

Question 2

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.linear_model import Ridge, Lasso
from sklearn.metrics import r2_score
import matplotlib.pyplot as plt

df=pd.read_csv("housing_price_dataset.csv")
df.head()

df.info()

df.isnull().sum()

df=df.drop_duplicates()

df.dropna()

numeric_df=df.select_dtypes(include=['int64','float64'])

y=numeric_df['Price']
x=numeric_df.drop('Price',axis=1)

x_train,x_test,y_train,y_test=train_test_split(x,y, test_size = 0.2,random_state=42)

ridge_params={'alpha':[0.1,1,10,50,100]}
ridge_cv=GridSearchCV(Ridge(),ridge_params,cv=5)
ridge_cv.fit(x_train,y_train)
ridge_pred=ridge_cv.best_estimator_.predict(x_test)

lasso_params={'alpha':[0.001,0.01,0.1,1,10]}
lasso_cv=GridSearchCV(Lasso(max_iter=5000),lasso_params,cv=5)
lasso_cv.fit(x_train,y_train)
lasso_pred=lasso_cv.best_estimator_.predict(x_test)

print("best ridge r2: ",r2_score(y_test,ridge_pred))
print("best lasso r2: ",r2_score(y_test,lasso_pred))

plt.figure(figsize=(8,6))
plt.scatter(y_test,ridge_pred,label="ridge prediction",alpha=0.6)
plt.scatter(y_test,lasso_pred,label="lasso prediction",alpha=0.6)
plt.plot([y_test.min(),y_test.max()], [y_test.min(),y_test.max()],linestyle="--")
plt.xlabel
plt.ylabel("actual price")
plt.legend("predicted price")
plt.title("actual vs predicted (ridge & less)")
plt.grid(True)
```

plt.show