

House_price_prediction (1)

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler, PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

df = pd.read_csv('../input/house-prices-advanced-regression-techniques/train.csv')
df.head()
df.columns
df.describe()
#Remove column ID
df.drop('Id', axis=1, inplace=True)
df.head()
df.rename(columns={'BedroomAbvGr':'Bedroom', 'KitchenAbvGr':'Kitchen'}, inplace=True)
df[['Bedroom', 'Kitchen']].head()
df['Bedroom'].value_counts()
df.isnull().sum()[df.isnull().sum() > 0]
df.drop(['MiscFeature', 'Alley', 'PoolQC', 'Fence'
df['Fireplaces'].value_counts()], axis=1, inplace=True)
df.drop(['FireplaceQu'], axis=1, inplace=True)
df['LotFrontage'].value_counts()
df.drop(['LotFrontage'], axis = 1, inplace = True)
df.isnull().sum()[df.isnull().sum()>0]
df['Electrical'].fillna(df['Electrical'].mode()[0], inplace=True)
df['Electrical'].isnull().sum()
df['MasVnrType'].value_counts()
df['MasVnrType'].fillna(df['MasVnrType'].mode()[0], inplace=True)
df.isnull().sum()[df.isnull().sum()>0]
df.corr()['SalePrice'].sort_values(ascending=False)
df[['Bedroom', 'SalePrice']].corr()
df[['Bedroom', 'SalePrice']].corr()
sns.regplot(x='Bedroom', y='SalePrice', data=df)
sns.regplot(x='Kitchen', y='SalePrice', data=df)
df[df['Kitchen']==0]
sns.regplot(x='OverallQual', y='SalePrice', data=df)
```

```

sns.regplot(x='GarageCars', y='SalePrice', data=df)
lm = LinearRegression()
features = ['OverallQual', 'GarageCars', 'GarageArea', 'TotalBsmntSF', '1stFlrSF', 'GrLivArea',
'FullBath', 'TotRmsAbvGrd', 'YearBuilt', 'YearRemodAdd']
lm.fit(df[features], df['SalePrice'])
lm.score(df[features], df['SalePrice'])
df[features].dtypes
from sklearn.impute import SimpleImputer
#pipeline
pipe = Pipeline(steps=[('scale', StandardScaler()),('preprocessor', SimpleImputer())
,('polynomial', PolynomialFeatures(include_bias=False)) ,('model', LinearRegression())])
pipe.fit(df[features], df['SalePrice'])
pipe.score(df[features], df['SalePrice'])
test_df = pd.read_csv('../input/house-prices-advanced-regression-techniques/test.csv')
test_df.head()
test_df[features].isnull().sum()[test_df[features].isnull().sum()>0]
test_df['GarageArea'].fillna(test_df['GarageArea'].mean(), inplace=True)
test_df['GarageCars'].fillna(test_df['GarageCars'].mean(), inplace=True)
test_df['TotalBsmntSF'].fillna(test_df['TotalBsmntSF'].mean(), inplace=True)
yhat = pipe.predict(test_df[features])
yhat[0:5]
'''submission = pd.DataFrame({'Id':test_df['Id'], 'SalePrice':yhat})
submission.to_csv('submission.csv', index=False)'''
from sklearn.neighbors import KNeighborsRegressor
knn = KNeighborsRegressor(n_neighbors=5)
knn.fit(df[features], df['SalePrice'])
knn.score(df[features], df['SalePrice'])
predictions = knn.predict(test_df[features])
'''submission = pd.DataFrame({'Id':test_df['Id'], 'SalePrice':predictions})
submission.to_csv('submission.csv', index=False)'''
submission =
pd.DataFrame({'Id':test_df['Id'], 'SalePrice':predictions})
submission.to_csv('submission.csv', index=False)'''
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(degree=2)
train_input_poly = poly.fit_transform(df[features])
poly.fit(train_input_poly, df['SalePrice'])
lm.fit(train_input_poly, df['SalePrice'])
predictions = lm.predict(poly.fit_transform(test_df[features]))
from sklearn.linear_model import Ridge
RidgeModel = Ridge(alpha=0.1)
RidgeModel.fit(train_input_poly, df['SalePrice'])
predictions = RidgeModel.predict(poly.fit_transform(test_df[features]))

```

```
submission = pd.DataFrame({'Id':test_df['Id'], 'SalePrice':predictions})  
submission.to_csv('submission.csv', index=False)
```

