Final_Project

April 25, 2022

[ ]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from numpy.linalg import inv
%matplotlib inline
from sklearn.model_selection import train_test_split, cross_validate,␣
↪cross_val_score
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.preprocessing import OrdinalEncoder
from sklearn.metrics import accuracy_score, r2_score, mean_squared_error,␣
↪mean_absolute_error
```

[ ]:
```
import warnings

def fxn():
    warnings.warn("deprecated", DeprecationWarning)

with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    fxn()
```

[ ]:
```
from google.colab import files
uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving real_estate_data.csv to real_estate_data.csv
```

1 EDA and Cleaning

[ ]:
```
data = pd.read_csv('real_estate_data.csv')
data.head()
```

[ ]:

	tx_price	beds	baths	sqft	year_built	lot_size	\
0	295850	1	1	584	2013	0	
1	216500	1	1	612	1965	0	
2	279900	1	1	615	1963	0	

```
3    379900     1      1   618        2000     33541
4    340000     1      1   634        1992         0

                   property_type exterior_walls                roof  \
0  Apartment / Condo / Townhouse    Wood Siding                 NaN
1  Apartment / Condo / Townhouse          Brick  Composition Shingle
2  Apartment / Condo / Townhouse    Wood Siding                 NaN
3  Apartment / Condo / Townhouse    Wood Siding                 NaN
4  Apartment / Condo / Townhouse          Brick                 NaN

   basement  …  beauty_spas  active_life  median_age  married  college_grad  \
0       NaN  …           47           58        33.0     65.0          84.0
1       1.0  …           26           14        39.0     73.0          69.0
2       NaN  …           74           62        28.0     15.0          86.0
3       NaN  …           72           83        36.0     25.0          91.0
4       NaN  …           50           73        37.0     20.0          75.0

   property_tax  insurance  median_school  num_schools  tx_year
0         234.0       81.0            9.0          3.0     2013
1         169.0       51.0            3.0          3.0     2006
2         216.0       74.0            8.0          3.0     2012
3         265.0       92.0            9.0          3.0     2005
4          88.0       30.0            9.0          3.0     2002

[5 rows x 26 columns]
```

[ ]:
```
data.isnull().sum()
```

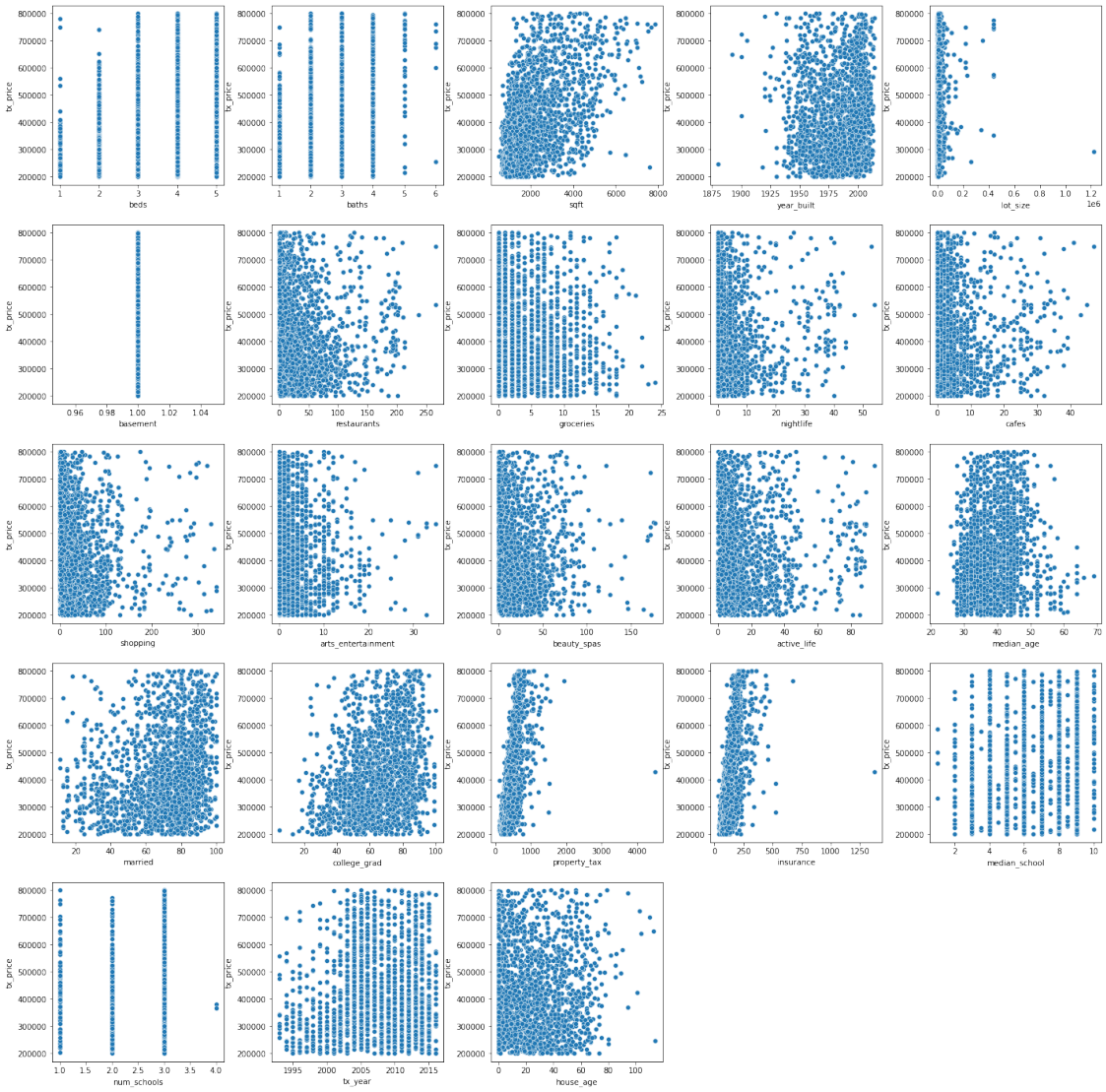
[ ]:
```
tx_price               0
beds                   0
baths                  0
sqft                   0
year_built             0
lot_size               0
property_type          0
exterior_walls       223
roof                 354
basement             226
restaurants            0
groceries              0
nightlife              0
cafes                  0
shopping               0
arts_entertainment     0
beauty_spas            0
active_life            0
median_age             0
```

```
married                 0
college_grad            0
property_tax            0
insurance               0
median_school           0
num_schools             0
tx_year                 0
dtype: int64
```

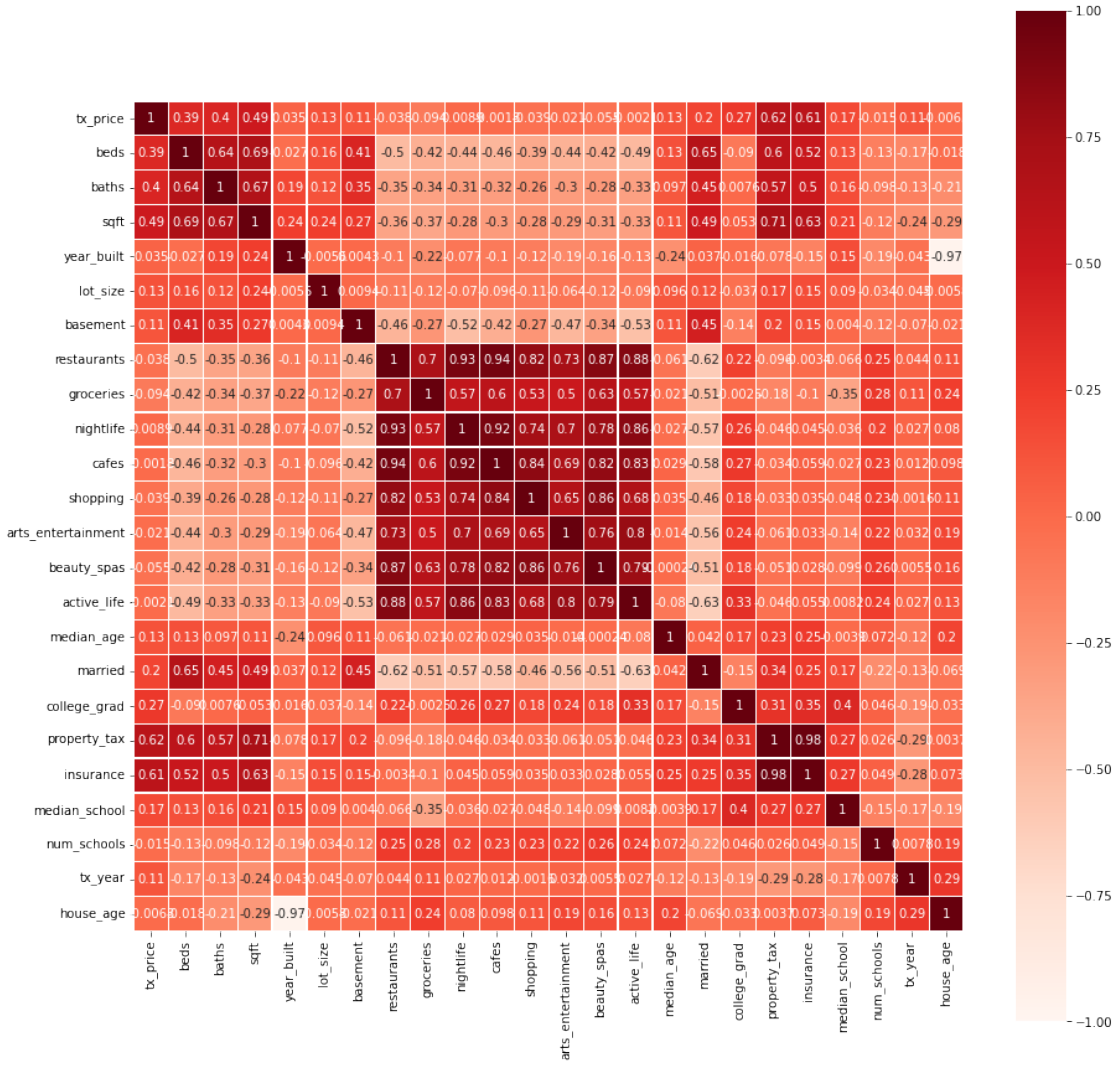
[ ]:
```
data['house_age'] = data['tx_year'] - data['year_built']
data = data[data['house_age'] >= 0]
```

[ ]:
```
features = data.columns.values.tolist()
features.remove('tx_price')

cat_features = ['property_type','exterior_walls','roof']
num_features = [val for val in features if val not in cat_features]
```

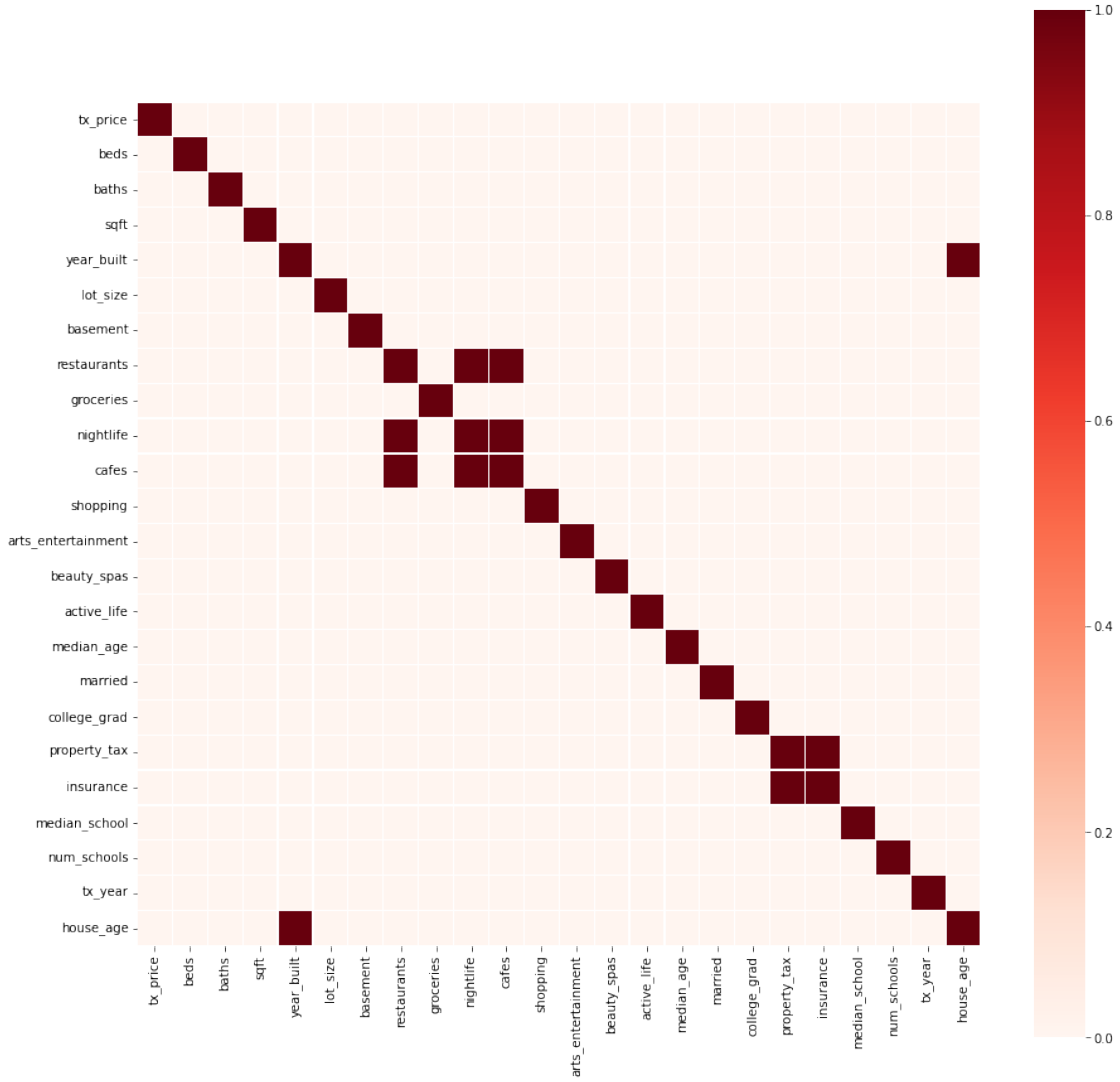
[ ]:
```
num_df = data[num_features]
target = data['tx_price']
fig, ax = plt.subplots(5, 5, figsize=(20,20))
fig.tight_layout(pad=3)
for var, subplot in zip(num_features, ax.flatten()): sns.scatterplot(x=var,␣
 ↪y=target, data=num_df, ax=subplot)
fig.delaxes(ax[4][3])
fig.delaxes(ax[4][4])
```



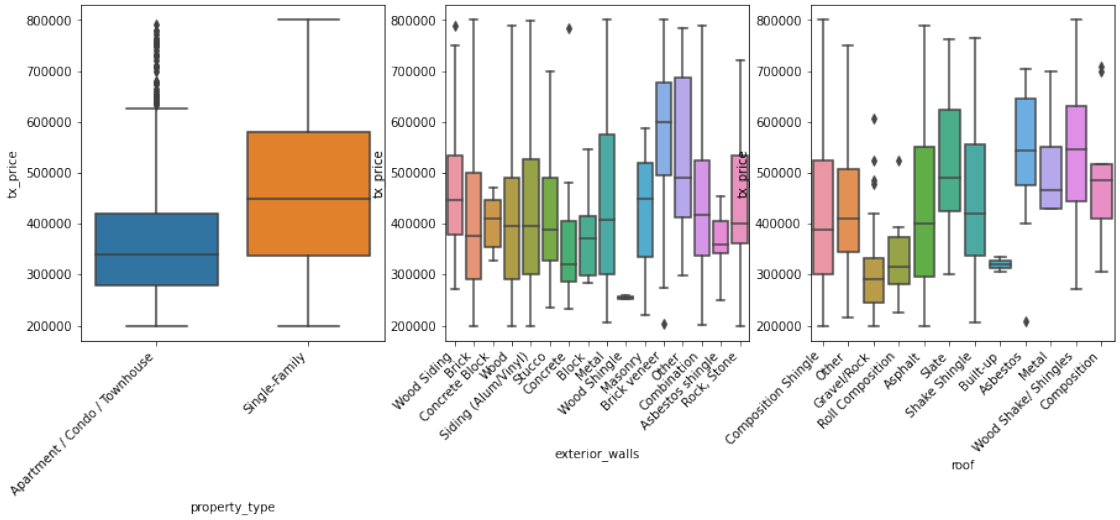
[ ]:
```
plt.figure(figsize=(15,15));
sns.heatmap(data=data.corr(),vmin=-1,vmax=1,linewidths=.3,annot=True, cmap=plt.
↪cm.Reds,square=True);
```



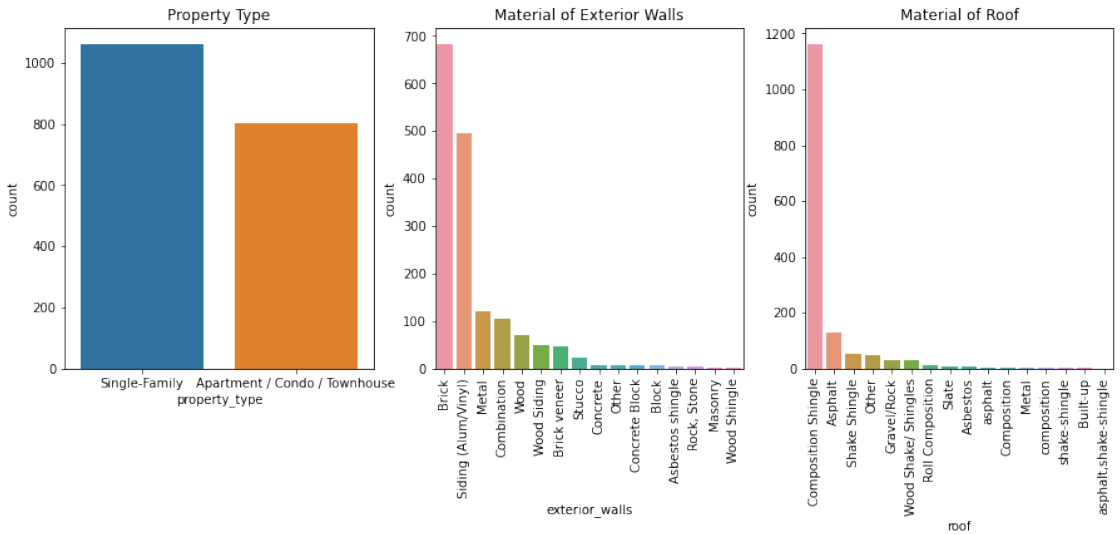
[ ]:
```
plt.figure(figsize=(15,15));
sns.heatmap(data=abs(data.corr())>.9,vmin=0,vmax=1,linewidths=.3, cmap=plt.cm.
↪Reds,square=True);
```



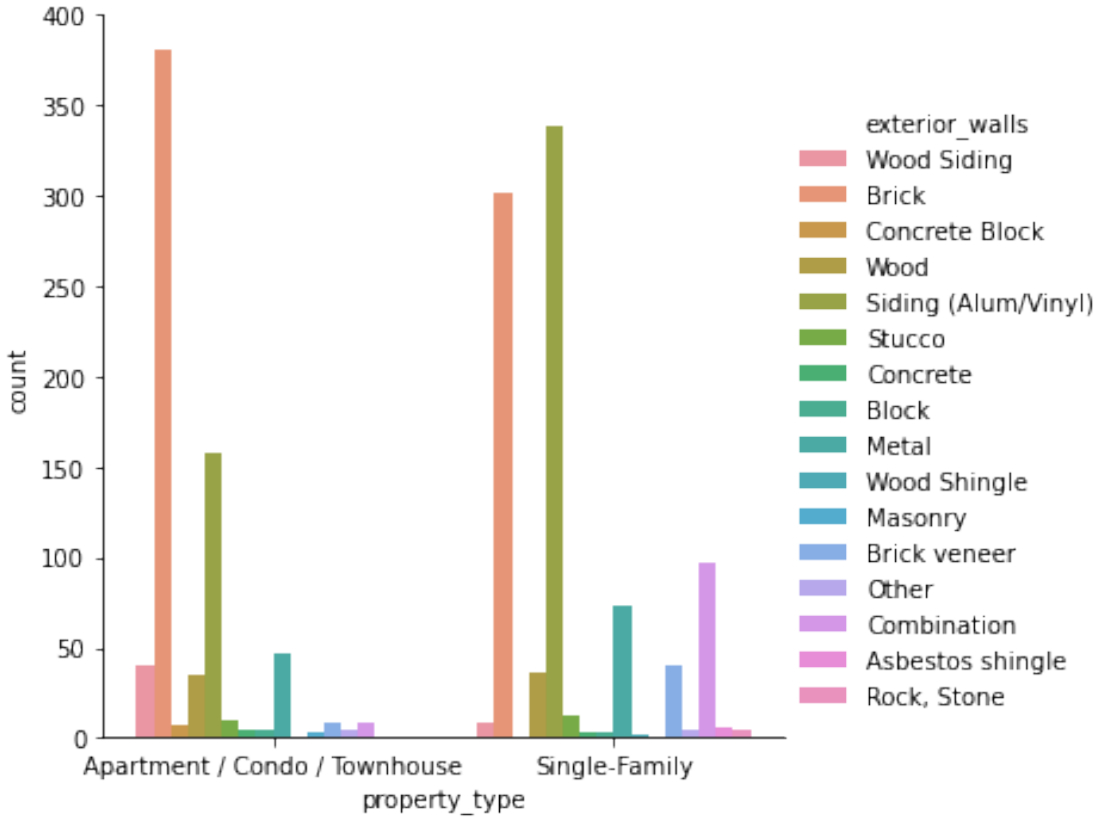
[ ]:
```
cat_df = data[cat_features]
target = data['tx_price']
fig, ax = plt.subplots(1, 3, figsize=(15,5))
for var, subplot in zip(cat_features, ax.flatten()):
    box = sns.boxplot(x=var, y=target, data=cat_df, ax=subplot)
    box.set_xticklabels(box.get_xticklabels(), rotation=45,␣
 ↪horizontalalignment='right')
```



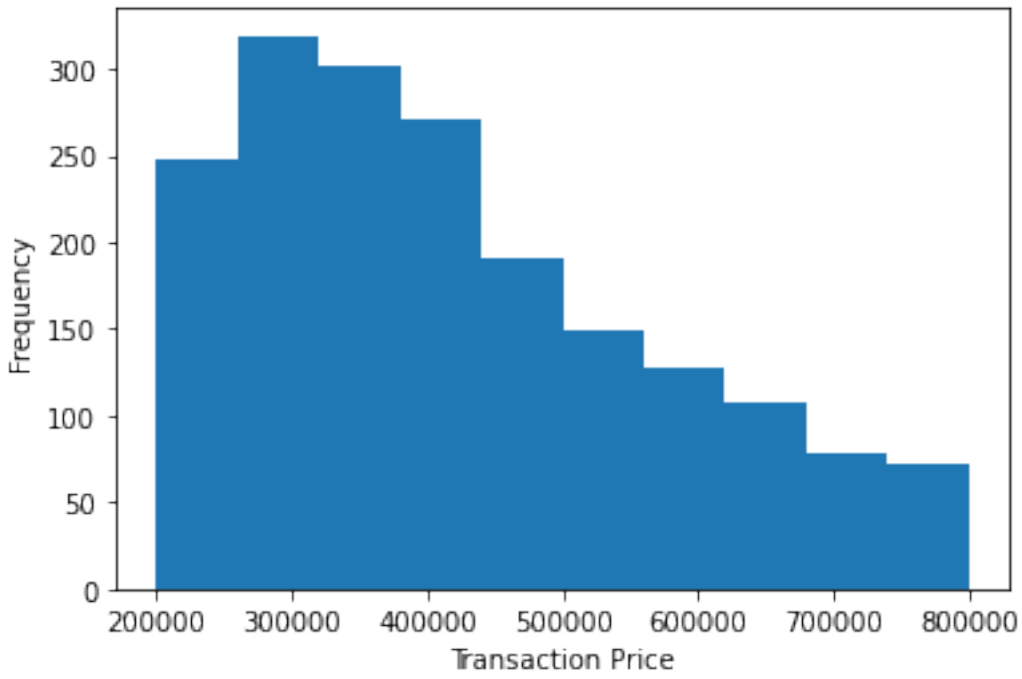
[ ]:
```
f, ax = plt.subplots(1, 3, figsize=(15,5))
sns.countplot(x='property_type', data=data, ax=ax[0],␣
 ↪order=data['property_type'].value_counts().index)
sns.countplot(x='exterior_walls', data=data, ax=ax[1],␣
 ↪order=data['exterior_walls'].value_counts().index)
sns.countplot(x='roof', data=data, ax=ax[2], order=data['roof'].value_counts().
 ↪index)
ax[0].set_title('Property Type')
ax[1].set_title('Material of Exterior Walls')
ax[1].tick_params(axis='x',rotation=90)
ax[2].set_title('Material of Roof')
ax[2].tick_params(axis='x',rotation=90)
```



[ ]:
```
ax = sns.catplot(x='property_type', hue='exterior_walls', data=data,␣
↪kind='count')
```



[ ]:
```
plt.hist(data.tx_price, bins=10)
plt.xlabel('Transaction Price')
plt.ylabel('Frequency')
plt.show()
```



[ ]:
```
data.basement = data.basement.fillna(0)

data.roof.replace('composition', 'Composition', inplace=True)
data.roof.replace('asphalt', 'Asphalt', inplace=True)
data.roof.replace(['asphalt,shake-shingle', 'shake-shingle'], 'Shake Shingle',␣
↪inplace=True)
data.roof = data.roof.fillna(data.roof.mode()[0])

sub_data = data['property_type'] == 'Single-Family'
data.loc[sub_data, 'exterior_walls'] = data.loc[sub_data, 'exterior_walls'].
↪fillna('Siding (Alum/Vinyl)')

sub_data = data['property_type'] == 'Apartment / Condo / Townhouse'
data.loc[sub_data, 'exterior_walls'] = data.loc[sub_data, 'exterior_walls'].
↪fillna('Brick')

data = data[(data['insurance'] < 1200) & (data['property_tax'] < 4000)]
```

2 Feature Engineering

[ ]:
```
correlation_matrix = data.corr().abs()
```

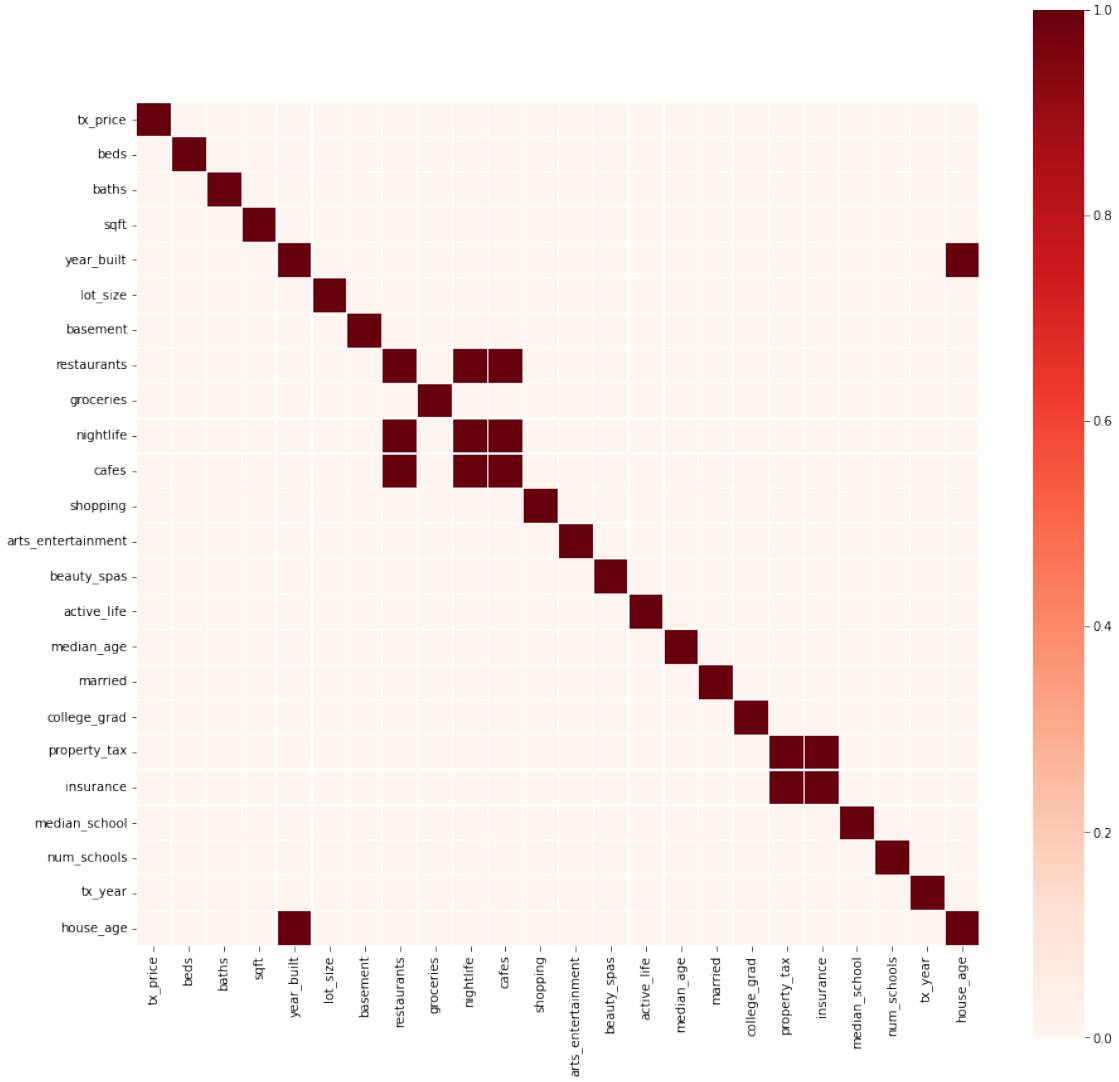
```
upper_triangle = correlation_matrix.where(np.triu(np.ones(correlation_matrix.
 ↪shape),k=1).astype(bool))
drop_columns = [column for column in upper_triangle.columns if␣
 ↪any(upper_triangle[column] >= 0.9)]
print("Highly Correlated Features which could be removed: " ,drop_columns)

drop_columns = ['nightlife', 'cafes', 'insurance', 'year_built']
data = data.drop(drop_columns, axis=1)
```

```
Highly Correlated Features which could be removed:  ['nightlife', 'cafes',
'insurance', 'house_age']
```

[ ]:
```
num_cols = data.columns[data.dtypes.apply(lambda c: np.issubdtype(c, np.
 ↪number))]
num_cols = num_cols.to_list()
num_cols.remove('tx_price')
```

[ ]:
```
enc = OrdinalEncoder(categories=[["Apartment / Condo / Townhouse",␣
 ↪"Single-Family"]])
data["property_type_enc"] = enc.fit_transform(data['property_type'].to_numpy().
 ↪reshape(-1,1))
data = data.drop(columns=['property_type'])

data = pd.get_dummies(data)
```

[ ]:
```
plt.figure(figsize=(15,15));
sns.heatmap(data=abs(data.corr())>.9,vmin=0,vmax=1,linewidths=.3,cmap=plt.cm.
 ↪Reds,square=True);
```



[ ]:
```
X = data.drop(columns=['tx_price'])
y = np.log(data['tx_price'])

feature_names = X.columns.values.tolist()

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.
↪2,random_state=42)

ss = StandardScaler()
X_train[num_cols] = ss.fit_transform(X_train[num_cols])
X_test[num_cols] = ss.transform(X_test[num_cols])
```

3 Models

Baseline Model

[ ]:
```
y_pred = np.ones(y_train.shape[0]) * y_train.mean()

print("\n========Evaluation on Train Set========\n")
print("R2 score on train set : ",r2_score(y_train, y_pred))
print("MSE on train set : ", mean_squared_error(y_train, y_pred))
print("MAE on train set : ", mean_absolute_error(y_train, y_pred))

y_pred = np.ones(y_test.shape[0]) * y_train.mean()

print("\n========Evaluation on Test Set========\n")
print("R2 score on test set : ",r2_score(y_test, y_pred))
print("MSE on test set : ", mean_squared_error(y_test, y_pred))
print("MAE on test set : ", mean_absolute_error(y_test, y_pred))
```

```
========Evaluation on Train Set========

R2 score on train set :  0.0
MSE on train set :  0.1256931780081801
MAE on train set :  0.2977844237150744

========Evaluation on Test Set========

R2 score on test set :  -0.006155613336636057
MSE on test set :  0.1277160240298454
MAE on test set :  0.29584966873610363
```

Regression Model (Lasso, Ridge, Elastic Net) (Pavan)

[ ]:
```
#Adding Bias
ones_train = np.ones((X_train.shape[0],1))
ones_test = np.ones((X_test.shape[0],1))
X_train_reg = np.hstack((X_train,ones_train))
X_test_reg = np.hstack((X_test,ones_test))
```

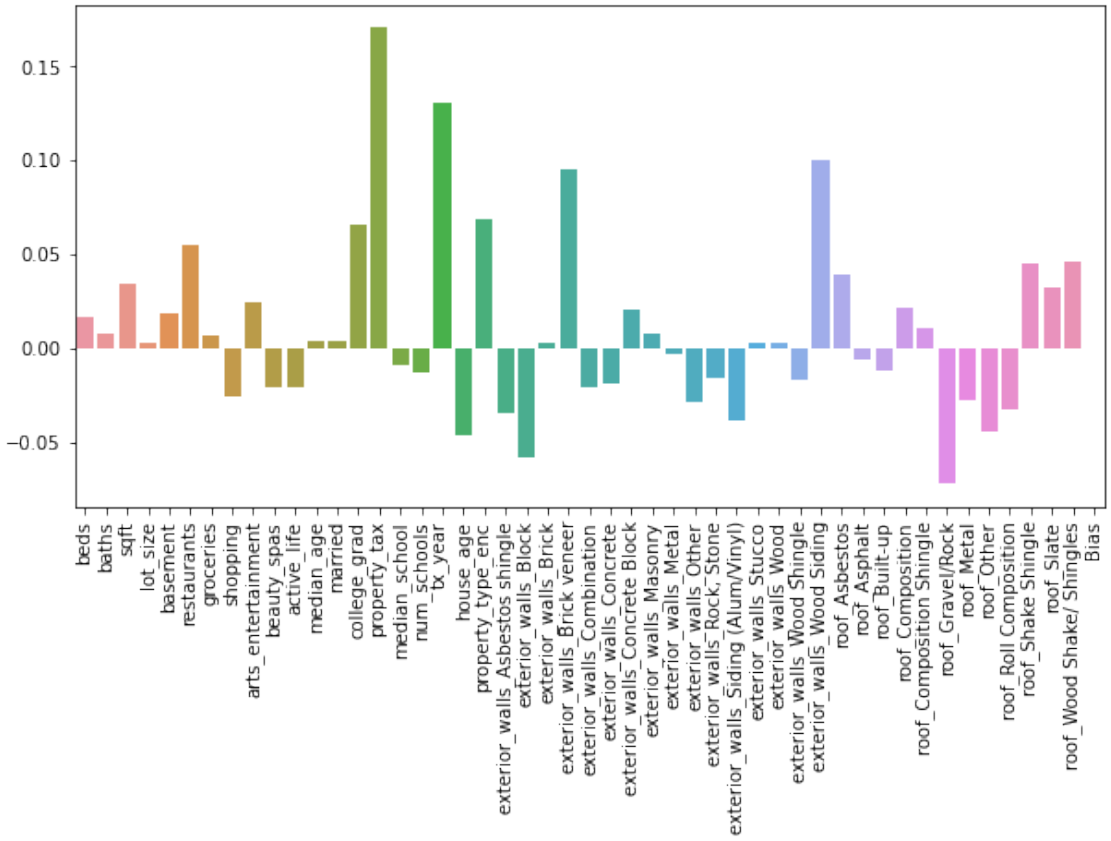
[ ]:
```
from sklearn.linear_model import Ridge

alphas = [0.001, 0.01, 0.1, 1, 10, 100]

print("\n========Model Training on Different Hyperparameters========\n")

cross_val_scores = []
hyper_params = []

for alpha in alphas:
```

```
    model = Ridge(alpha=alpha)
    scores = cross_val_score(model, X_train_reg, y_train, cv=5,␣
 ↪error_score="raise")
    cross_val_scores.append(np.mean(scores))
    print("alpha:",alpha, " score:",np.mean(scores))
    hyper_params.append(alpha)

best_alpha = hyper_params[np.argmax(cross_val_scores)]

print("\n========Best Hyper Parameters========\n")
print("best_alpha:",best_alpha)

model = Ridge(alpha=best_alpha)
model.fit(X_train_reg, y_train)


y_pred_train = model.predict(X_train_reg)

print("\n========Evaluation on Train Set========\n")
print("R2 score on train set : ",r2_score(y_train, y_pred_train))
print("MSE on train set : ", mean_squared_error(y_train, y_pred_train))
print("MAE on train set : ", mean_absolute_error(y_train, y_pred_train))

y_pred_test = model.predict(X_test_reg)

print("\n========Evaluation on Test Set========\n")
print("R2 score on test set : ",r2_score(y_test, y_pred_test))
print("MSE on test set : ", mean_squared_error(y_test, y_pred_test))
print("MAE on test set : ", mean_absolute_error(y_test, y_pred_test))
```

```
========Model Training on Different Hyperparameters========

alpha: 0.001  score: 0.4737188593005611
alpha: 0.01  score: 0.473782333223439
alpha: 0.1  score: 0.47437735506829065
alpha: 1  score: 0.47812268975396177
alpha: 10  score: 0.48552100481509647
alpha: 100  score: 0.48542113374847207

========Best Hyper Parameters========

best_alpha: 10

========Evaluation on Train Set========

R2 score on train set :  0.518496272388511
MSE on train set :  0.06052173374627315
```

```
MAE on train set :  0.20000892100510467

========Evaluation on Test Set========

R2 score on test set :  0.4300177969270602
MSE on test set :  0.07235049904739933
MAE on test set :  0.21241927286460208
```

[ ]:
```
features_reg = feature_names+['Bias']
fig = plt.figure(figsize=(10, 5))
ax = sns.barplot(x=features_reg,y=model.coef_)
ax.tick_params(axis='x',rotation=90)
```



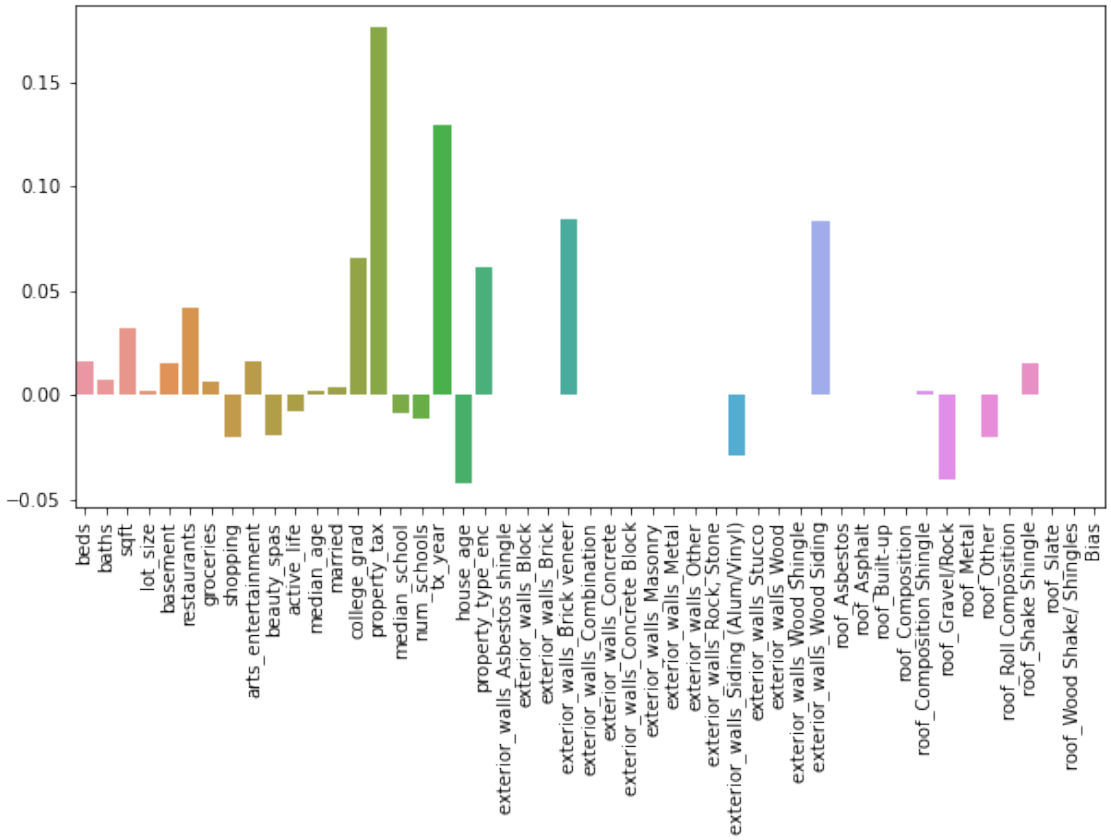
[ ]:
```
from sklearn.linear_model import Lasso

alphas = [0.001, 0.01, 0.1, 1, 10, 100]

print("\n========Model Training on Different Hyperparameters========\n")

cross_val_scores = []
```

```
hyper_params = []

for alpha in alphas:
    model = Lasso(alpha=alpha)
    scores = cross_val_score(model, X_train_reg, y_train, cv=5,␣
 ↪error_score="raise")
    cross_val_scores.append(np.mean(scores))
    print("alpha:",alpha, " score:",np.mean(scores))
    hyper_params.append(alpha)

best_alpha = hyper_params[np.argmax(cross_val_scores)]

print("\n========Best Hyper Parameters========\n")
print("best_alpha:",best_alpha)

model = Lasso(alpha=best_alpha)
model.fit(X_train_reg, y_train)

y_pred_train = model.predict(X_train_reg)

print("\n========Evaluation on Train Set========\n")
print("R2 score on train set : ",r2_score(y_train, y_pred_train))
print("MSE on train set : ", mean_squared_error(y_train, y_pred_train))
print("MAE on train set : ", mean_absolute_error(y_train, y_pred_train))

y_pred_test = model.predict(X_test_reg)

print("\n========Evaluation on Test Set========\n")
print("R2 score on test set : ",r2_score(y_test, y_pred_test))
print("MSE on test set : ", mean_squared_error(y_test, y_pred_test))
print("MAE on test set : ", mean_absolute_error(y_test, y_pred_test))
```

```
========Model Training on Different Hyperparameters========

alpha: 0.001  score: 0.48718597719390966
alpha: 0.01  score: 0.47774179879623124
alpha: 0.1  score: 0.2814091186575013
alpha: 1  score: -0.0013724153754979707
alpha: 10  score: -0.0013724153754979707
alpha: 100  score: -0.0013724153754979707

========Best Hyper Parameters========

best_alpha: 0.001

========Evaluation on Train Set========
```

```
R2 score on train set :  0.5140713216417605
MSE on train set :  0.061077919868161884
MAE on train set :  0.2011548240607999

========Evaluation on Test Set========

R2 score on test set :  0.4359036043762211
MSE on test set :  0.07160338606045369
MAE on test set :  0.21108647350809953
```

[ ]:
```
features_reg = feature_names+['Bias']
fig = plt.figure(figsize=(10, 5))
ax = sns.barplot(x=features_reg,y=model.coef_)
ax.tick_params(axis='x',rotation=90)
```



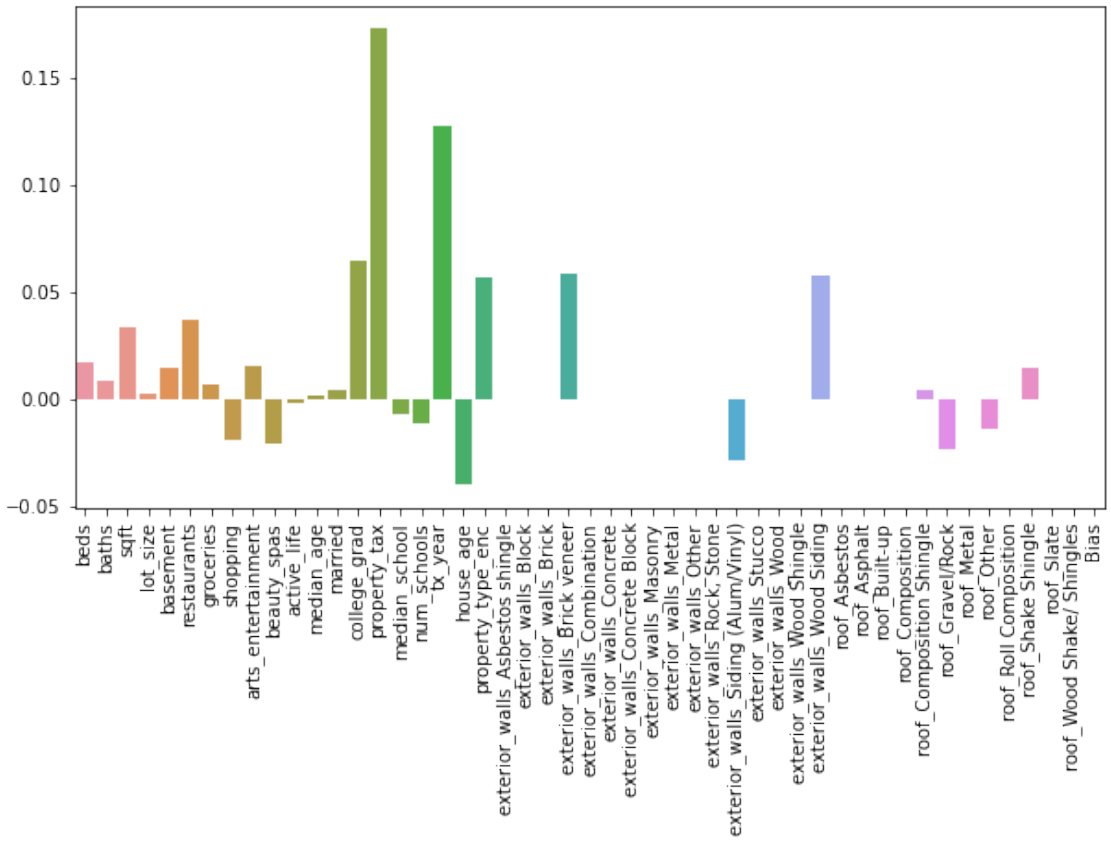
[ ]:
```
from sklearn.linear_model import ElasticNet

alphas = [0.001, 0.01, 0.1, 1, 10, 100]
l1_ratios = [0.1, 0.3, 0.5, 0.7, 0.9]
```

```
print("\n========Model Training on Different Hyperparameters========\n")

cross_val_scores = []
hyper_params = []

for alpha in alphas:
    for l1_ratio in l1_ratios:
        model = ElasticNet(alpha=alpha, l1_ratio=l1_ratio)
        scores = cross_val_score(model, X_train_reg, y_train, cv=5,␣
 ↪error_score="raise")
        cross_val_scores.append(np.mean(scores))
        print("alpha:",alpha, " score:",np.mean(scores))
        hyper_params.append([alpha,l1_ratio])

best_alpha, best_l1_ratio = hyper_params[np.argmax(cross_val_scores)]

print("\n========Best Hyper Parameters========\n")
print("best_alpha:",best_alpha, " best_l1_ratio:",best_l1_ratio)

model = ElasticNet(alpha=best_alpha, l1_ratio=best_l1_ratio)
model.fit(X_train_reg, y_train)

y_pred_train = model.predict(X_train_reg)

print("\n========Evaluation on Train Set========\n")
print("R2 score on train set : ",r2_score(y_train, y_pred_train))
print("MSE on train set : ", mean_squared_error(y_train, y_pred_train))
print("MAE on train set : ", mean_absolute_error(y_train, y_pred_train))

y_pred_test = model.predict(X_test_reg)

print("\n========Evaluation on Test Set========\n")
print("R2 score on test set : ",r2_score(y_test, y_pred_test))
print("MSE on test set : ", mean_squared_error(y_test, y_pred_test))
print("MAE on test set : ", mean_absolute_error(y_test, y_pred_test))
```

```
========Model Training on Different Hyperparameters========

alpha: 0.001  score: 0.4811815749223646
alpha: 0.001  score: 0.48447530488151946
alpha: 0.001  score: 0.4861736089956712
alpha: 0.001  score: 0.48659002903775866
alpha: 0.001  score: 0.4870362905250677
alpha: 0.01  score: 0.487645316281048
alpha: 0.01  score: 0.48432142854056937
alpha: 0.01  score: 0.48112782053881303
alpha: 0.01  score: 0.47943220068003767
```

```
alpha: 0.01  score: 0.47849619333357507
alpha: 0.1  score: 0.47157161500313477
alpha: 0.1  score: 0.44216648383884055
alpha: 0.1  score: 0.3924379844032929
alpha: 0.1  score: 0.3359750854822444
alpha: 0.1  score: 0.29448327696674326
alpha: 1  score: 0.19197844976326564
alpha: 1  score: -0.0013724153754979707
alpha: 1  score: -0.0013724153754979707
alpha: 1  score: -0.0013724153754979707
alpha: 1  score: -0.0013724153754979707
alpha: 10  score: -0.0013724153754979707
alpha: 10  score: -0.0013724153754979707
alpha: 10  score: -0.0013724153754979707
alpha: 10  score: -0.0013724153754979707
alpha: 10  score: -0.0013724153754979707
alpha: 100  score: -0.0013724153754979707
alpha: 100  score: -0.0013724153754979707
alpha: 100  score: -0.0013724153754979707
alpha: 100  score: -0.0013724153754979707
alpha: 100  score: -0.0013724153754979707

========Best Hyper Parameters========

best_alpha: 0.01  best_l1_ratio: 0.1

========Evaluation on Train Set========

R2 score on train set :  0.5122080259598225
MSE on train set :  0.061312123423993586
MAE on train set :  0.20165938325872482

========Evaluation on Test Set========

R2 score on test set :  0.4389138526567997
MSE on test set :  0.07122128120843857
MAE on test set :  0.21046977724444504
```

[ ]:
```
features_reg = feature_names+['Bias']
fig = plt.figure(figsize=(10, 5))
ax = sns.barplot(x=features_reg,y=model.coef_)
ax.tick_params(axis='x',rotation=90)
```



Random Forest & Gradient Boosting Models (Pavan)

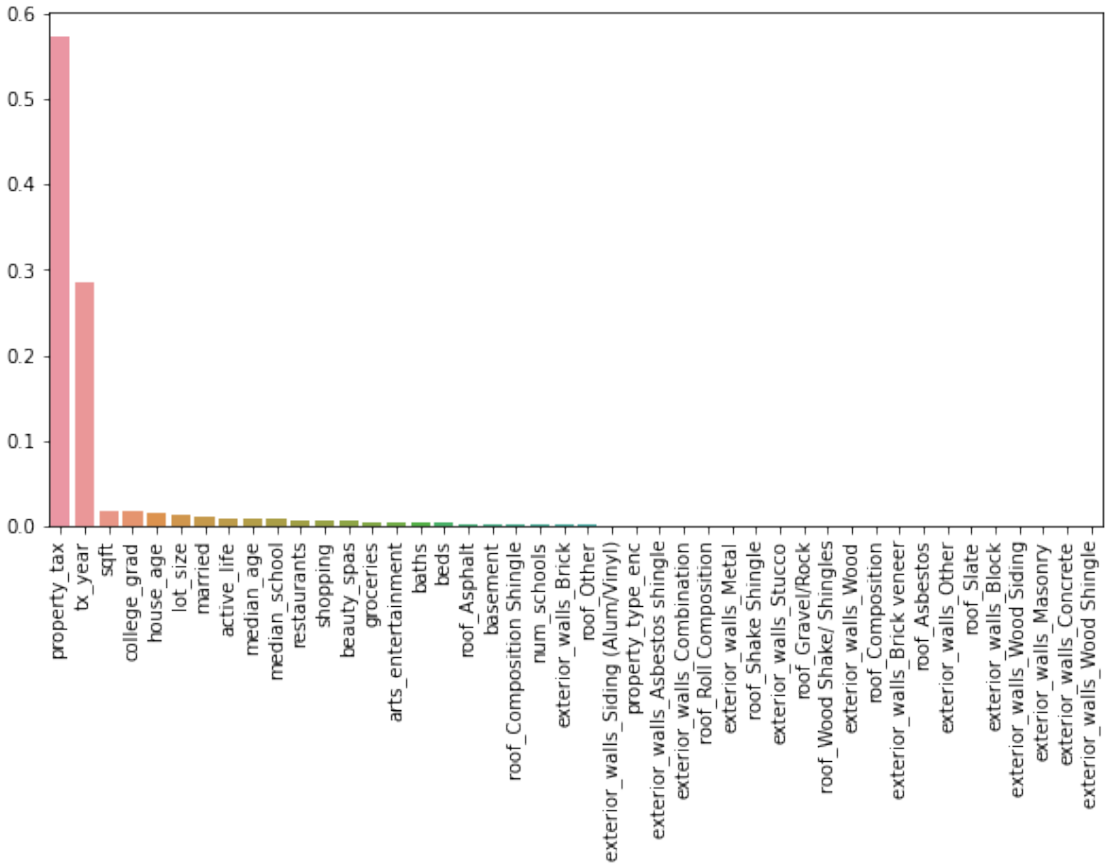
[ ]:
```
from sklearn.ensemble import RandomForestRegressor

n_estimators = [10, 25, 50]
max_features = [10, 20, 30, 'auto']
max_depths = [3, 6, 9]

print("\n========Model Training on Different Hyperparameters========\n")

cross_val_scores = []
hyper_params = []
for n_estimator in n_estimators:
    for max_depth in max_depths:
        for max_feature in max_features:
            model =␣
↪RandomForestRegressor(n_estimators=n_estimator,max_depth=max_depth,max_features=max_feature
            scores = cross_val_score(model, X_train, y_train, cv=5,␣
↪error_score="raise")
            cross_val_scores.append(np.mean(scores))
```

```
            print("n_estimator:",n_estimator," max_depth:",max_depth, "␣
 ↪max_features:",max_feature, " score:",np.mean(scores))
            hyper_params.append([n_estimator,max_depth,max_feature])

best_n_estimator, best_max_depth, best_max_feature = hyper_params[np.
 ↪argmax(cross_val_scores)]

print("\n========Best Hyper Parameters========\n")
print("best_n_estimator:",best_n_estimator," best_max_depth:",best_max_depth,"␣
 ↪best_max_features:",best_max_feature)

model = RandomForestRegressor(n_estimators=best_n_estimator,␣
 ↪max_depth=best_max_depth, max_features=best_max_feature)
model.fit(X_train, y_train)

y_pred_train = model.predict(X_train)

print("\n========Evaluation on Train Set========\n")
print("R2 score on train set : ",r2_score(y_train, y_pred_train))
print("MSE on train set : ", mean_squared_error(y_train, y_pred_train))
print("MAE on train set : ", mean_absolute_error(y_train, y_pred_train))

y_pred_test = model.predict(X_test)

print("\n========Evaluation on Test Set========\n")
print("R2 score on test set : ",r2_score(y_test, y_pred_test))
print("MSE on test set : ", mean_squared_error(y_test, y_pred_test))
print("MAE on test set : ", mean_absolute_error(y_test, y_pred_test))
```

```
========Model Training on Different Hyperparameters========

n_estimator: 10  max_depth: 3  max_features: 10  score: 0.5040749920315644
n_estimator: 10  max_depth: 3  max_features: 20  score: 0.5972790485531583
n_estimator: 10  max_depth: 3  max_features: 30  score: 0.6596703416894666
n_estimator: 10  max_depth: 3  max_features: auto  score: 0.7067372951985467
n_estimator: 10  max_depth: 6  max_features: 10  score: 0.6506337045405308
n_estimator: 10  max_depth: 6  max_features: 20  score: 0.7443789147327153
n_estimator: 10  max_depth: 6  max_features: 30  score: 0.7596248632428406
n_estimator: 10  max_depth: 6  max_features: auto  score: 0.7814409763788692
n_estimator: 10  max_depth: 9  max_features: 10  score: 0.6873090404598746
n_estimator: 10  max_depth: 9  max_features: 20  score: 0.7519857174170221
n_estimator: 10  max_depth: 9  max_features: 30  score: 0.7727514712819026
n_estimator: 10  max_depth: 9  max_features: auto  score: 0.7796742516632801
n_estimator: 25  max_depth: 3  max_features: 10  score: 0.48024478529243525
n_estimator: 25  max_depth: 3  max_features: 20  score: 0.5862779255937758
n_estimator: 25  max_depth: 3  max_features: 30  score: 0.6680332431856
```

```
n_estimator: 25  max_depth: 3  max_features: auto  score: 0.7102769134609774
n_estimator: 25  max_depth: 6  max_features: 10  score: 0.661876916027996
n_estimator: 25  max_depth: 6  max_features: 20  score: 0.7414623728900194
n_estimator: 25  max_depth: 6  max_features: 30  score: 0.7762685172766849
n_estimator: 25  max_depth: 6  max_features: auto  score: 0.7853386403232232
n_estimator: 25  max_depth: 9  max_features: 10  score: 0.6971646782717673
n_estimator: 25  max_depth: 9  max_features: 20  score: 0.7742555105417581
n_estimator: 25  max_depth: 9  max_features: 30  score: 0.785104269763828
n_estimator: 25  max_depth: 9  max_features: auto  score: 0.7935709357183716
n_estimator: 50  max_depth: 3  max_features: 10  score: 0.4930112060709709
n_estimator: 50  max_depth: 3  max_features: 20  score: 0.603364734135825
n_estimator: 50  max_depth: 3  max_features: 30  score: 0.6713175362205861
n_estimator: 50  max_depth: 3  max_features: auto  score: 0.7105147045142874
n_estimator: 50  max_depth: 6  max_features: 10  score: 0.6516176497740106
n_estimator: 50  max_depth: 6  max_features: 20  score: 0.7472586910819514
n_estimator: 50  max_depth: 6  max_features: 30  score: 0.7761936451686953
n_estimator: 50  max_depth: 6  max_features: auto  score: 0.7875017548080151
n_estimator: 50  max_depth: 9  max_features: 10  score: 0.7197929607252675
n_estimator: 50  max_depth: 9  max_features: 20  score: 0.774436174714395
n_estimator: 50  max_depth: 9  max_features: 30  score: 0.7896104050363079
n_estimator: 50  max_depth: 9  max_features: auto  score: 0.7931372024639989

========Best Hyper Parameters========

best_n_estimator: 25  best_max_depth: 9  best_max_features: auto

========Evaluation on Train Set========

R2 score on train set :  0.9349233281394684
MSE on train set :  0.008179693700345714
MAE on train set :  0.0704355795895439

========Evaluation on Test Set========

R2 score on test set :  0.7560925738937901
MSE on test set :  0.03096030701487134
MAE on test set :  0.13085582193177087
```

[ ]:
```
feat_imps = zip(feature_names, model.feature_importances_)
feats, imps = zip(*(sorted(list(filter(lambda x:x[1] != 0, feat_imps)),␣
 ↪key=lambda x: x[1], reverse=True)))

plt.figure(figsize=(10, 5))
ax = sns.barplot(x=list(feats), y=list(imps))
ax.tick_params(axis='x', rotation=90)
plt.show()
```



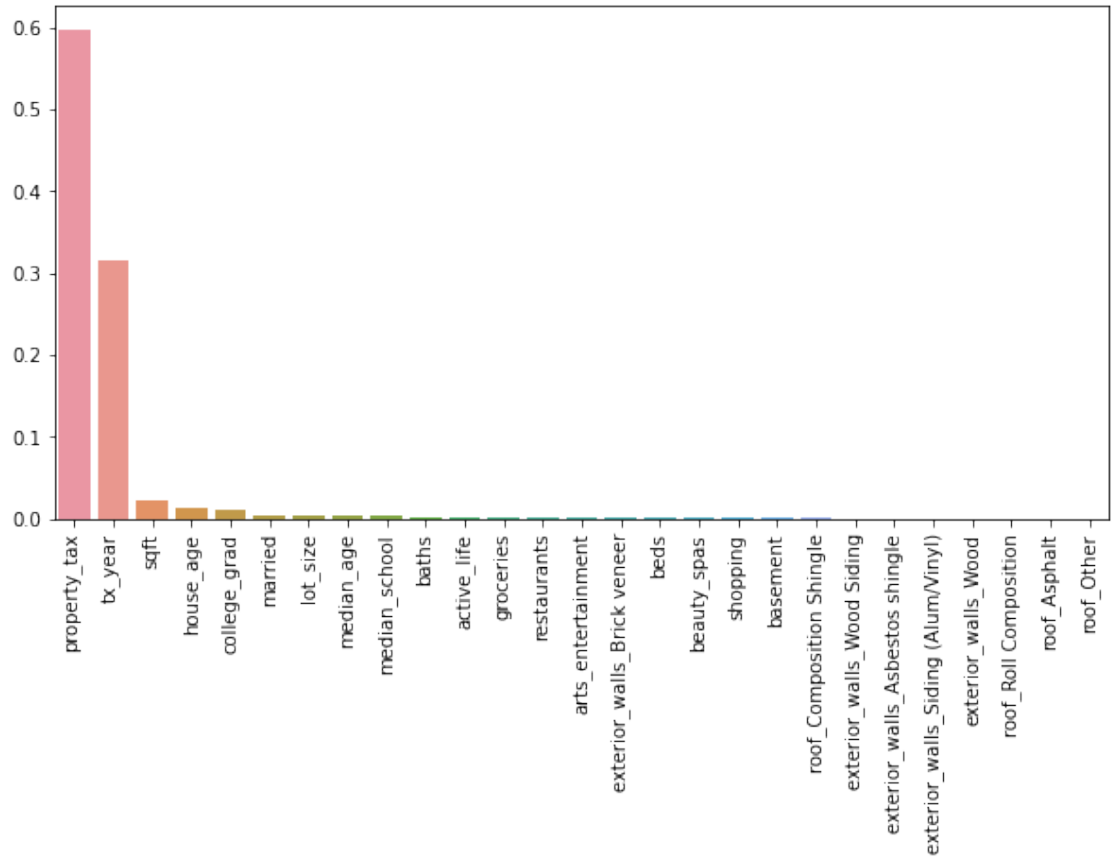
[ ]:
```
from sklearn.ensemble import GradientBoostingRegressor

n_estimators = [10, 25, 50]
learning_rates = [0.01, 0.1, 0.2]
max_depths = [3, 6, 9]

print("\n========Model Training on Different Hyperparameters========\n")

cross_val_scores = []
hyper_params = []
for n_estimator in n_estimators:
    for max_depth in max_depths:
        for learning_rate in learning_rates:
            model =␣
↪GradientBoostingRegressor(n_estimators=n_estimator,max_depth=max_depth,learning_rate=learni
            scores = cross_val_score(model, X_train, y_train, cv=5,␣
↪error_score="raise")
            cross_val_scores.append(np.mean(scores))
```

```
            print("n_estimator:",n_estimator," max_depth:",max_depth, "␣
 ↪learning_rate:",learning_rate, " score:",np.mean(scores))
            hyper_params.append([n_estimator,max_depth,learning_rate])

best_n_estimator, best_max_depth, best_learning_rate = hyper_params[np.
 ↪argmax(cross_val_scores)]

print("\n========Best Hyper Parameters========\n")
print("best_n_estimator:",best_n_estimator," best_max_depth:",best_max_depth,"␣
 ↪best_learning_rate:",best_learning_rate)

model = GradientBoostingRegressor(n_estimators=best_n_estimator,␣
 ↪max_depth=best_max_depth, learning_rate=best_learning_rate)
model.fit(X_train, y_train)



y_pred_train = model.predict(X_train)

print("\n========Evaluation on Train Set========\n")
print("R2 score on train set : ",r2_score(y_train, y_pred_train))
print("MSE on train set : ", mean_squared_error(y_train, y_pred_train))
print("MAE on train set : ", mean_absolute_error(y_train, y_pred_train))

y_pred_test = model.predict(X_test)

print("\n========Evaluation on Test Set========\n")
print("R2 score on test set : ",r2_score(y_test, y_pred_test))
print("MSE on test set : ", mean_squared_error(y_test, y_pred_test))
print("MAE on test set : ", mean_absolute_error(y_test, y_pred_test))
```

```
========Model Training on Different Hyperparameters========

n_estimator: 10  max_depth: 3  learning_rate: 0.01  score: 0.12341525295124271
n_estimator: 10  max_depth: 3  learning_rate: 0.1  score: 0.626509373413328
n_estimator: 10  max_depth: 3  learning_rate: 0.2  score: 0.7484318209648195
n_estimator: 10  max_depth: 6  learning_rate: 0.01  score: 0.14289105679371336
n_estimator: 10  max_depth: 6  learning_rate: 0.1  score: 0.6838730586571821
n_estimator: 10  max_depth: 6  learning_rate: 0.2  score: 0.7740573981268922
n_estimator: 10  max_depth: 9  learning_rate: 0.01  score: 0.14614065115312672
n_estimator: 10  max_depth: 9  learning_rate: 0.1  score: 0.6795213348963011
n_estimator: 10  max_depth: 9  learning_rate: 0.2  score: 0.7440706214181456
n_estimator: 25  max_depth: 3  learning_rate: 0.01  score: 0.2695835706049201
n_estimator: 25  max_depth: 3  learning_rate: 0.1  score: 0.7665374292288959
n_estimator: 25  max_depth: 3  learning_rate: 0.2  score: 0.7984587533629193
n_estimator: 25  max_depth: 6  learning_rate: 0.01  score: 0.3095250738817463
```

```
n_estimator: 25  max_depth: 6  learning_rate: 0.1  score: 0.7828759264602334
n_estimator: 25  max_depth: 6  learning_rate: 0.2  score: 0.7921709757590101
n_estimator: 25  max_depth: 9  learning_rate: 0.01  score: 0.3154718580424814
n_estimator: 25  max_depth: 9  learning_rate: 0.1  score: 0.7579973289151141
n_estimator: 25  max_depth: 9  learning_rate: 0.2  score: 0.7537876891324675
n_estimator: 50  max_depth: 3  learning_rate: 0.01  score: 0.43851886345892754
n_estimator: 50  max_depth: 3  learning_rate: 0.1  score: 0.7986578394823856
n_estimator: 50  max_depth: 3  learning_rate: 0.2  score: 0.8067405374360022
n_estimator: 50  max_depth: 6  learning_rate: 0.01  score: 0.49397474514314954
n_estimator: 50  max_depth: 6  learning_rate: 0.1  score: 0.7924144628517932
n_estimator: 50  max_depth: 6  learning_rate: 0.2  score: 0.7906841582124032
n_estimator: 50  max_depth: 9  learning_rate: 0.01  score: 0.4987844663459152
n_estimator: 50  max_depth: 9  learning_rate: 0.1  score: 0.7616436726998824
n_estimator: 50  max_depth: 9  learning_rate: 0.2  score: 0.7507979887908485

========Best Hyper Parameters========

best_n_estimator: 50  best_max_depth: 3  best_learning_rate: 0.2

========Evaluation on Train Set========

R2 score on train set :  0.8819881133228504
MSE on train set :  0.014833289079192145
MAE on train set :  0.09303013400918134

========Evaluation on Test Set========

R2 score on test set :  0.7691331249239745
MSE on test set :  0.029305009060303148
MAE on test set :  0.12493915657010664
```

[ ]:
```
feat_imps = zip(feature_names, model.feature_importances_)
feats, imps = zip(*(sorted(list(filter(lambda x:x[1] != 0, feat_imps)),␣
 ↪key=lambda x: x[1], reverse=True)))

plt.figure(figsize=(10, 5))
ax = sns.barplot(x=list(feats), y=list(imps))
ax.tick_params(axis='x', rotation=90)
plt.show()
```



XGBoost (Tengteng)

[ ]:
```
scalerX = StandardScaler()                                    # Set our standard␣
 ↪scaler
#scalerX = MinMaxScaler()
scalerY = StandardScaler()                                    # Set our standard␣
 ↪scaler
#scalerY = MinMaxScaler()


df_X_train = scalerX.fit_transform(X_train)                  # Fit and transform␣
 ↪scalar on X_train
df_X_test = scalerX.transform(X_test)                        # Transform X_test


df_y_train = scalerX.fit_transform(y_train.values.reshape(-1,1))                #␣
 ↪Fit and transform scalar on X_train
df_y_test = scalerX.transform(y_test.values.reshape(-1,1))                      #␣
 ↪Transform X_test
```

[ ]:
```
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV
from xgboost import XGBRegressor

n_estimators = [100, 150]
learning_rate = np.logspace(-1,0, 2)
max_depth = [7, 15]

params = {'n_estimators': n_estimators, 'learning_rate': learning_rate,␣
 ↪'max_depth': max_depth}

reg = XGBRegressor()

print()
print("Here is XGBoosting Regression result:")
grid = GridSearchCV(reg, params, cv=5, return_train_score=True, n_jobs=-1)
grid.fit(df_X_train, df_y_train)

print('Corresponding parameters are, :', grid.best_params_)
```

```
Here is XGBoosting Regression result:
[02:42:31] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear
is now deprecated in favor of reg:squarederror.
Corresponding parameters are, : {'learning_rate': 0.1, 'max_depth': 7,
'n_estimators': 150}
```

[ ]:
```
xg_y_train_pred = grid.best_estimator_.predict(df_X_train)

print("\n========Evaluation on Train Set========\n")
print("R2 score on train set : ",r2_score(df_y_train, xg_y_train_pred))
print("MSE on train set : ", mean_squared_error(df_y_train, xg_y_train_pred))
print("MAE on train set : ", mean_absolute_error(df_y_train, xg_y_train_pred))


xg_y_test_pred = grid.best_estimator_.predict(df_X_test)

print("\n========Evaluation on Test Set========\n")
print("R2 score on test set : ",r2_score(df_y_test, xg_y_test_pred))
print("MSE on test set : ", mean_squared_error(df_y_test, xg_y_test_pred))
print("MAE on test set : ", mean_absolute_error(df_y_test, xg_y_test_pred))
```

```
========Evaluation on Train Set========

R2 score on train set :  0.9948644066858797
MSE on train set :  0.005135593314120305
MAE on train set :  0.050849287279688876
```

```
========Evaluation on Test Set========

R2 score on test set :  0.7684313255489851
MSE on test set :  0.23385590366547207
MAE on test set :  0.3447049055132908
```

LightGBM (Tengteng)

[ ]:
```
import lightgbm as lgb

n_estimators = [100, 150]
learning_rate = np.logspace(-1,0, 2)
max_depth = [7, 15]


params = {'n_estimators': n_estimators, 'learning_rate': learning_rate,␣
 ↪'max_depth': max_depth}

reg = lgb.LGBMRegressor()

print()
print("Here is LightGBM Regressor result:")
grid_lgb = GridSearchCV(reg, params, cv=5, return_train_score=True, n_jobs=-1)
grid_lgb.fit(df_X_train, df_y_train)

print()
print('Corresponding parameters are, :', grid_lgb.best_params_)
```

```
Here is LightGBM Regressor result:

Corresponding parameters are, : {'learning_rate': 0.1, 'max_depth': 7,
'n_estimators': 100}

/usr/local/lib/python3.7/dist-packages/sklearn/utils/validation.py:993:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

[ ]:
```
lgb_y_train_pred = grid_lgb.best_estimator_.predict(df_X_train)

print("\n========Evaluation on Train Set========\n")
print("R2 score on train set : ",r2_score(df_y_train, lgb_y_train_pred))
print("MSE on train set : ", mean_squared_error(df_y_train, lgb_y_train_pred))
print("MAE on train set : ", mean_absolute_error(df_y_train, lgb_y_train_pred))


lgb_y_test_pred = grid_lgb.best_estimator_.predict(df_X_test)
```

```
print("\n========Evaluation on Test Set========\n")
print("R2 score on test set : ",r2_score(df_y_test, lgb_y_test_pred))
print("MSE on test set : ", mean_squared_error(df_y_test, lgb_y_test_pred))
print("MAE on test set : ", mean_absolute_error(df_y_test, lgb_y_test_pred))
```

```
========Evaluation on Train Set========

R2 score on train set :  0.9519563604350191
MSE on train set :  0.048043639564980936
MAE on train set :  0.1648385372669007

========Evaluation on Test Set========

R2 score on test set :  0.7810348307894475
MSE on test set :  0.22112791222039282
MAE on test set :  0.33388223471879186
```

Catboost (Nitya)

[ ]:
```
!pip install catboost
from catboost import CatBoostRegressor
from sklearn.model_selection import RandomizedSearchCV

catr = CatBoostRegressor()

grid = {'learning_rate': [0.03, 0.1],
        'max_depth': [3, 5, 7],
        'l2_leaf_reg': [1, 3, 5]}

print("Training Catboost...")
rand_catr = RandomizedSearchCV(catr, grid, cv=5, return_train_score=True,␣
↪n_jobs=-1)
rand_catr.fit(df_X_train, df_y_train, verbose = 1)


print(f'Best Params: {rand_catr.best_params_}')
```

```
Requirement already satisfied: catboost in /usr/local/lib/python3.7/dist-
packages (1.0.5)
Requirement already satisfied: graphviz in /usr/local/lib/python3.7/dist-
packages (from catboost) (0.10.1)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages
(from catboost) (1.4.1)
Requirement already satisfied: pandas>=0.24.0 in /usr/local/lib/python3.7/dist-
packages (from catboost) (1.3.5)
Requirement already satisfied: plotly in /usr/local/lib/python3.7/dist-packages
(from catboost) (5.5.0)
```

```
Requirement already satisfied: numpy>=1.16.0 in /usr/local/lib/python3.7/dist-
packages (from catboost) (1.21.6)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-
packages (from catboost) (3.2.2)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages
(from catboost) (1.15.0)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.7/dist-packages (from pandas>=0.24.0->catboost) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-
packages (from pandas>=0.24.0->catboost) (2022.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.7/dist-packages (from matplotlib->catboost) (1.4.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/usr/local/lib/python3.7/dist-packages (from matplotlib->catboost) (3.0.8)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-
packages (from matplotlib->catboost) (0.11.0)
Requirement already satisfied: typing-extensions in
/usr/local/lib/python3.7/dist-packages (from
kiwisolver>=1.0.1->matplotlib->catboost) (4.2.0)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.7/dist-
packages (from plotly->catboost) (8.0.1)
Training Catboost…
0:      learn: 0.9827107        total: 4.88ms   remaining: 4.88s
1:      learn: 0.9655269        total: 7.84ms   remaining: 3.91s
2:      learn: 0.9504189        total: 10.7ms   remaining: 3.57s
3:      learn: 0.9366980        total: 13.9ms   remaining: 3.46s
4:      learn: 0.9214764        total: 21.5ms   remaining: 4.28s
5:      learn: 0.9075581        total: 29.5ms   remaining: 4.89s
6:      learn: 0.8937307        total: 34.9ms   remaining: 4.95s
7:      learn: 0.8804368        total: 37.9ms   remaining: 4.7s
8:      learn: 0.8668799        total: 40.9ms   remaining: 4.5s
9:      learn: 0.8534923        total: 44.2ms   remaining: 4.37s
10:     learn: 0.8414446        total: 47.2ms   remaining: 4.25s
11:     learn: 0.8282680        total: 50.4ms   remaining: 4.15s
12:     learn: 0.8167125        total: 53.6ms   remaining: 4.07s
13:     learn: 0.8065785        total: 56.8ms   remaining: 4s
14:     learn: 0.7945816        total: 59.6ms   remaining: 3.92s
15:     learn: 0.7842481        total: 62.9ms   remaining: 3.87s
16:     learn: 0.7732247        total: 66ms     remaining: 3.82s
17:     learn: 0.7633497        total: 69ms     remaining: 3.77s
18:     learn: 0.7537773        total: 72.3ms   remaining: 3.73s
19:     learn: 0.7436438        total: 75.2ms   remaining: 3.69s
20:     learn: 0.7336916        total: 78.1ms   remaining: 3.64s
21:     learn: 0.7252978        total: 81.5ms   remaining: 3.62s
22:     learn: 0.7174009        total: 84.6ms   remaining: 3.59s
23:     learn: 0.7089060        total: 87.5ms   remaining: 3.56s
24:     learn: 0.7018508        total: 90.9ms   remaining: 3.54s
25:     learn: 0.6934555        total: 94.1ms   remaining: 3.52s
```

```
26:     learn: 0.6860228        total: 97.1ms   remaining: 3.5s
27:     learn: 0.6776117        total: 100ms    remaining: 3.48s
28:     learn: 0.6710389        total: 103ms    remaining: 3.46s
29:     learn: 0.6643926        total: 107ms    remaining: 3.44s
30:     learn: 0.6575939        total: 110ms    remaining: 3.44s
31:     learn: 0.6509907        total: 113ms    remaining: 3.41s
32:     learn: 0.6436512        total: 116ms    remaining: 3.39s
33:     learn: 0.6374014        total: 119ms    remaining: 3.38s
34:     learn: 0.6307884        total: 122ms    remaining: 3.36s
35:     learn: 0.6244608        total: 125ms    remaining: 3.35s
36:     learn: 0.6193008        total: 128ms    remaining: 3.34s
37:     learn: 0.6143099        total: 131ms    remaining: 3.32s
38:     learn: 0.6082869        total: 134ms    remaining: 3.31s
39:     learn: 0.6037112        total: 137ms    remaining: 3.3s
40:     learn: 0.5989070        total: 141ms    remaining: 3.29s
41:     learn: 0.5942192        total: 144ms    remaining: 3.28s
42:     learn: 0.5895513        total: 147ms    remaining: 3.27s
43:     learn: 0.5855265        total: 150ms    remaining: 3.26s
44:     learn: 0.5808461        total: 153ms    remaining: 3.25s
45:     learn: 0.5754597        total: 156ms    remaining: 3.23s
46:     learn: 0.5702807        total: 159ms    remaining: 3.23s
47:     learn: 0.5659465        total: 162ms    remaining: 3.22s
48:     learn: 0.5614389        total: 165ms    remaining: 3.2s
49:     learn: 0.5579985        total: 169ms    remaining: 3.2s
50:     learn: 0.5530157        total: 172ms    remaining: 3.19s
51:     learn: 0.5487417        total: 175ms    remaining: 3.18s
52:     learn: 0.5450684        total: 178ms    remaining: 3.18s
53:     learn: 0.5414306        total: 181ms    remaining: 3.17s
54:     learn: 0.5376529        total: 184ms    remaining: 3.16s
55:     learn: 0.5339162        total: 187ms    remaining: 3.16s
56:     learn: 0.5310544        total: 190ms    remaining: 3.15s
57:     learn: 0.5286000        total: 193ms    remaining: 3.14s
58:     learn: 0.5256233        total: 196ms    remaining: 3.13s
59:     learn: 0.5235150        total: 203ms    remaining: 3.18s
60:     learn: 0.5205599        total: 207ms    remaining: 3.18s
61:     learn: 0.5175741        total: 209ms    remaining: 3.17s
62:     learn: 0.5147463        total: 212ms    remaining: 3.15s
63:     learn: 0.5123176        total: 215ms    remaining: 3.15s
64:     learn: 0.5089347        total: 218ms    remaining: 3.13s
65:     learn: 0.5065564        total: 221ms    remaining: 3.12s
66:     learn: 0.5034578        total: 224ms    remaining: 3.12s
67:     learn: 0.5009684        total: 227ms    remaining: 3.11s
68:     learn: 0.4990431        total: 230ms    remaining: 3.1s
69:     learn: 0.4966597        total: 233ms    remaining: 3.1s
70:     learn: 0.4935162        total: 236ms    remaining: 3.09s
71:     learn: 0.4911610        total: 239ms    remaining: 3.08s
72:     learn: 0.4890322        total: 243ms    remaining: 3.08s
73:     learn: 0.4866404        total: 246ms    remaining: 3.07s
```

```
74:     learn: 0.4839010        total: 249ms    remaining: 3.07s
75:     learn: 0.4812218        total: 252ms    remaining: 3.06s
76:     learn: 0.4792851        total: 255ms    remaining: 3.05s
77:     learn: 0.4767787        total: 258ms    remaining: 3.04s
78:     learn: 0.4749966        total: 261ms    remaining: 3.04s
79:     learn: 0.4735583        total: 264ms    remaining: 3.03s
80:     learn: 0.4711553        total: 267ms    remaining: 3.03s
81:     learn: 0.4695001        total: 270ms    remaining: 3.02s
82:     learn: 0.4676650        total: 273ms    remaining: 3.02s
83:     learn: 0.4657880        total: 277ms    remaining: 3.02s
84:     learn: 0.4644440        total: 280ms    remaining: 3.01s
85:     learn: 0.4625330        total: 283ms    remaining: 3.01s
86:     learn: 0.4611068        total: 286ms    remaining: 3s
87:     learn: 0.4596662        total: 289ms    remaining: 3s
88:     learn: 0.4581686        total: 292ms    remaining: 2.99s
89:     learn: 0.4566008        total: 295ms    remaining: 2.98s
90:     learn: 0.4552882        total: 298ms    remaining: 2.98s
91:     learn: 0.4536967        total: 302ms    remaining: 2.98s
92:     learn: 0.4518242        total: 305ms    remaining: 2.97s
93:     learn: 0.4502535        total: 308ms    remaining: 2.96s
94:     learn: 0.4486914        total: 311ms    remaining: 2.96s
95:     learn: 0.4473664        total: 314ms    remaining: 2.96s
96:     learn: 0.4458921        total: 317ms    remaining: 2.95s
97:     learn: 0.4448693        total: 320ms    remaining: 2.95s
98:     learn: 0.4443004        total: 323ms    remaining: 2.94s
99:     learn: 0.4434989        total: 326ms    remaining: 2.94s
100:    learn: 0.4428355        total: 329ms    remaining: 2.93s
101:    learn: 0.4419910        total: 332ms    remaining: 2.93s
102:    learn: 0.4409034        total: 335ms    remaining: 2.92s
103:    learn: 0.4399473        total: 339ms    remaining: 2.92s
104:    learn: 0.4386747        total: 342ms    remaining: 2.91s
105:    learn: 0.4375659        total: 345ms    remaining: 2.91s
106:    learn: 0.4359976        total: 348ms    remaining: 2.9s
107:    learn: 0.4348545        total: 351ms    remaining: 2.9s
108:    learn: 0.4340842        total: 354ms    remaining: 2.89s
109:    learn: 0.4328873        total: 357ms    remaining: 2.89s
110:    learn: 0.4314416        total: 361ms    remaining: 2.89s
111:    learn: 0.4305980        total: 364ms    remaining: 2.88s
112:    learn: 0.4296729        total: 366ms    remaining: 2.88s
113:    learn: 0.4286424        total: 370ms    remaining: 2.87s
114:    learn: 0.4278724        total: 373ms    remaining: 2.87s
115:    learn: 0.4265957        total: 375ms    remaining: 2.86s
116:    learn: 0.4255755        total: 379ms    remaining: 2.86s
117:    learn: 0.4244277        total: 382ms    remaining: 2.85s
118:    learn: 0.4237947        total: 385ms    remaining: 2.85s
119:    learn: 0.4232806        total: 388ms    remaining: 2.84s
120:    learn: 0.4224512        total: 391ms    remaining: 2.84s
121:    learn: 0.4217995        total: 394ms    remaining: 2.83s
```

```
122:    learn: 0.4210953        total: 398ms    remaining: 2.84s
123:    learn: 0.4199933        total: 403ms    remaining: 2.84s
124:    learn: 0.4192911        total: 405ms    remaining: 2.84s
125:    learn: 0.4180702        total: 408ms    remaining: 2.83s
126:    learn: 0.4175670        total: 411ms    remaining: 2.83s
127:    learn: 0.4169847        total: 414ms    remaining: 2.82s
128:    learn: 0.4161701        total: 417ms    remaining: 2.82s
129:    learn: 0.4151108        total: 420ms    remaining: 2.81s
130:    learn: 0.4145477        total: 423ms    remaining: 2.81s
131:    learn: 0.4139129        total: 426ms    remaining: 2.8s
132:    learn: 0.4132794        total: 429ms    remaining: 2.8s
133:    learn: 0.4121353        total: 437ms    remaining: 2.82s
134:    learn: 0.4113378        total: 439ms    remaining: 2.81s
135:    learn: 0.4107389        total: 442ms    remaining: 2.81s
136:    learn: 0.4101065        total: 445ms    remaining: 2.81s
137:    learn: 0.4095135        total: 448ms    remaining: 2.8s
138:    learn: 0.4089404        total: 451ms    remaining: 2.79s
139:    learn: 0.4082144        total: 455ms    remaining: 2.79s
140:    learn: 0.4074555        total: 458ms    remaining: 2.79s
141:    learn: 0.4067639        total: 462ms    remaining: 2.79s
142:    learn: 0.4061365        total: 465ms    remaining: 2.79s
143:    learn: 0.4056056        total: 468ms    remaining: 2.78s
144:    learn: 0.4049792        total: 471ms    remaining: 2.78s
145:    learn: 0.4043610        total: 475ms    remaining: 2.77s
146:    learn: 0.4037115        total: 478ms    remaining: 2.77s
147:    learn: 0.4033425        total: 481ms    remaining: 2.77s
148:    learn: 0.4028000        total: 484ms    remaining: 2.76s
149:    learn: 0.4019708        total: 487ms    remaining: 2.76s
150:    learn: 0.4014277        total: 490ms    remaining: 2.75s
151:    learn: 0.4008229        total: 493ms    remaining: 2.75s
152:    learn: 0.4002507        total: 496ms    remaining: 2.75s
153:    learn: 0.3996946        total: 499ms    remaining: 2.74s
154:    learn: 0.3992588        total: 503ms    remaining: 2.74s
155:    learn: 0.3986876        total: 506ms    remaining: 2.73s
156:    learn: 0.3980115        total: 509ms    remaining: 2.73s
157:    learn: 0.3973042        total: 512ms    remaining: 2.73s
158:    learn: 0.3968278        total: 515ms    remaining: 2.72s
159:    learn: 0.3963891        total: 518ms    remaining: 2.72s
160:    learn: 0.3959149        total: 522ms    remaining: 2.72s
161:    learn: 0.3954298        total: 524ms    remaining: 2.71s
162:    learn: 0.3947380        total: 528ms    remaining: 2.71s
163:    learn: 0.3942310        total: 531ms    remaining: 2.71s
164:    learn: 0.3938828        total: 534ms    remaining: 2.7s
165:    learn: 0.3932959        total: 537ms    remaining: 2.7s
166:    learn: 0.3925025        total: 540ms    remaining: 2.69s
167:    learn: 0.3921199        total: 543ms    remaining: 2.69s
168:    learn: 0.3915056        total: 547ms    remaining: 2.69s
169:    learn: 0.3911121        total: 550ms    remaining: 2.69s
```

```
170:    learn: 0.3902829        total: 553ms    remaining: 2.68s
171:    learn: 0.3896720        total: 556ms    remaining: 2.67s
172:    learn: 0.3890256        total: 559ms    remaining: 2.67s
173:    learn: 0.3884055        total: 562ms    remaining: 2.67s
174:    learn: 0.3879461        total: 565ms    remaining: 2.66s
175:    learn: 0.3875166        total: 569ms    remaining: 2.66s
176:    learn: 0.3871203        total: 572ms    remaining: 2.66s
177:    learn: 0.3867836        total: 575ms    remaining: 2.65s
178:    learn: 0.3861166        total: 578ms    remaining: 2.65s
179:    learn: 0.3854853        total: 581ms    remaining: 2.65s
180:    learn: 0.3848311        total: 584ms    remaining: 2.64s
181:    learn: 0.3842597        total: 588ms    remaining: 2.64s
182:    learn: 0.3838766        total: 594ms    remaining: 2.65s
183:    learn: 0.3834929        total: 604ms    remaining: 2.68s
184:    learn: 0.3831314        total: 609ms    remaining: 2.68s
185:    learn: 0.3825736        total: 612ms    remaining: 2.68s
186:    learn: 0.3818517        total: 615ms    remaining: 2.67s
187:    learn: 0.3812085        total: 618ms    remaining: 2.67s
188:    learn: 0.3809268        total: 621ms    remaining: 2.66s
189:    learn: 0.3802744        total: 624ms    remaining: 2.66s
190:    learn: 0.3796956        total: 628ms    remaining: 2.66s
191:    learn: 0.3791390        total: 631ms    remaining: 2.65s
192:    learn: 0.3788400        total: 634ms    remaining: 2.65s
193:    learn: 0.3782584        total: 637ms    remaining: 2.65s
194:    learn: 0.3779654        total: 640ms    remaining: 2.64s
195:    learn: 0.3774170        total: 643ms    remaining: 2.64s
196:    learn: 0.3771041        total: 647ms    remaining: 2.64s
197:    learn: 0.3766180        total: 650ms    remaining: 2.63s
198:    learn: 0.3762479        total: 653ms    remaining: 2.63s
199:    learn: 0.3757335        total: 656ms    remaining: 2.63s
200:    learn: 0.3751394        total: 660ms    remaining: 2.62s
201:    learn: 0.3746481        total: 663ms    remaining: 2.62s
202:    learn: 0.3743246        total: 666ms    remaining: 2.62s
203:    learn: 0.3741069        total: 669ms    remaining: 2.61s
204:    learn: 0.3735886        total: 672ms    remaining: 2.61s
205:    learn: 0.3732210        total: 676ms    remaining: 2.6s
206:    learn: 0.3727545        total: 679ms    remaining: 2.6s
207:    learn: 0.3721588        total: 682ms    remaining: 2.6s
208:    learn: 0.3714911        total: 685ms    remaining: 2.59s
209:    learn: 0.3712710        total: 688ms    remaining: 2.59s
210:    learn: 0.3707968        total: 691ms    remaining: 2.58s
211:    learn: 0.3704607        total: 695ms    remaining: 2.58s
212:    learn: 0.3699765        total: 698ms    remaining: 2.58s
213:    learn: 0.3693712        total: 701ms    remaining: 2.57s
214:    learn: 0.3690869        total: 705ms    remaining: 2.57s
215:    learn: 0.3685389        total: 708ms    remaining: 2.57s
216:    learn: 0.3680403        total: 711ms    remaining: 2.56s
217:    learn: 0.3674312        total: 714ms    remaining: 2.56s
```

```
218:    learn: 0.3669835        total: 718ms    remaining: 2.56s
219:    learn: 0.3665678        total: 721ms    remaining: 2.56s
220:    learn: 0.3663156        total: 725ms    remaining: 2.55s
221:    learn: 0.3660976        total: 728ms    remaining: 2.55s
222:    learn: 0.3656527        total: 731ms    remaining: 2.54s
223:    learn: 0.3652874        total: 734ms    remaining: 2.54s
224:    learn: 0.3650034        total: 737ms    remaining: 2.54s
225:    learn: 0.3645569        total: 740ms    remaining: 2.53s
226:    learn: 0.3640764        total: 743ms    remaining: 2.53s
227:    learn: 0.3637389        total: 746ms    remaining: 2.53s
228:    learn: 0.3632701        total: 749ms    remaining: 2.52s
229:    learn: 0.3625769        total: 753ms    remaining: 2.52s
230:    learn: 0.3621559        total: 756ms    remaining: 2.52s
231:    learn: 0.3617416        total: 759ms    remaining: 2.51s
232:    learn: 0.3612724        total: 766ms    remaining: 2.52s
233:    learn: 0.3608917        total: 769ms    remaining: 2.52s
234:    learn: 0.3605548        total: 772ms    remaining: 2.51s
235:    learn: 0.3601491        total: 775ms    remaining: 2.51s
236:    learn: 0.3596670        total: 778ms    remaining: 2.5s
237:    learn: 0.3591000        total: 781ms    remaining: 2.5s
238:    learn: 0.3582760        total: 784ms    remaining: 2.5s
239:    learn: 0.3577918        total: 787ms    remaining: 2.49s
240:    learn: 0.3574919        total: 793ms    remaining: 2.5s
241:    learn: 0.3570694        total: 797ms    remaining: 2.5s
242:    learn: 0.3566944        total: 801ms    remaining: 2.49s
243:    learn: 0.3564330        total: 804ms    remaining: 2.49s
244:    learn: 0.3561140        total: 807ms    remaining: 2.49s
245:    learn: 0.3555835        total: 810ms    remaining: 2.48s
246:    learn: 0.3551715        total: 814ms    remaining: 2.48s
247:    learn: 0.3549527        total: 819ms    remaining: 2.48s
248:    learn: 0.3545276        total: 822ms    remaining: 2.48s
249:    learn: 0.3539866        total: 826ms    remaining: 2.48s
250:    learn: 0.3536796        total: 829ms    remaining: 2.47s
251:    learn: 0.3530706        total: 832ms    remaining: 2.47s
252:    learn: 0.3525998        total: 835ms    remaining: 2.46s
253:    learn: 0.3520914        total: 838ms    remaining: 2.46s
254:    learn: 0.3517401        total: 842ms    remaining: 2.46s
255:    learn: 0.3511620        total: 844ms    remaining: 2.45s
256:    learn: 0.3507303        total: 847ms    remaining: 2.45s
257:    learn: 0.3503318        total: 851ms    remaining: 2.45s
258:    learn: 0.3500448        total: 854ms    remaining: 2.44s
259:    learn: 0.3496282        total: 857ms    remaining: 2.44s
260:    learn: 0.3491745        total: 860ms    remaining: 2.44s
261:    learn: 0.3488041        total: 863ms    remaining: 2.43s
262:    learn: 0.3483788        total: 866ms    remaining: 2.43s
263:    learn: 0.3480180        total: 870ms    remaining: 2.42s
264:    learn: 0.3476384        total: 873ms    remaining: 2.42s
265:    learn: 0.3471361        total: 876ms    remaining: 2.42s
```

```
266:    learn: 0.3468241        total: 879ms    remaining: 2.41s
267:    learn: 0.3462413        total: 882ms    remaining: 2.41s
268:    learn: 0.3459332        total: 885ms    remaining: 2.4s
269:    learn: 0.3453147        total: 888ms    remaining: 2.4s
270:    learn: 0.3449393        total: 892ms    remaining: 2.4s
271:    learn: 0.3443518        total: 895ms    remaining: 2.39s
272:    learn: 0.3438437        total: 898ms    remaining: 2.39s
273:    learn: 0.3435384        total: 901ms    remaining: 2.39s
274:    learn: 0.3431538        total: 904ms    remaining: 2.38s
275:    learn: 0.3425958        total: 907ms    remaining: 2.38s
276:    learn: 0.3422907        total: 910ms    remaining: 2.37s
277:    learn: 0.3418824        total: 913ms    remaining: 2.37s
278:    learn: 0.3413369        total: 916ms    remaining: 2.37s
279:    learn: 0.3409664        total: 919ms    remaining: 2.36s
280:    learn: 0.3404419        total: 922ms    remaining: 2.36s
281:    learn: 0.3400684        total: 926ms    remaining: 2.36s
282:    learn: 0.3396227        total: 928ms    remaining: 2.35s
283:    learn: 0.3392939        total: 931ms    remaining: 2.35s
284:    learn: 0.3389803        total: 934ms    remaining: 2.34s
285:    learn: 0.3385931        total: 938ms    remaining: 2.34s
286:    learn: 0.3382573        total: 940ms    remaining: 2.34s
287:    learn: 0.3378476        total: 943ms    remaining: 2.33s
288:    learn: 0.3374253        total: 947ms    remaining: 2.33s
289:    learn: 0.3372038        total: 950ms    remaining: 2.32s
290:    learn: 0.3368959        total: 953ms    remaining: 2.32s
291:    learn: 0.3363577        total: 956ms    remaining: 2.32s
292:    learn: 0.3360276        total: 961ms    remaining: 2.32s
293:    learn: 0.3357511        total: 964ms    remaining: 2.31s
294:    learn: 0.3355192        total: 967ms    remaining: 2.31s
295:    learn: 0.3351597        total: 970ms    remaining: 2.31s
296:    learn: 0.3347934        total: 973ms    remaining: 2.3s
297:    learn: 0.3343358        total: 976ms    remaining: 2.3s
298:    learn: 0.3339986        total: 979ms    remaining: 2.29s
299:    learn: 0.3336667        total: 982ms    remaining: 2.29s
300:    learn: 0.3334428        total: 985ms    remaining: 2.29s
301:    learn: 0.3330536        total: 991ms    remaining: 2.29s
302:    learn: 0.3327122        total: 1000ms   remaining: 2.3s
303:    learn: 0.3324794        total: 1.01s    remaining: 2.31s
304:    learn: 0.3319972        total: 1.01s    remaining: 2.31s
305:    learn: 0.3317190        total: 1.01s    remaining: 2.3s
306:    learn: 0.3313393        total: 1.02s    remaining: 2.3s
307:    learn: 0.3309484        total: 1.02s    remaining: 2.29s
308:    learn: 0.3305084        total: 1.02s    remaining: 2.29s
309:    learn: 0.3301091        total: 1.03s    remaining: 2.29s
310:    learn: 0.3298155        total: 1.03s    remaining: 2.28s
311:    learn: 0.3292964        total: 1.03s    remaining: 2.28s
312:    learn: 0.3290170        total: 1.03s    remaining: 2.27s
313:    learn: 0.3285003        total: 1.04s    remaining: 2.27s
```

```
314:    learn: 0.3282849        total: 1.04s    remaining: 2.27s
315:    learn: 0.3279096        total: 1.04s    remaining: 2.26s
316:    learn: 0.3274942        total: 1.05s    remaining: 2.26s
317:    learn: 0.3270677        total: 1.05s    remaining: 2.25s
318:    learn: 0.3267778        total: 1.05s    remaining: 2.25s
319:    learn: 0.3265777        total: 1.06s    remaining: 2.25s
320:    learn: 0.3260716        total: 1.06s    remaining: 2.24s
321:    learn: 0.3258062        total: 1.06s    remaining: 2.24s
322:    learn: 0.3255994        total: 1.07s    remaining: 2.23s
323:    learn: 0.3252777        total: 1.07s    remaining: 2.23s
324:    learn: 0.3249597        total: 1.07s    remaining: 2.23s
325:    learn: 0.3246742        total: 1.07s    remaining: 2.22s
326:    learn: 0.3243974        total: 1.08s    remaining: 2.22s
327:    learn: 0.3240590        total: 1.08s    remaining: 2.22s
328:    learn: 0.3237655        total: 1.08s    remaining: 2.21s
329:    learn: 0.3233802        total: 1.09s    remaining: 2.21s
330:    learn: 0.3229905        total: 1.09s    remaining: 2.21s
331:    learn: 0.3226894        total: 1.09s    remaining: 2.2s
332:    learn: 0.3223690        total: 1.1s     remaining: 2.2s
333:    learn: 0.3223117        total: 1.1s     remaining: 2.19s
334:    learn: 0.3220016        total: 1.1s     remaining: 2.19s
335:    learn: 0.3216683        total: 1.11s    remaining: 2.19s
336:    learn: 0.3213071        total: 1.11s    remaining: 2.18s
337:    learn: 0.3212096        total: 1.11s    remaining: 2.18s
338:    learn: 0.3208887        total: 1.11s    remaining: 2.17s
339:    learn: 0.3207614        total: 1.12s    remaining: 2.17s
340:    learn: 0.3203833        total: 1.12s    remaining: 2.17s
341:    learn: 0.3200767        total: 1.13s    remaining: 2.16s
342:    learn: 0.3196983        total: 1.13s    remaining: 2.16s
343:    learn: 0.3193527        total: 1.13s    remaining: 2.16s
344:    learn: 0.3190555        total: 1.13s    remaining: 2.15s
345:    learn: 0.3188762        total: 1.14s    remaining: 2.15s
346:    learn: 0.3184471        total: 1.14s    remaining: 2.15s
347:    learn: 0.3183359        total: 1.14s    remaining: 2.14s
348:    learn: 0.3178176        total: 1.15s    remaining: 2.14s
349:    learn: 0.3173876        total: 1.15s    remaining: 2.13s
350:    learn: 0.3170744        total: 1.15s    remaining: 2.13s
351:    learn: 0.3169743        total: 1.16s    remaining: 2.13s
352:    learn: 0.3163685        total: 1.16s    remaining: 2.12s
353:    learn: 0.3161285        total: 1.16s    remaining: 2.12s
354:    learn: 0.3158970        total: 1.17s    remaining: 2.12s
355:    learn: 0.3155847        total: 1.17s    remaining: 2.11s
356:    learn: 0.3151753        total: 1.17s    remaining: 2.11s
357:    learn: 0.3148713        total: 1.17s    remaining: 2.1s
358:    learn: 0.3144190        total: 1.18s    remaining: 2.1s
359:    learn: 0.3140998        total: 1.18s    remaining: 2.1s
360:    learn: 0.3136892        total: 1.18s    remaining: 2.09s
361:    learn: 0.3132847        total: 1.19s    remaining: 2.09s
```

```
362:    learn: 0.3132370        total: 1.19s    remaining: 2.09s
363:    learn: 0.3130365        total: 1.2s     remaining: 2.09s
364:    learn: 0.3125643        total: 1.2s     remaining: 2.09s
365:    learn: 0.3122668        total: 1.2s     remaining: 2.08s
366:    learn: 0.3119834        total: 1.21s    remaining: 2.08s
367:    learn: 0.3116524        total: 1.21s    remaining: 2.08s
368:    learn: 0.3111770        total: 1.21s    remaining: 2.07s
369:    learn: 0.3107933        total: 1.22s    remaining: 2.07s
370:    learn: 0.3107573        total: 1.22s    remaining: 2.06s
371:    learn: 0.3104239        total: 1.22s    remaining: 2.06s
372:    learn: 0.3099796        total: 1.23s    remaining: 2.06s
373:    learn: 0.3095875        total: 1.23s    remaining: 2.06s
374:    learn: 0.3092725        total: 1.23s    remaining: 2.05s
375:    learn: 0.3091573        total: 1.24s    remaining: 2.05s
376:    learn: 0.3091217        total: 1.24s    remaining: 2.04s
377:    learn: 0.3086718        total: 1.24s    remaining: 2.04s
378:    learn: 0.3085476        total: 1.24s    remaining: 2.04s
379:    learn: 0.3083841        total: 1.25s    remaining: 2.04s
380:    learn: 0.3083432        total: 1.25s    remaining: 2.03s
381:    learn: 0.3083091        total: 1.25s    remaining: 2.03s
382:    learn: 0.3081094        total: 1.26s    remaining: 2.02s
383:    learn: 0.3075481        total: 1.26s    remaining: 2.02s
384:    learn: 0.3071474        total: 1.26s    remaining: 2.02s
385:    learn: 0.3068887        total: 1.26s    remaining: 2.01s
386:    learn: 0.3068574        total: 1.27s    remaining: 2.01s
387:    learn: 0.3063982        total: 1.27s    remaining: 2s
388:    learn: 0.3060883        total: 1.27s    remaining: 2s
389:    learn: 0.3060501        total: 1.28s    remaining: 2s
390:    learn: 0.3057849        total: 1.28s    remaining: 1.99s
391:    learn: 0.3057575        total: 1.28s    remaining: 1.99s
392:    learn: 0.3052564        total: 1.29s    remaining: 1.99s
393:    learn: 0.3048711        total: 1.29s    remaining: 1.98s
394:    learn: 0.3048409        total: 1.29s    remaining: 1.98s
395:    learn: 0.3047847        total: 1.29s    remaining: 1.98s
396:    learn: 0.3043777        total: 1.3s     remaining: 1.97s
397:    learn: 0.3040173        total: 1.3s     remaining: 1.97s
398:    learn: 0.3039528        total: 1.3s     remaining: 1.97s
399:    learn: 0.3037040        total: 1.31s    remaining: 1.96s
400:    learn: 0.3032906        total: 1.31s    remaining: 1.96s
401:    learn: 0.3029777        total: 1.31s    remaining: 1.96s
402:    learn: 0.3027165        total: 1.32s    remaining: 1.95s
403:    learn: 0.3025614        total: 1.32s    remaining: 1.95s
404:    learn: 0.3020912        total: 1.32s    remaining: 1.95s
405:    learn: 0.3018451        total: 1.33s    remaining: 1.94s
406:    learn: 0.3016527        total: 1.33s    remaining: 1.94s
407:    learn: 0.3015919        total: 1.33s    remaining: 1.94s
408:    learn: 0.3012001        total: 1.34s    remaining: 1.93s
409:    learn: 0.3011752        total: 1.34s    remaining: 1.93s
```

```
410:    learn: 0.3008838        total: 1.34s    remaining: 1.93s
411:    learn: 0.3006083        total: 1.35s    remaining: 1.92s
412:    learn: 0.3004078        total: 1.35s    remaining: 1.92s
413:    learn: 0.3002012        total: 1.35s    remaining: 1.92s
414:    learn: 0.3001178        total: 1.36s    remaining: 1.91s
415:    learn: 0.2997678        total: 1.36s    remaining: 1.91s
416:    learn: 0.2992540        total: 1.36s    remaining: 1.91s
417:    learn: 0.2989950        total: 1.37s    remaining: 1.9s
418:    learn: 0.2987206        total: 1.37s    remaining: 1.9s
419:    learn: 0.2986808        total: 1.37s    remaining: 1.9s
420:    learn: 0.2981085        total: 1.38s    remaining: 1.89s
421:    learn: 0.2978565        total: 1.38s    remaining: 1.89s
422:    learn: 0.2975623        total: 1.38s    remaining: 1.89s
423:    learn: 0.2971618        total: 1.39s    remaining: 1.88s
424:    learn: 0.2968941        total: 1.39s    remaining: 1.88s
425:    learn: 0.2966845        total: 1.4s     remaining: 1.88s
426:    learn: 0.2961733        total: 1.4s     remaining: 1.88s
427:    learn: 0.2959120        total: 1.4s     remaining: 1.87s
428:    learn: 0.2954805        total: 1.4s     remaining: 1.87s
429:    learn: 0.2951225        total: 1.41s    remaining: 1.87s
430:    learn: 0.2948618        total: 1.41s    remaining: 1.86s
431:    learn: 0.2946094        total: 1.41s    remaining: 1.86s
432:    learn: 0.2943840        total: 1.42s    remaining: 1.86s
433:    learn: 0.2943512        total: 1.42s    remaining: 1.85s
434:    learn: 0.2941116        total: 1.42s    remaining: 1.85s
435:    learn: 0.2937312        total: 1.43s    remaining: 1.85s
436:    learn: 0.2932490        total: 1.43s    remaining: 1.84s
437:    learn: 0.2928659        total: 1.43s    remaining: 1.84s
438:    learn: 0.2926966        total: 1.44s    remaining: 1.84s
439:    learn: 0.2924263        total: 1.44s    remaining: 1.83s
440:    learn: 0.2921706        total: 1.44s    remaining: 1.83s
441:    learn: 0.2918039        total: 1.45s    remaining: 1.83s
442:    learn: 0.2917780        total: 1.45s    remaining: 1.82s
443:    learn: 0.2915749        total: 1.45s    remaining: 1.82s
444:    learn: 0.2913062        total: 1.46s    remaining: 1.81s
445:    learn: 0.2912912        total: 1.46s    remaining: 1.81s
446:    learn: 0.2911928        total: 1.46s    remaining: 1.81s
447:    learn: 0.2909355        total: 1.46s    remaining: 1.8s
448:    learn: 0.2906245        total: 1.47s    remaining: 1.8s
449:    learn: 0.2902732        total: 1.47s    remaining: 1.8s
450:    learn: 0.2899616        total: 1.47s    remaining: 1.79s
451:    learn: 0.2897829        total: 1.48s    remaining: 1.79s
452:    learn: 0.2895286        total: 1.48s    remaining: 1.79s
453:    learn: 0.2891551        total: 1.48s    remaining: 1.78s
454:    learn: 0.2888420        total: 1.49s    remaining: 1.78s
455:    learn: 0.2885878        total: 1.49s    remaining: 1.78s
456:    learn: 0.2883462        total: 1.49s    remaining: 1.77s
457:    learn: 0.2878400        total: 1.5s     remaining: 1.77s
```

```
458:    learn: 0.2874377        total: 1.5s     remaining: 1.77s
459:    learn: 0.2870542        total: 1.5s     remaining: 1.76s
460:    learn: 0.2869648        total: 1.5s     remaining: 1.76s
461:    learn: 0.2865520        total: 1.51s    remaining: 1.76s
462:    learn: 0.2863631        total: 1.51s    remaining: 1.75s
463:    learn: 0.2861793        total: 1.51s    remaining: 1.75s
464:    learn: 0.2860867        total: 1.52s    remaining: 1.75s
465:    learn: 0.2857281        total: 1.52s    remaining: 1.75s
466:    learn: 0.2853487        total: 1.53s    remaining: 1.74s
467:    learn: 0.2848835        total: 1.53s    remaining: 1.74s
468:    learn: 0.2845134        total: 1.53s    remaining: 1.74s
469:    learn: 0.2844673        total: 1.54s    remaining: 1.73s
470:    learn: 0.2843917        total: 1.54s    remaining: 1.73s
471:    learn: 0.2843080        total: 1.54s    remaining: 1.73s
472:    learn: 0.2840396        total: 1.55s    remaining: 1.72s
473:    learn: 0.2837989        total: 1.55s    remaining: 1.72s
474:    learn: 0.2835503        total: 1.55s    remaining: 1.72s
475:    learn: 0.2832974        total: 1.55s    remaining: 1.71s
476:    learn: 0.2832520        total: 1.56s    remaining: 1.71s
477:    learn: 0.2829295        total: 1.56s    remaining: 1.71s
478:    learn: 0.2825976        total: 1.56s    remaining: 1.7s
479:    learn: 0.2825767        total: 1.57s    remaining: 1.7s
480:    learn: 0.2822759        total: 1.57s    remaining: 1.7s
481:    learn: 0.2819864        total: 1.57s    remaining: 1.69s
482:    learn: 0.2817572        total: 1.58s    remaining: 1.69s
483:    learn: 0.2817429        total: 1.58s    remaining: 1.68s
484:    learn: 0.2814117        total: 1.58s    remaining: 1.68s
485:    learn: 0.2813231        total: 1.59s    remaining: 1.68s
486:    learn: 0.2812322        total: 1.59s    remaining: 1.68s
487:    learn: 0.2809880        total: 1.6s     remaining: 1.68s
488:    learn: 0.2809632        total: 1.6s     remaining: 1.67s
489:    learn: 0.2808274        total: 1.6s     remaining: 1.67s
490:    learn: 0.2804848        total: 1.6s     remaining: 1.66s
491:    learn: 0.2802176        total: 1.61s    remaining: 1.66s
492:    learn: 0.2799894        total: 1.61s    remaining: 1.66s
493:    learn: 0.2796431        total: 1.61s    remaining: 1.65s
494:    learn: 0.2794490        total: 1.62s    remaining: 1.65s
495:    learn: 0.2793438        total: 1.62s    remaining: 1.65s
496:    learn: 0.2791250        total: 1.62s    remaining: 1.64s
497:    learn: 0.2790850        total: 1.63s    remaining: 1.64s
498:    learn: 0.2787507        total: 1.63s    remaining: 1.64s
499:    learn: 0.2785307        total: 1.63s    remaining: 1.63s
500:    learn: 0.2783276        total: 1.64s    remaining: 1.63s
501:    learn: 0.2781150        total: 1.64s    remaining: 1.63s
502:    learn: 0.2777883        total: 1.64s    remaining: 1.62s
503:    learn: 0.2777391        total: 1.65s    remaining: 1.62s
504:    learn: 0.2774417        total: 1.65s    remaining: 1.61s
505:    learn: 0.2774227        total: 1.65s    remaining: 1.61s
```

```
506:    learn: 0.2771573        total: 1.65s    remaining: 1.61s
507:    learn: 0.2769331        total: 1.66s    remaining: 1.6s
508:    learn: 0.2767458        total: 1.66s    remaining: 1.6s
509:    learn: 0.2764884        total: 1.66s    remaining: 1.6s
510:    learn: 0.2762200        total: 1.67s    remaining: 1.59s
511:    learn: 0.2759885        total: 1.67s    remaining: 1.59s
512:    learn: 0.2756204        total: 1.67s    remaining: 1.59s
513:    learn: 0.2754560        total: 1.67s    remaining: 1.58s
514:    learn: 0.2752848        total: 1.68s    remaining: 1.58s
515:    learn: 0.2750510        total: 1.68s    remaining: 1.58s
516:    learn: 0.2747510        total: 1.68s    remaining: 1.57s
517:    learn: 0.2745620        total: 1.69s    remaining: 1.57s
518:    learn: 0.2743889        total: 1.69s    remaining: 1.56s
519:    learn: 0.2739728        total: 1.69s    remaining: 1.56s
520:    learn: 0.2736283        total: 1.7s     remaining: 1.56s
521:    learn: 0.2734316        total: 1.7s     remaining: 1.55s
522:    learn: 0.2731621        total: 1.7s     remaining: 1.55s
523:    learn: 0.2729874        total: 1.71s    remaining: 1.55s
524:    learn: 0.2727757        total: 1.71s    remaining: 1.55s
525:    learn: 0.2725471        total: 1.71s    remaining: 1.54s
526:    learn: 0.2725303        total: 1.71s    remaining: 1.54s
527:    learn: 0.2722146        total: 1.72s    remaining: 1.53s
528:    learn: 0.2721095        total: 1.72s    remaining: 1.53s
529:    learn: 0.2719031        total: 1.72s    remaining: 1.53s
530:    learn: 0.2718552        total: 1.73s    remaining: 1.52s
531:    learn: 0.2716921        total: 1.73s    remaining: 1.52s
532:    learn: 0.2714623        total: 1.73s    remaining: 1.52s
533:    learn: 0.2711769        total: 1.74s    remaining: 1.51s
534:    learn: 0.2709086        total: 1.74s    remaining: 1.51s
535:    learn: 0.2706641        total: 1.74s    remaining: 1.51s
536:    learn: 0.2705950        total: 1.75s    remaining: 1.5s
537:    learn: 0.2703562        total: 1.75s    remaining: 1.5s
538:    learn: 0.2701609        total: 1.75s    remaining: 1.5s
539:    learn: 0.2698678        total: 1.75s    remaining: 1.5s
540:    learn: 0.2698280        total: 1.76s    remaining: 1.49s
541:    learn: 0.2695468        total: 1.76s    remaining: 1.49s
542:    learn: 0.2693119        total: 1.76s    remaining: 1.49s
543:    learn: 0.2692820        total: 1.77s    remaining: 1.48s
544:    learn: 0.2692442        total: 1.77s    remaining: 1.48s
545:    learn: 0.2689174        total: 1.77s    remaining: 1.48s
546:    learn: 0.2687655        total: 1.78s    remaining: 1.47s
547:    learn: 0.2683454        total: 1.78s    remaining: 1.47s
548:    learn: 0.2682803        total: 1.79s    remaining: 1.47s
549:    learn: 0.2680416        total: 1.79s    remaining: 1.47s
550:    learn: 0.2676661        total: 1.79s    remaining: 1.46s
551:    learn: 0.2676491        total: 1.8s     remaining: 1.46s
552:    learn: 0.2672009        total: 1.8s     remaining: 1.46s
553:    learn: 0.2668600        total: 1.8s     remaining: 1.45s
```

```
554:    learn: 0.2664064        total: 1.81s    remaining: 1.45s
555:    learn: 0.2663929        total: 1.81s    remaining: 1.45s
556:    learn: 0.2662056        total: 1.81s    remaining: 1.44s
557:    learn: 0.2660139        total: 1.82s    remaining: 1.44s
558:    learn: 0.2657098        total: 1.82s    remaining: 1.44s
559:    learn: 0.2654068        total: 1.82s    remaining: 1.43s
560:    learn: 0.2653603        total: 1.83s    remaining: 1.43s
561:    learn: 0.2649720        total: 1.83s    remaining: 1.43s
562:    learn: 0.2647814        total: 1.84s    remaining: 1.43s
563:    learn: 0.2643122        total: 1.84s    remaining: 1.42s
564:    learn: 0.2642852        total: 1.84s    remaining: 1.42s
565:    learn: 0.2639711        total: 1.84s    remaining: 1.42s
566:    learn: 0.2637063        total: 1.85s    remaining: 1.41s
567:    learn: 0.2634968        total: 1.85s    remaining: 1.41s
568:    learn: 0.2633448        total: 1.85s    remaining: 1.41s
569:    learn: 0.2632641        total: 1.86s    remaining: 1.4s
570:    learn: 0.2629475        total: 1.86s    remaining: 1.4s
571:    learn: 0.2627791        total: 1.86s    remaining: 1.4s
572:    learn: 0.2625550        total: 1.87s    remaining: 1.39s
573:    learn: 0.2625439        total: 1.87s    remaining: 1.39s
574:    learn: 0.2623499        total: 1.87s    remaining: 1.39s
575:    learn: 0.2621918        total: 1.88s    remaining: 1.38s
576:    learn: 0.2619109        total: 1.88s    remaining: 1.38s
577:    learn: 0.2618405        total: 1.88s    remaining: 1.38s
578:    learn: 0.2616307        total: 1.89s    remaining: 1.37s
579:    learn: 0.2612859        total: 1.89s    remaining: 1.37s
580:    learn: 0.2608188        total: 1.89s    remaining: 1.36s
581:    learn: 0.2605995        total: 1.9s     remaining: 1.36s
582:    learn: 0.2602539        total: 1.9s     remaining: 1.36s
583:    learn: 0.2601276        total: 1.9s     remaining: 1.35s
584:    learn: 0.2598679        total: 1.91s    remaining: 1.35s
585:    learn: 0.2598408        total: 1.91s    remaining: 1.35s
586:    learn: 0.2594913        total: 1.91s    remaining: 1.34s
587:    learn: 0.2590898        total: 1.92s    remaining: 1.34s
588:    learn: 0.2589656        total: 1.92s    remaining: 1.34s
589:    learn: 0.2587272        total: 1.92s    remaining: 1.34s
590:    learn: 0.2587081        total: 1.93s    remaining: 1.33s
591:    learn: 0.2585697        total: 1.93s    remaining: 1.33s
592:    learn: 0.2584129        total: 1.93s    remaining: 1.33s
593:    learn: 0.2580131        total: 1.94s    remaining: 1.32s
594:    learn: 0.2577845        total: 1.94s    remaining: 1.32s
595:    learn: 0.2576504        total: 1.94s    remaining: 1.32s
596:    learn: 0.2573247        total: 1.94s    remaining: 1.31s
597:    learn: 0.2571011        total: 1.95s    remaining: 1.31s
598:    learn: 0.2570876        total: 1.95s    remaining: 1.31s
599:    learn: 0.2568975        total: 1.95s    remaining: 1.3s
600:    learn: 0.2567441        total: 1.96s    remaining: 1.3s
601:    learn: 0.2565521        total: 1.96s    remaining: 1.3s
```

```
602:    learn: 0.2565206        total: 1.96s    remaining: 1.29s
603:    learn: 0.2564307        total: 1.97s    remaining: 1.29s
604:    learn: 0.2564182        total: 1.97s    remaining: 1.29s
605:    learn: 0.2564097        total: 1.98s    remaining: 1.29s
606:    learn: 0.2561647        total: 1.99s    remaining: 1.29s
607:    learn: 0.2557978        total: 1.99s    remaining: 1.28s
608:    learn: 0.2553549        total: 2s       remaining: 1.28s
609:    learn: 0.2551348        total: 2s       remaining: 1.28s
610:    learn: 0.2549816        total: 2s       remaining: 1.27s
611:    learn: 0.2547888        total: 2s       remaining: 1.27s
612:    learn: 0.2547654        total: 2.01s    remaining: 1.27s
613:    learn: 0.2545709        total: 2.01s    remaining: 1.26s
614:    learn: 0.2543814        total: 2.01s    remaining: 1.26s
615:    learn: 0.2541635        total: 2.02s    remaining: 1.26s
616:    learn: 0.2539681        total: 2.02s    remaining: 1.25s
617:    learn: 0.2539378        total: 2.02s    remaining: 1.25s
618:    learn: 0.2537829        total: 2.03s    remaining: 1.25s
619:    learn: 0.2534416        total: 2.03s    remaining: 1.24s
620:    learn: 0.2534170        total: 2.03s    remaining: 1.24s
621:    learn: 0.2531610        total: 2.04s    remaining: 1.24s
622:    learn: 0.2528626        total: 2.04s    remaining: 1.23s
623:    learn: 0.2526167        total: 2.04s    remaining: 1.23s
624:    learn: 0.2523874        total: 2.05s    remaining: 1.23s
625:    learn: 0.2519677        total: 2.05s    remaining: 1.22s
626:    learn: 0.2518111        total: 2.05s    remaining: 1.22s
627:    learn: 0.2516209        total: 2.06s    remaining: 1.22s
628:    learn: 0.2513761        total: 2.06s    remaining: 1.21s
629:    learn: 0.2512743        total: 2.06s    remaining: 1.21s
630:    learn: 0.2511947        total: 2.06s    remaining: 1.21s
631:    learn: 0.2509002        total: 2.07s    remaining: 1.2s
632:    learn: 0.2507234        total: 2.07s    remaining: 1.2s
633:    learn: 0.2505446        total: 2.07s    remaining: 1.2s
634:    learn: 0.2505135        total: 2.08s    remaining: 1.19s
635:    learn: 0.2502229        total: 2.08s    remaining: 1.19s
636:    learn: 0.2502106        total: 2.08s    remaining: 1.19s
637:    learn: 0.2500598        total: 2.08s    remaining: 1.18s
638:    learn: 0.2497958        total: 2.09s    remaining: 1.18s
639:    learn: 0.2494204        total: 2.09s    remaining: 1.18s
640:    learn: 0.2491630        total: 2.1s     remaining: 1.17s
641:    learn: 0.2490190        total: 2.1s     remaining: 1.17s
642:    learn: 0.2487124        total: 2.1s     remaining: 1.17s
643:    learn: 0.2486881        total: 2.1s     remaining: 1.16s
644:    learn: 0.2483947        total: 2.11s    remaining: 1.16s
645:    learn: 0.2481200        total: 2.11s    remaining: 1.16s
646:    learn: 0.2478612        total: 2.11s    remaining: 1.15s
647:    learn: 0.2477335        total: 2.12s    remaining: 1.15s
648:    learn: 0.2474466        total: 2.12s    remaining: 1.15s
649:    learn: 0.2472537        total: 2.12s    remaining: 1.14s
```

```
650:    learn: 0.2469652        total: 2.13s    remaining: 1.14s
651:    learn: 0.2467832        total: 2.13s    remaining: 1.14s
652:    learn: 0.2466224        total: 2.13s    remaining: 1.13s
653:    learn: 0.2464255        total: 2.13s    remaining: 1.13s
654:    learn: 0.2463274        total: 2.14s    remaining: 1.13s
655:    learn: 0.2461457        total: 2.14s    remaining: 1.12s
656:    learn: 0.2460558        total: 2.14s    remaining: 1.12s
657:    learn: 0.2458846        total: 2.15s    remaining: 1.12s
658:    learn: 0.2455059        total: 2.15s    remaining: 1.11s
659:    learn: 0.2451460        total: 2.15s    remaining: 1.11s
660:    learn: 0.2449708        total: 2.16s    remaining: 1.1s
661:    learn: 0.2447682        total: 2.16s    remaining: 1.1s
662:    learn: 0.2446255        total: 2.16s    remaining: 1.1s
663:    learn: 0.2443746        total: 2.17s    remaining: 1.09s
664:    learn: 0.2441475        total: 2.17s    remaining: 1.09s
665:    learn: 0.2438251        total: 2.17s    remaining: 1.09s
666:    learn: 0.2434464        total: 2.17s    remaining: 1.08s
667:    learn: 0.2433918        total: 2.18s    remaining: 1.08s
668:    learn: 0.2431945        total: 2.19s    remaining: 1.08s
669:    learn: 0.2429251        total: 2.19s    remaining: 1.08s
670:    learn: 0.2428964        total: 2.19s    remaining: 1.07s
671:    learn: 0.2426840        total: 2.19s    remaining: 1.07s
672:    learn: 0.2425817        total: 2.2s     remaining: 1.07s
673:    learn: 0.2424548        total: 2.2s     remaining: 1.06s
674:    learn: 0.2422082        total: 2.2s     remaining: 1.06s
675:    learn: 0.2420168        total: 2.21s    remaining: 1.06s
676:    learn: 0.2418276        total: 2.21s    remaining: 1.06s
677:    learn: 0.2416180        total: 2.22s    remaining: 1.05s
678:    learn: 0.2412427        total: 2.22s    remaining: 1.05s
679:    learn: 0.2412280        total: 2.22s    remaining: 1.05s
680:    learn: 0.2412049        total: 2.23s    remaining: 1.04s
681:    learn: 0.2409497        total: 2.23s    remaining: 1.04s
682:    learn: 0.2406569        total: 2.23s    remaining: 1.04s
683:    learn: 0.2404911        total: 2.24s    remaining: 1.03s
684:    learn: 0.2402646        total: 2.24s    remaining: 1.03s
685:    learn: 0.2399450        total: 2.24s    remaining: 1.03s
686:    learn: 0.2398183        total: 2.25s    remaining: 1.02s
687:    learn: 0.2396608        total: 2.25s    remaining: 1.02s
688:    learn: 0.2394613        total: 2.25s    remaining: 1.02s
689:    learn: 0.2393626        total: 2.25s    remaining: 1.01s
690:    learn: 0.2391162        total: 2.26s    remaining: 1.01s
691:    learn: 0.2388193        total: 2.26s    remaining: 1.01s
692:    learn: 0.2385761        total: 2.27s    remaining: 1s
693:    learn: 0.2383877        total: 2.27s    remaining: 1s
694:    learn: 0.2383105        total: 2.27s    remaining: 997ms
695:    learn: 0.2380411        total: 2.27s    remaining: 994ms
696:    learn: 0.2377778        total: 2.28s    remaining: 990ms
697:    learn: 0.2375927        total: 2.28s    remaining: 987ms
```

```
698:    learn: 0.2373272        total: 2.28s    remaining: 984ms
699:    learn: 0.2372035        total: 2.29s    remaining: 980ms
700:    learn: 0.2369018        total: 2.29s    remaining: 977ms
701:    learn: 0.2368696        total: 2.29s    remaining: 974ms
702:    learn: 0.2367635        total: 2.3s     remaining: 970ms
703:    learn: 0.2365455        total: 2.3s     remaining: 967ms
704:    learn: 0.2365143        total: 2.3s     remaining: 964ms
705:    learn: 0.2364809        total: 2.31s    remaining: 960ms
706:    learn: 0.2361314        total: 2.31s    remaining: 957ms
707:    learn: 0.2361013        total: 2.31s    remaining: 954ms
708:    learn: 0.2359751        total: 2.31s    remaining: 950ms
709:    learn: 0.2357228        total: 2.32s    remaining: 947ms
710:    learn: 0.2356127        total: 2.32s    remaining: 943ms
711:    learn: 0.2354883        total: 2.32s    remaining: 940ms
712:    learn: 0.2352878        total: 2.33s    remaining: 937ms
713:    learn: 0.2350192        total: 2.33s    remaining: 933ms
714:    learn: 0.2347870        total: 2.33s    remaining: 930ms
715:    learn: 0.2344491        total: 2.34s    remaining: 927ms
716:    learn: 0.2342354        total: 2.34s    remaining: 923ms
717:    learn: 0.2339609        total: 2.34s    remaining: 920ms
718:    learn: 0.2336749        total: 2.35s    remaining: 917ms
719:    learn: 0.2334609        total: 2.35s    remaining: 913ms
720:    learn: 0.2331049        total: 2.35s    remaining: 910ms
721:    learn: 0.2330653        total: 2.35s    remaining: 907ms
722:    learn: 0.2329520        total: 2.36s    remaining: 903ms
723:    learn: 0.2327381        total: 2.36s    remaining: 900ms
724:    learn: 0.2325753        total: 2.36s    remaining: 897ms
725:    learn: 0.2325547        total: 2.37s    remaining: 894ms
726:    learn: 0.2323533        total: 2.37s    remaining: 891ms
727:    learn: 0.2321250        total: 2.38s    remaining: 889ms
728:    learn: 0.2320733        total: 2.39s    remaining: 888ms
729:    learn: 0.2319031        total: 2.39s    remaining: 885ms
730:    learn: 0.2317457        total: 2.39s    remaining: 881ms
731:    learn: 0.2315824        total: 2.4s     remaining: 878ms
732:    learn: 0.2314015        total: 2.4s     remaining: 875ms
733:    learn: 0.2311594        total: 2.4s     remaining: 871ms
734:    learn: 0.2309234        total: 2.41s    remaining: 868ms
735:    learn: 0.2305537        total: 2.41s    remaining: 865ms
736:    learn: 0.2303958        total: 2.41s    remaining: 861ms
737:    learn: 0.2302361        total: 2.42s    remaining: 858ms
738:    learn: 0.2302151        total: 2.42s    remaining: 855ms
739:    learn: 0.2299779        total: 2.42s    remaining: 852ms
740:    learn: 0.2296686        total: 2.43s    remaining: 848ms
741:    learn: 0.2295727        total: 2.43s    remaining: 845ms
742:    learn: 0.2293869        total: 2.43s    remaining: 842ms
743:    learn: 0.2291315        total: 2.44s    remaining: 839ms
744:    learn: 0.2291186        total: 2.44s    remaining: 835ms
745:    learn: 0.2289159        total: 2.44s    remaining: 832ms
```

```
746:    learn: 0.2287273        total: 2.45s    remaining: 829ms
747:    learn: 0.2285158        total: 2.45s    remaining: 826ms
748:    learn: 0.2283466        total: 2.45s    remaining: 822ms
749:    learn: 0.2280649        total: 2.46s    remaining: 819ms
750:    learn: 0.2279671        total: 2.46s    remaining: 816ms
751:    learn: 0.2277113        total: 2.46s    remaining: 813ms
752:    learn: 0.2276812        total: 2.47s    remaining: 809ms
753:    learn: 0.2274502        total: 2.47s    remaining: 806ms
754:    learn: 0.2274332        total: 2.47s    remaining: 803ms
755:    learn: 0.2272300        total: 2.48s    remaining: 799ms
756:    learn: 0.2270049        total: 2.48s    remaining: 796ms
757:    learn: 0.2268756        total: 2.48s    remaining: 793ms
758:    learn: 0.2266189        total: 2.49s    remaining: 790ms
759:    learn: 0.2264619        total: 2.49s    remaining: 787ms
760:    learn: 0.2261689        total: 2.5s     remaining: 784ms
761:    learn: 0.2258564        total: 2.5s     remaining: 780ms
762:    learn: 0.2258510        total: 2.5s     remaining: 777ms
763:    learn: 0.2255870        total: 2.5s     remaining: 774ms
764:    learn: 0.2252912        total: 2.51s    remaining: 771ms
765:    learn: 0.2250597        total: 2.51s    remaining: 768ms
766:    learn: 0.2248763        total: 2.52s    remaining: 764ms
767:    learn: 0.2246962        total: 2.52s    remaining: 761ms
768:    learn: 0.2245111        total: 2.52s    remaining: 758ms
769:    learn: 0.2244940        total: 2.53s    remaining: 755ms
770:    learn: 0.2244336        total: 2.53s    remaining: 751ms
771:    learn: 0.2243562        total: 2.53s    remaining: 748ms
772:    learn: 0.2243276        total: 2.54s    remaining: 745ms
773:    learn: 0.2241223        total: 2.54s    remaining: 742ms
774:    learn: 0.2240226        total: 2.54s    remaining: 738ms
775:    learn: 0.2238133        total: 2.55s    remaining: 735ms
776:    learn: 0.2236289        total: 2.55s    remaining: 732ms
777:    learn: 0.2233804        total: 2.55s    remaining: 729ms
778:    learn: 0.2233593        total: 2.56s    remaining: 725ms
779:    learn: 0.2231431        total: 2.56s    remaining: 722ms
780:    learn: 0.2230672        total: 2.56s    remaining: 719ms
781:    learn: 0.2228923        total: 2.57s    remaining: 715ms
782:    learn: 0.2226930        total: 2.57s    remaining: 712ms
783:    learn: 0.2225133        total: 2.57s    remaining: 709ms
784:    learn: 0.2224316        total: 2.58s    remaining: 707ms
785:    learn: 0.2224086        total: 2.58s    remaining: 704ms
786:    learn: 0.2223744        total: 2.59s    remaining: 700ms
787:    learn: 0.2223553        total: 2.59s    remaining: 697ms
788:    learn: 0.2222261        total: 2.6s     remaining: 695ms
789:    learn: 0.2221863        total: 2.6s     remaining: 691ms
790:    learn: 0.2221641        total: 2.6s     remaining: 688ms
791:    learn: 0.2218757        total: 2.61s    remaining: 685ms
792:    learn: 0.2218515        total: 2.61s    remaining: 682ms
793:    learn: 0.2216957        total: 2.61s    remaining: 678ms
```

```
794:    learn: 0.2215699        total: 2.62s    remaining: 675ms
795:    learn: 0.2213731        total: 2.62s    remaining: 672ms
796:    learn: 0.2210375        total: 2.62s    remaining: 668ms
797:    learn: 0.2209049        total: 2.63s    remaining: 665ms
798:    learn: 0.2208752        total: 2.63s    remaining: 662ms
799:    learn: 0.2206877        total: 2.63s    remaining: 658ms
800:    learn: 0.2206830        total: 2.64s    remaining: 655ms
801:    learn: 0.2205898        total: 2.64s    remaining: 652ms
802:    learn: 0.2204481        total: 2.64s    remaining: 648ms
803:    learn: 0.2202546        total: 2.65s    remaining: 645ms
804:    learn: 0.2201267        total: 2.65s    remaining: 642ms
805:    learn: 0.2198908        total: 2.65s    remaining: 638ms
806:    learn: 0.2198332        total: 2.65s    remaining: 635ms
807:    learn: 0.2196216        total: 2.66s    remaining: 632ms
808:    learn: 0.2195279        total: 2.66s    remaining: 628ms
809:    learn: 0.2194200        total: 2.66s    remaining: 625ms
810:    learn: 0.2193920        total: 2.67s    remaining: 622ms
811:    learn: 0.2191773        total: 2.67s    remaining: 618ms
812:    learn: 0.2191521        total: 2.67s    remaining: 615ms
813:    learn: 0.2191302        total: 2.68s    remaining: 612ms
814:    learn: 0.2190502        total: 2.68s    remaining: 608ms
815:    learn: 0.2188611        total: 2.68s    remaining: 605ms
816:    learn: 0.2188506        total: 2.69s    remaining: 602ms
817:    learn: 0.2188302        total: 2.69s    remaining: 598ms
818:    learn: 0.2185721        total: 2.69s    remaining: 595ms
819:    learn: 0.2184899        total: 2.69s    remaining: 592ms
820:    learn: 0.2184632        total: 2.7s     remaining: 588ms
821:    learn: 0.2181852        total: 2.7s     remaining: 585ms
822:    learn: 0.2181013        total: 2.7s     remaining: 582ms
823:    learn: 0.2179502        total: 2.71s    remaining: 578ms
824:    learn: 0.2178021        total: 2.71s    remaining: 575ms
825:    learn: 0.2175732        total: 2.71s    remaining: 572ms
826:    learn: 0.2173131        total: 2.72s    remaining: 568ms
827:    learn: 0.2171245        total: 2.72s    remaining: 565ms
828:    learn: 0.2169231        total: 2.72s    remaining: 562ms
829:    learn: 0.2167522        total: 2.73s    remaining: 558ms
830:    learn: 0.2166152        total: 2.73s    remaining: 555ms
831:    learn: 0.2164667        total: 2.73s    remaining: 552ms
832:    learn: 0.2162557        total: 2.73s    remaining: 548ms
833:    learn: 0.2161076        total: 2.74s    remaining: 545ms
834:    learn: 0.2160940        total: 2.74s    remaining: 542ms
835:    learn: 0.2159157        total: 2.74s    remaining: 538ms
836:    learn: 0.2157140        total: 2.75s    remaining: 535ms
837:    learn: 0.2155108        total: 2.75s    remaining: 532ms
838:    learn: 0.2153985        total: 2.75s    remaining: 529ms
839:    learn: 0.2153172        total: 2.76s    remaining: 525ms
840:    learn: 0.2152829        total: 2.76s    remaining: 522ms
841:    learn: 0.2152627        total: 2.76s    remaining: 519ms
```

```
842:    learn: 0.2151444        total: 2.77s    remaining: 515ms
843:    learn: 0.2149944        total: 2.77s    remaining: 512ms
844:    learn: 0.2148469        total: 2.77s    remaining: 509ms
845:    learn: 0.2147945        total: 2.78s    remaining: 506ms
846:    learn: 0.2145770        total: 2.78s    remaining: 503ms
847:    learn: 0.2144190        total: 2.79s    remaining: 499ms
848:    learn: 0.2141830        total: 2.79s    remaining: 496ms
849:    learn: 0.2140389        total: 2.79s    remaining: 493ms
850:    learn: 0.2138342        total: 2.79s    remaining: 490ms
851:    learn: 0.2138287        total: 2.8s     remaining: 486ms
852:    learn: 0.2137599        total: 2.8s     remaining: 483ms
853:    learn: 0.2137347        total: 2.81s    remaining: 480ms
854:    learn: 0.2136706        total: 2.81s    remaining: 476ms
855:    learn: 0.2134925        total: 2.81s    remaining: 473ms
856:    learn: 0.2134463        total: 2.81s    remaining: 470ms
857:    learn: 0.2132111        total: 2.82s    remaining: 467ms
858:    learn: 0.2129903        total: 2.82s    remaining: 463ms
859:    learn: 0.2129682        total: 2.83s    remaining: 460ms
860:    learn: 0.2127533        total: 2.83s    remaining: 457ms
861:    learn: 0.2126173        total: 2.83s    remaining: 453ms
862:    learn: 0.2124956        total: 2.83s    remaining: 450ms
863:    learn: 0.2124156        total: 2.84s    remaining: 447ms
864:    learn: 0.2122690        total: 2.84s    remaining: 444ms
865:    learn: 0.2121548        total: 2.85s    remaining: 440ms
866:    learn: 0.2121478        total: 2.85s    remaining: 437ms
867:    learn: 0.2118770        total: 2.85s    remaining: 434ms
868:    learn: 0.2117727        total: 2.85s    remaining: 430ms
869:    learn: 0.2114121        total: 2.86s    remaining: 427ms
870:    learn: 0.2114070        total: 2.86s    remaining: 424ms
871:    learn: 0.2113839        total: 2.86s    remaining: 421ms
872:    learn: 0.2112237        total: 2.87s    remaining: 417ms
873:    learn: 0.2110426        total: 2.87s    remaining: 414ms
874:    learn: 0.2109114        total: 2.88s    remaining: 411ms
875:    learn: 0.2107116        total: 2.88s    remaining: 407ms
876:    learn: 0.2105437        total: 2.88s    remaining: 404ms
877:    learn: 0.2104674        total: 2.88s    remaining: 401ms
878:    learn: 0.2103248        total: 2.89s    remaining: 398ms
879:    learn: 0.2101225        total: 2.89s    remaining: 394ms
880:    learn: 0.2100992        total: 2.89s    remaining: 391ms
881:    learn: 0.2099238        total: 2.9s     remaining: 388ms
882:    learn: 0.2097897        total: 2.9s     remaining: 385ms
883:    learn: 0.2096107        total: 2.91s    remaining: 381ms
884:    learn: 0.2093927        total: 2.91s    remaining: 378ms
885:    learn: 0.2093653        total: 2.91s    remaining: 375ms
886:    learn: 0.2093228        total: 2.92s    remaining: 372ms
887:    learn: 0.2090440        total: 2.92s    remaining: 368ms
888:    learn: 0.2087804        total: 2.92s    remaining: 365ms
889:    learn: 0.2085017        total: 2.93s    remaining: 362ms
```

```
890:    learn: 0.2082189        total: 2.93s    remaining: 358ms
891:    learn: 0.2080056        total: 2.93s    remaining: 355ms
892:    learn: 0.2078584        total: 2.94s    remaining: 352ms
893:    learn: 0.2078423        total: 2.94s    remaining: 348ms
894:    learn: 0.2077673        total: 2.94s    remaining: 345ms
895:    learn: 0.2076113        total: 2.95s    remaining: 342ms
896:    learn: 0.2076074        total: 2.95s    remaining: 339ms
897:    learn: 0.2074773        total: 2.96s    remaining: 336ms
898:    learn: 0.2074728        total: 2.97s    remaining: 333ms
899:    learn: 0.2074422        total: 2.97s    remaining: 330ms
900:    learn: 0.2073597        total: 2.97s    remaining: 327ms
901:    learn: 0.2071327        total: 2.98s    remaining: 324ms
902:    learn: 0.2069530        total: 2.98s    remaining: 320ms
903:    learn: 0.2069298        total: 2.98s    remaining: 317ms
904:    learn: 0.2068240        total: 2.99s    remaining: 314ms
905:    learn: 0.2066047        total: 2.99s    remaining: 310ms
906:    learn: 0.2063650        total: 2.99s    remaining: 307ms
907:    learn: 0.2062281        total: 3s       remaining: 304ms
908:    learn: 0.2061563        total: 3s       remaining: 300ms
909:    learn: 0.2060436        total: 3s       remaining: 297ms
910:    learn: 0.2058605        total: 3s       remaining: 294ms
911:    learn: 0.2057574        total: 3.01s    remaining: 290ms
912:    learn: 0.2057538        total: 3.01s    remaining: 287ms
913:    learn: 0.2057352        total: 3.02s    remaining: 284ms
914:    learn: 0.2057106        total: 3.02s    remaining: 280ms
915:    learn: 0.2055290        total: 3.02s    remaining: 277ms
916:    learn: 0.2053828        total: 3.02s    remaining: 274ms
917:    learn: 0.2052177        total: 3.03s    remaining: 270ms
918:    learn: 0.2050193        total: 3.03s    remaining: 267ms
919:    learn: 0.2050148        total: 3.03s    remaining: 264ms
920:    learn: 0.2048049        total: 3.04s    remaining: 260ms
921:    learn: 0.2046159        total: 3.04s    remaining: 257ms
922:    learn: 0.2045282        total: 3.04s    remaining: 254ms
923:    learn: 0.2042743        total: 3.04s    remaining: 251ms
924:    learn: 0.2042326        total: 3.05s    remaining: 247ms
925:    learn: 0.2040443        total: 3.05s    remaining: 244ms
926:    learn: 0.2039872        total: 3.06s    remaining: 241ms
927:    learn: 0.2039243        total: 3.06s    remaining: 237ms
928:    learn: 0.2038848        total: 3.06s    remaining: 234ms
929:    learn: 0.2036763        total: 3.06s    remaining: 231ms
930:    learn: 0.2035486        total: 3.07s    remaining: 227ms
931:    learn: 0.2035351        total: 3.07s    remaining: 224ms
932:    learn: 0.2033629        total: 3.07s    remaining: 221ms
933:    learn: 0.2033588        total: 3.08s    remaining: 217ms
934:    learn: 0.2032377        total: 3.08s    remaining: 214ms
935:    learn: 0.2030574        total: 3.08s    remaining: 211ms
936:    learn: 0.2029188        total: 3.09s    remaining: 208ms
937:    learn: 0.2028691        total: 3.09s    remaining: 204ms
```

```
938:    learn: 0.2027381        total: 3.09s    remaining: 201ms
939:    learn: 0.2025531        total: 3.1s     remaining: 198ms
940:    learn: 0.2024457        total: 3.1s     remaining: 194ms
941:    learn: 0.2024235        total: 3.1s     remaining: 191ms
942:    learn: 0.2022799        total: 3.11s    remaining: 188ms
943:    learn: 0.2020831        total: 3.11s    remaining: 184ms
944:    learn: 0.2019041        total: 3.11s    remaining: 181ms
945:    learn: 0.2018987        total: 3.12s    remaining: 178ms
946:    learn: 0.2017013        total: 3.12s    remaining: 175ms
947:    learn: 0.2016500        total: 3.12s    remaining: 171ms
948:    learn: 0.2013254        total: 3.12s    remaining: 168ms
949:    learn: 0.2011336        total: 3.13s    remaining: 165ms
950:    learn: 0.2010283        total: 3.13s    remaining: 161ms
951:    learn: 0.2009113        total: 3.13s    remaining: 158ms
952:    learn: 0.2007571        total: 3.14s    remaining: 155ms
953:    learn: 0.2005603        total: 3.14s    remaining: 151ms
954:    learn: 0.2003819        total: 3.14s    remaining: 148ms
955:    learn: 0.2003001        total: 3.15s    remaining: 145ms
956:    learn: 0.2001874        total: 3.15s    remaining: 142ms
957:    learn: 0.2000581        total: 3.15s    remaining: 138ms
958:    learn: 0.1999976        total: 3.16s    remaining: 135ms
959:    learn: 0.1997268        total: 3.16s    remaining: 132ms
960:    learn: 0.1995515        total: 3.17s    remaining: 128ms
961:    learn: 0.1995203        total: 3.17s    remaining: 125ms
962:    learn: 0.1994345        total: 3.17s    remaining: 122ms
963:    learn: 0.1994283        total: 3.18s    remaining: 119ms
964:    learn: 0.1994223        total: 3.18s    remaining: 115ms
965:    learn: 0.1992861        total: 3.19s    remaining: 112ms
966:    learn: 0.1992562        total: 3.19s    remaining: 109ms
967:    learn: 0.1992527        total: 3.19s    remaining: 106ms
968:    learn: 0.1991132        total: 3.19s    remaining: 102ms
969:    learn: 0.1990293        total: 3.2s     remaining: 98.9ms
970:    learn: 0.1988213        total: 3.2s     remaining: 95.6ms
971:    learn: 0.1987153        total: 3.2s     remaining: 92.3ms
972:    learn: 0.1986020        total: 3.21s    remaining: 89ms
973:    learn: 0.1985979        total: 3.21s    remaining: 85.7ms
974:    learn: 0.1984297        total: 3.21s    remaining: 82.4ms
975:    learn: 0.1982741        total: 3.22s    remaining: 79.1ms
976:    learn: 0.1981735        total: 3.22s    remaining: 75.8ms
977:    learn: 0.1980436        total: 3.22s    remaining: 72.5ms
978:    learn: 0.1978917        total: 3.23s    remaining: 69.2ms
979:    learn: 0.1978571        total: 3.23s    remaining: 65.9ms
980:    learn: 0.1977262        total: 3.23s    remaining: 62.6ms
981:    learn: 0.1976122        total: 3.23s    remaining: 59.3ms
982:    learn: 0.1974288        total: 3.24s    remaining: 56ms
983:    learn: 0.1972608        total: 3.24s    remaining: 52.7ms
984:    learn: 0.1972160        total: 3.24s    remaining: 49.4ms
985:    learn: 0.1969890        total: 3.25s    remaining: 46.1ms
```

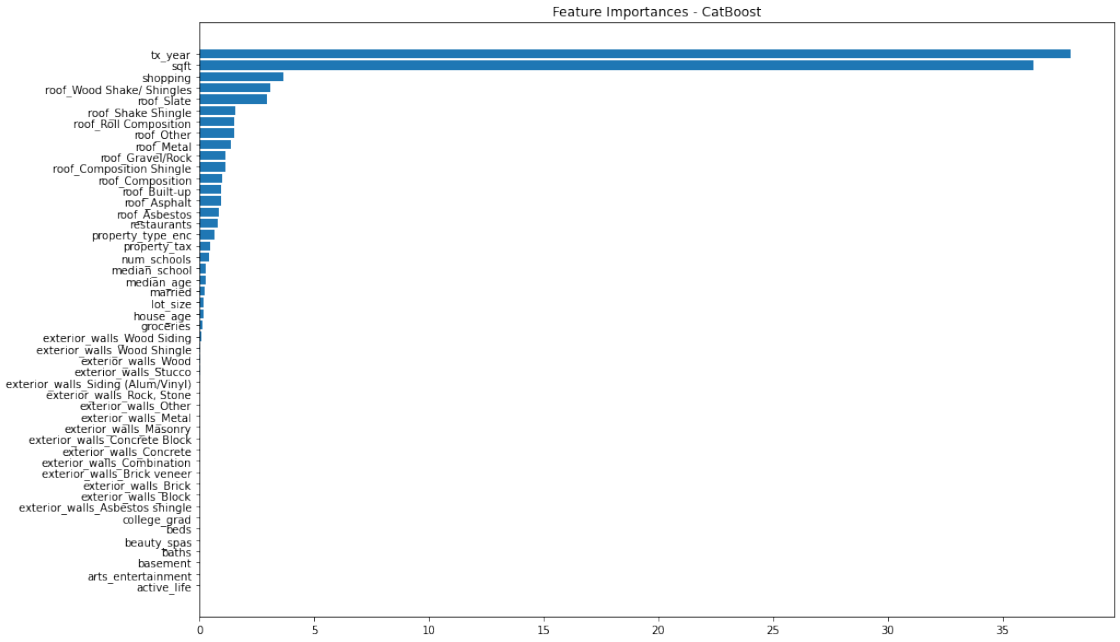
```
986:    learn: 0.1968416        total: 3.25s    remaining: 42.8ms
987:    learn: 0.1966424        total: 3.25s    remaining: 39.5ms
988:    learn: 0.1964443        total: 3.26s    remaining: 36.2ms
989:    learn: 0.1963107        total: 3.26s    remaining: 32.9ms
990:    learn: 0.1961886        total: 3.26s    remaining: 29.6ms
991:    learn: 0.1959819        total: 3.27s    remaining: 26.3ms
992:    learn: 0.1957636        total: 3.27s    remaining: 23ms
993:    learn: 0.1955199        total: 3.27s    remaining: 19.8ms
994:    learn: 0.1955138        total: 3.27s    remaining: 16.5ms
995:    learn: 0.1954326        total: 3.28s    remaining: 13.2ms
996:    learn: 0.1952748        total: 3.28s    remaining: 9.88ms
997:    learn: 0.1951148        total: 3.28s    remaining: 6.58ms
998:    learn: 0.1949578        total: 3.29s    remaining: 3.29ms
999:    learn: 0.1947785        total: 3.29s    remaining: 0us
Best Params: {'max_depth': 7, 'learning_rate': 0.03, 'l2_leaf_reg': 5}
```

[ ]:
```
cat_y_train_pred = rand_catr.best_estimator_.predict(df_X_train)

print("\n========Evaluation on Train Set========\n")
print("R2 score on train set : ",r2_score(df_y_train, cat_y_train_pred))
print("MSE on train set : ", mean_squared_error(df_y_train, cat_y_train_pred))
print("MAE on train set : ", mean_absolute_error(df_y_train, cat_y_train_pred))

cat_y_test_pred = rand_catr.best_estimator_.predict(df_X_test)

print("\n========Evaluation on Test Set========\n")
print("R2 score on test set : ",r2_score(df_y_test, cat_y_test_pred))
print("MSE on test set : ", mean_squared_error(df_y_test, cat_y_test_pred))
print("MAE on test set : ", mean_absolute_error(df_y_test, cat_y_test_pred))
```

```
========Evaluation on Train Set========

R2 score on train set :  0.9620613487120456
MSE on train set :  0.0379386512879544
MAE on train set :  0.14938293271083736

========Evaluation on Test Set========

R2 score on test set :  0.7933381530590226
MSE on test set :  0.20870306868635244
MAE on test set :  0.330770470417155
```

[ ]:
```
plt.figure(figsize = (15,10))
plt.barh(sorted(X_train.columns), sorted(rand_catr.best_estimator_.
 ↪feature_importances_))
plt.title('Feature Importances - CatBoost')
```

[ ]: Text(0.5, 1.0, 'Feature Importances - CatBoost')



Support Vector Regression (Yunze)

[ ]:
```
from sklearn.svm import SVR

Cs = [1, 10, 100, 1000, 10000]
gammas = [0.001, 0.01, 0.1, 'auto']

print("\n========Model Training on Different Hyperparameters========\n")

cross_val_scores = []
hyper_params = []
for C in Cs:
    for gamma in gammas:
        model = SVR(C=C, gamma=gamma, kernel='rbf')
        scores = cross_val_score(model, X_train, y_train, cv=5,␣
 ↪error_score="raise")
        cross_val_scores.append(np.mean(scores))
        print("penalty (C):", C, " gamma:", gamma, " score:", np.mean(scores))
        hyper_params.append([C, gamma])

best_C, best_gamma = hyper_params[np.argmax(cross_val_scores)]

print("\n========Best Hyper Parameters========\n")
print("best_C:", best_C, " best_gamma:", gamma)
```

```
model = SVR(C=best_C, gamma=best_gamma, kernel='rbf')
model.fit(X_train, y_train)

y_pred = model.predict(X_train)

print("\n========Evaluation on Train Set========\n")
print("R2 score on train set : ",r2_score(y_train, y_pred))
print("MSE on train set : ", mean_squared_error(y_train, y_pred))
print("MAE on train set : ", mean_absolute_error(y_train, y_pred))

y_pred = model.predict(X_test)

print("\n========Evaluation on Test Set========\n")
print("R2 score on test set : ",r2_score(y_test, y_pred))
print("MSE on test set : ", mean_squared_error(y_test, y_pred))
print("MAE on test set : ", mean_absolute_error(y_test, y_pred))
```

```
========Model Training on Different Hyperparameters========

penalty (C): 1  gamma: 0.001  score: 0.5014672875946322
penalty (C): 1  gamma: 0.01  score: 0.6579375097925888
penalty (C): 1  gamma: 0.1  score: 0.646044968190475
penalty (C): 1  gamma: auto  score: 0.6816185147825985
penalty (C): 10  gamma: 0.001  score: 0.5561993034732742
penalty (C): 10  gamma: 0.01  score: 0.6789694021389262
penalty (C): 10  gamma: 0.1  score: 0.6270296414608236
penalty (C): 10  gamma: auto  score: 0.6633503923128858
penalty (C): 100  gamma: 0.001  score: 0.5741902111269256
penalty (C): 100  gamma: 0.01  score: 0.6110162074867976
penalty (C): 100  gamma: 0.1  score: 0.6263593006977013
penalty (C): 100  gamma: auto  score: 0.5461828475370367
penalty (C): 1000  gamma: 0.001  score: 0.5956175876891612
penalty (C): 1000  gamma: 0.01  score: 0.3813214798074968
penalty (C): 1000  gamma: 0.1  score: 0.6263593006977013
penalty (C): 1000  gamma: auto  score: 0.4468200908296769
penalty (C): 10000  gamma: 0.001  score: 0.5418241273742306
penalty (C): 10000  gamma: 0.01  score: 0.18215068050132596
penalty (C): 10000  gamma: 0.1  score: 0.6263593006977013
penalty (C): 10000  gamma: auto  score: 0.4468200908296769

========Best Hyper Parameters========

best_C: 1  best_gamma: auto

========Evaluation on Train Set========
```

```
R2 score on train set :  0.7852103751809434
MSE on train set :  0.0269975905466919
MAE on train set :  0.1270983683940383

========Evaluation on Test Set========

R2 score on test set :  0.6549377583901819
MSE on test set :  0.04380035946436355
MAE on test set :  0.1619498619399685
```

Deep Learning Model (John)

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.layers import Dense
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam, RMSprop, SGD
from tensorflow.keras.layers import Dropout, BatchNormalization, LeakyReLU
from tensorflow.keras.losses import MeanAbsoluteError, MeanSquaredError,␣
↪MeanSquaredLogarithmicError
from sklearn.preprocessing import StandardScaler, RobustScaler
!pip install -q -U keras-tuner
import keras_tuner as kt
```

[ ]:
```
mae = MeanAbsoluteError()
mse = MeanSquaredError()
msle = MeanSquaredLogarithmicError()

scaler = StandardScaler()
scaler.fit(X_train)
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

[ ]:
```
def build_model(hp):
  model_opt = Sequential()

  hp_units1 = hp.Int('units1', min_value=48, max_value=528, step=24)
  hp_units2 = hp.Int('units2', min_value=48, max_value=528, step=24)
  hp_units3 = hp.Int('units3', min_value=48, max_value=528, step=24)

  model_opt.add(Dense(units=hp_units1, activation=hp.
↪Choice('dense_activation1', values=['relu', 'linear'])))
  model_opt.add(Dense(units=hp_units2, activation=hp.
↪Choice('dense_activation2', values=['relu', 'linear'])))
  model_opt.add(Dense(units=hp_units3, activation=hp.
↪Choice('dense_activation3', values=['relu', 'linear'])))
  model_opt.add(Dense(1))
  model_opt.compile(optimizer='adam',loss='mse',metrics=['mean_squared_error'])
```

```
  return model_opt

tuner = kt.
 ↪RandomSearch(build_model,objective='mean_squared_error',seed=42,max_trials=50,␣
 ↪overwrite=True)
tuner.search(X_train_scaled, y_train, epochs=10)
```

```
Trial 50 Complete [00h 00m 02s]
mean_squared_error: 0.6800670027732849

Best mean_squared_error So Far: 0.07778429985046387
Total elapsed time: 00h 03m 32s
INFO:tensorflow:Oracle triggered exit
```

[ ]:
```
for h_param in [f"units{i}" for i in range(1,4)] + [f"dense_activation{i}" for␣
 ↪i in range(1,4)]:
  print(h_param, tuner.get_best_hyperparameters()[0].get(h_param))
```

```
units1 312
units2 72
units3 504
dense_activation1 linear
dense_activation2 linear
dense_activation3 linear
```

[ ]:
```
best_model = tuner.get_best_models(num_models=1)[0]
loss, mse = best_model.evaluate(X_test_scaled, y_test)
```

```
12/12 [==============================] - 0s 3ms/step - loss: 0.0820 -
mean_squared_error: 0.0820
```

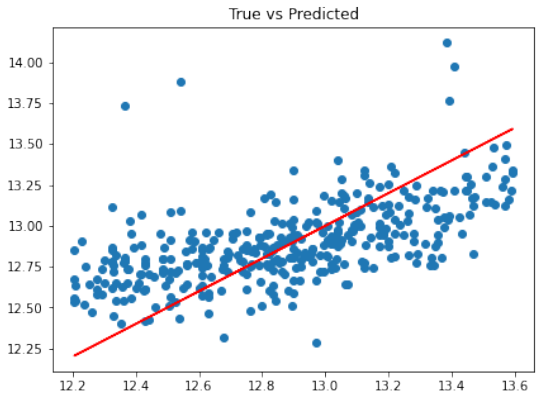
[ ]:
```
best_model.summary()
```

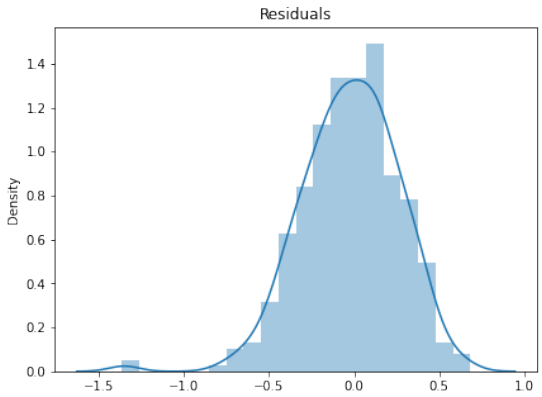
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 312)               15288

 dense_1 (Dense)             (None, 72)                22536

 dense_2 (Dense)             (None, 504)               36792

 dense_3 (Dense)             (None, 1)                 505

=================================================================
Total params: 75,121
Trainable params: 75,121
```

```
Non-trainable params: 0
_________________________________________________________________
```

[ ]:
```
y_pred = best_model.predict(X_test_scaled)
y_pred_train = best_model.predict(X_train_scaled)

print("\n========Evaluation on Train Set========\n")
print("R2 score on test set : ",r2_score(y_train, y_pred_train))
print("MSE on test set : ", mean_squared_error(y_train, y_pred_train))
print("MAE on test set : ", mean_absolute_error(y_train, y_pred_train))

print("\n========Evaluation on Test Set========\n")
print("R2 score on test set : ",r2_score(y_test, y_pred))
print("MSE on test set : ", mean_squared_error(y_test, y_pred))
print("MAE on test set : ", mean_absolute_error(y_test, y_pred))

fig, (ax1, ax2) = plt.subplots(1, 2)
fig.set_figheight(5)
fig.set_figwidth(15)

ax1.title.set_text('True vs Predicted')
ax1.scatter(y_test,y_pred)
ax1.plot(y_test,y_test,'r')

y = np.expand_dims(y_test, axis=1)
ax2 = sns.distplot(y- y_pred)
ax2.title.set_text("Residuals")
```

```
========Evaluation on Train Set========

R2 score on test set :  0.44107096825314585
MSE on test set :  0.07025356628129707
MAE on test set :  0.21254409341597352

========Evaluation on Test Set========

R2 score on test set :  0.3537604525748559
MSE on test set :  0.08203019937868378
MAE on test set :  0.22508319548596395

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```





[ ]:
```
#Baseline
model = Sequential()
model.add(Dense(48))
model.add(Dense(48))
model.add(Dense(48))
model.add(Dense(1))
```

[ ]:
```
#LeakyBaseline
model_lr = Sequential()
model_lr.add(Dense(48))
model_lr.add(LeakyReLU(alpha=0.05))
model_lr.add(Dense(48))
model_lr.add(LeakyReLU(alpha=0.05))
model_lr.add(Dense(48))
model_lr.add(LeakyReLU(alpha=0.05))
model_lr.add(Dense(48))
model_lr.add(LeakyReLU(alpha=0.05))
model_lr.add(Dense(48))
model_lr.add(Dense(1))
```

[ ]:
```
model.compile(optimizer="adam", loss='mse', metrics=['mean_squared_error'])
model_fitted = model.fit(X_train_scaled, y_train, batch_size=128, epochs=400,␣
 ↪validation_data=(X_test_scaled, y_test), verbose = 0)
model.evaluate(X_test, y_test)
```

```
12/12 [==============================] - 0s 4ms/step - loss: 221117.7031 -
mean_squared_error: 221117.7031
```

[ ]:
```
[221117.703125, 221117.703125]
```

[ ]:
```
model.summary()
```

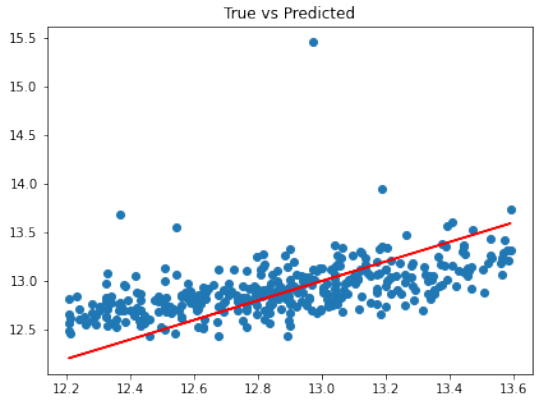
```
Model: "sequential_1"
_________________________________________________________________
```

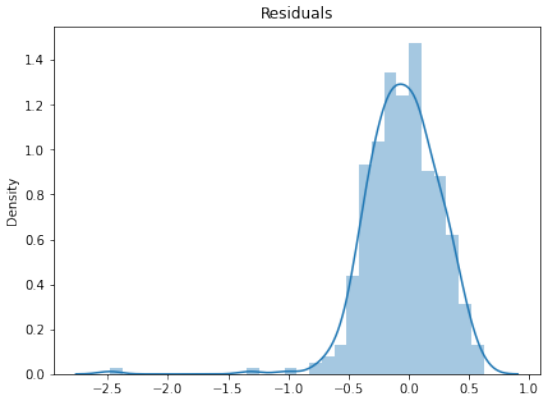
```
 Layer (type)                 Output Shape              Param #
=================================================================
 dense_4 (Dense)              (None, 48)                2352

 dense_5 (Dense)              (None, 48)                2352

 dense_6 (Dense)              (None, 48)                2352

 dense_7 (Dense)              (None, 1)                 49

=================================================================
Total params: 7,105
Trainable params: 7,105
Non-trainable params: 0
_________________________________________________________________
```

[ ]:
```python
y_pred = model.predict(X_test_scaled)
y_pred_train = best_model.predict(X_train_scaled)

print("\n========Evaluation on Train Set========\n")
print("R2 score on test set : ",r2_score(y_train, y_pred_train))
print("MSE on test set : ", mean_squared_error(y_train, y_pred_train))
print("MAE on test set : ", mean_absolute_error(y_train, y_pred_train))

print("\n========Evaluation on Test Set========\n")
print("R2 score on test set : ",r2_score(y_test, y_pred))
print("MSE on test set : ", mean_squared_error(y_test, y_pred))
print("MAE on test set : ", mean_absolute_error(y_test, y_pred))

fig, (ax1, ax2) = plt.subplots(1, 2)
fig.set_figheight(5)
fig.set_figwidth(15)

ax1.title.set_text('True vs Predicted')
ax1.scatter(y_test,y_pred)
ax1.plot(y_test,y_test,'r')

y = np.expand_dims(y_test, axis=1)
ax2 = sns.distplot(y- y_pred)
ax2.title.set_text("Residuals")
```

```
========Evaluation on Train Set========

R2 score on test set :  0.44107096825314585
MSE on test set :  0.07025356628129707
MAE on test set :  0.21254409341597352
```

```
========Evaluation on Test Set========

R2 score on test set :  0.23985796651576374
MSE on test set :  0.09648837309829407
MAE on test set :  0.23316967668629496

/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```





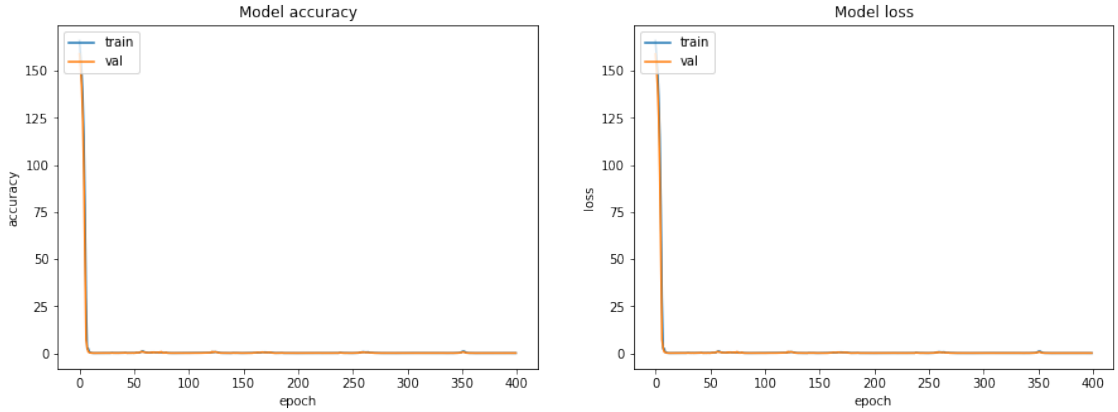
[ ]:
```
print(model_fitted.history.keys())
```

```
dict_keys(['loss', 'mean_squared_error', 'val_loss', 'val_mean_squared_error'])
```

[ ]:
```
fig, (ax1, ax2) = plt.subplots(1, 2)
fig.set_figheight(5)
fig.set_figwidth(15)
ax1.plot(model_fitted.history['mean_squared_error'])
ax1.plot(model_fitted.history['val_mean_squared_error'])
ax1.title.set_text('Model accuracy')
ax1.set_ylabel('accuracy')
ax1.set_xlabel('epoch')
ax1.legend(['train', 'val'], loc='upper left')

ax2.plot(model_fitted.history['loss'])
ax2.plot(model_fitted.history['val_loss'])
ax2.title.set_text('Model loss')
ax2.set_ylabel('loss')
ax2.set_xlabel('epoch')
ax2.legend(['train', 'val'], loc='upper left')
```

```
score = model.evaluate(X_test, y_test, verbose=0)
print("Test loss:", score[0])
```

```
Test loss: 221117.703125
```



[ ]: