR	Menu/187912.json	
4	GRILL MASTERS, ADA Heights, Abohar - Hanumanga...	Menu/543530.json	

	sub_area	area	cuisine1	cuisine2
0	Abohar	Abohar	Beverages	Pizzas
1	Abohar	Abohar	Sweets	Bakery
2	Abohar	Abohar	Beverages	Beverages
3	Abohar	Abohar	Fast Food	Indian
4	Abohar	Abohar	Italian-American	Fast Food

```
[5]: X = df.drop('cost', axis=1)
y = df['cost']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
↳random_state=42)
```

```
[6]: cat_cols = ['rating_count', 'sub_area', 'area', 'cuisine1', 'cuisine2', 'city']
cat_transformer = Pipeline(steps=[
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])
```

```
[7]: preprocessor = ColumnTransformer(transformers=[
    ('cat', cat_transformer, cat_cols)
])
```

```
[8]: dt_pipe = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('scaler', StandardScaler(with_mean=False)),
    ('regressor', DecisionTreeRegressor())
])

rf_pipe = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('scaler', StandardScaler(with_mean=False)),
    ('regressor', RandomForestRegressor())
])
```

```
[9]: dt_params = {
    'regressor__max_depth': [3, 5, 7],
    'regressor__min_samples_split': [2, 3, 4]
}

rf_params = {
    'regressor__n_estimators': [100, 200],
    'regressor__max_depth': [5, 7],
    'regressor__min_samples_split': [2, 3],
    'regressor__max_features': ['sqrt', 'log2']
}
```

```
[10]: dt_grid = GridSearchCV(dt_pipe, dt_params, cv=5)
      rf_grid = GridSearchCV(rf_pipe, rf_params, cv=5)
```

```
[11]: dt_grid.fit(X_train, y_train)
      rf_grid.fit(X_train, y_train)
```

```
[11]: GridSearchCV(cv=5,
                  estimator=Pipeline(steps=[('preprocessor',
                                             ColumnTransformer(transformers=[('cat',
Pipeline(steps=[('onehot',
                  OneHotEncoder(handle_unknown='ignore'))])),
['rating_count',
'sub_area',
'area',
'cuisine1',
'cuisine2',
'city']))]),
                  ('scaler',
                   StandardScaler(with_mean=False)),
                  ('regressor', RandomForestRegressor()))],
                  param_grid={'regressor__max_depth': [5, 7],
                              'regressor__max_features': ['sqrt', 'log2'],
                              'regressor__min_samples_split': [2, 3],
                              'regressor__n_estimators': [100, 200]})
```

```
[12]: dt_pred = dt_grid.predict(X_test)
      rf_pred = rf_grid.predict(X_test)
```

```
[13]: dt_r2 = r2_score(y_test, dt_pred)
      dt_rmse = mean_squared_error(y_test, dt_pred, squared=False)
      rf_r2 = r2_score(y_test, rf_pred)
      rf_rmse = mean_squared_error(y_test, rf_pred, squared=False)

      print(f'Decision Tree R2 score: {dt_r2:.4f}')
      print(f'Decision Tree RMSE: {dt_rmse:.2f}')
      print(f'Random Forest R2 score: {rf_r2:.4f}')
      print(f'Random Forest RMSE: {rf_rmse:.2f}')
```

```
Decision Tree R2 score: 0.0466
Decision Tree RMSE: 149.59
Random Forest R2 score: 0.0369
Random Forest RMSE: 150.35
```