**Q 1 Python**

import pandas as pd

def process\_data():

# Do not alter this line.

biopics = pd.read\_csv("biopics.csv", encoding='latin-1')

# Write your code here.

# drop duplicates

biopics = biopics.drop\_duplicates()

# rename variable

biopics = biopics.rename(columns={"box\_office" : "earnings"})

# Filter missing earnings

biopics = biopics[~biopics['earnings'].isna()]

# keep only 1990 or later

biopics = biopics[biopics['year\_release'] >= 1990]

biopics['type\_of\_subject'] = biopics['type\_of\_subject'].astype('category')

biopics['country'] = biopics['country'].astype('category')

# new variable

biopics.loc[biopics['lead\_actor\_actress'].isna() , 'lead\_actor\_actress\_known'] = False

biopics['lead\_actor\_actress\_known'] = biopics['lead\_actor\_actress\_known'].fillna(True)

biopics['earnings'] = round(biopics['earnings']/1000000, 4)

biopics = biopics[["title", "year\_release", "earnings", "country", "type\_of\_subject", "lead\_actor\_actress", "lead\_actor\_actress\_known"]]

biopics = biopics.sort\_values(by='earnings', ascending=False)

# Remember to return the right object.

return biopics.reset\_index(drop=True)

process\_data()

**Machine Learning**

import pandas as pd

data\_train = pd.read\_csv("data\_train.csv")

data\_test = pd.read\_csv("data\_test.csv")

import warnings

warnings.filterwarnings("ignore")

import pandas as pd

import numpy as np

from sklearn import linear\_model

def regression(data\_train, data\_test):

alpha = np.logspace(-4, -1, 4)

Y = data\_train['price'].values

X = data\_train.drop(columns = ['price'])

ridge\_model = linear\_model.RidgeCV(alphas = alpha, normalize=True)

ridge\_model.fit(X, Y)

ridge\_best\_alpha = ridge\_model.alpha\_

ridge\_model = linear\_model.Ridge(alpha = ridge\_best\_alpha, normalize=True)

ridge\_model.fit(X, Y)

ridge\_pred = ridge\_model.predict(data\_test.drop(columns=['price'])).round(2)

coef = ridge\_model.coef\_

columns = X.columns.values.reshape(-1, 1)

data = pd.DataFrame(data=columns, columns=['variable'])

data['coef'] = coef

ridge\_data = data[abs(data['coef']) > 0.001]

del data

lasso\_model = linear\_model.LassoCV(alphas = alpha, normalize=True)

lasso\_model.fit(X, Y)

lasso\_best\_alpha = lasso\_model.alpha\_

lasso\_model = linear\_model.Lasso(alpha = lasso\_best\_alpha, normalize=True)

lasso\_model.fit(X, Y)

lasso\_pred = lasso\_model.predict(data\_test.drop(columns=['price'])).round(2)

lasso\_coef = lasso\_model.coef\_

columns = X.columns.values.reshape(-1, 1)

data = pd.DataFrame(data=columns, columns=['variable'])

data['coef'] = lasso\_coef

lassoe\_data = data[data['coef'] != 0]

del data

l1\_ratio = np.arange(0.6, 1, 0.1)

elastic\_model = linear\_model.ElasticNetCV(alphas = alpha, normalize=True, l1\_ratio=l1\_ratio)

elastic\_model.fit(X, Y)

elastic\_best\_alpha = elastic\_model.alpha\_

elastic\_best\_l1ration = elastic\_model.l1\_ratio\_

elastic\_model = linear\_model.ElasticNet(alpha = elastic\_best\_alpha, normalize=True, l1\_ratio=elastic\_best\_l1ration)

elastic\_model.fit(X, Y)

elastic\_pred = elastic\_model.predict(data\_test.drop(columns=['price'])).round(2)

elastic\_coef = elastic\_model.coef\_

columns = X.columns.values.reshape(-1, 1)

data = pd.DataFrame(data=columns, columns=['variable'])

data['coef'] = elastic\_coef

elastic\_data = data[abs(data['coef']) > 0.001]

return {'ridge': {'alpha': ridge\_best\_alpha, 'pred': ridge\_pred, 'coefficients': ridge\_data},

'lasso': {'alpha': lasso\_best\_alpha, 'pred': lasso\_pred, 'coefficients': lassoe\_data},

'elastic\_net': {'alpha': elastic\_best\_alpha, 'l1\_ratio': elastic\_best\_l1ration, 'pred': elastic\_pred, 'coefficients': elastic\_data}

}

regression(data\_train, data\_test)

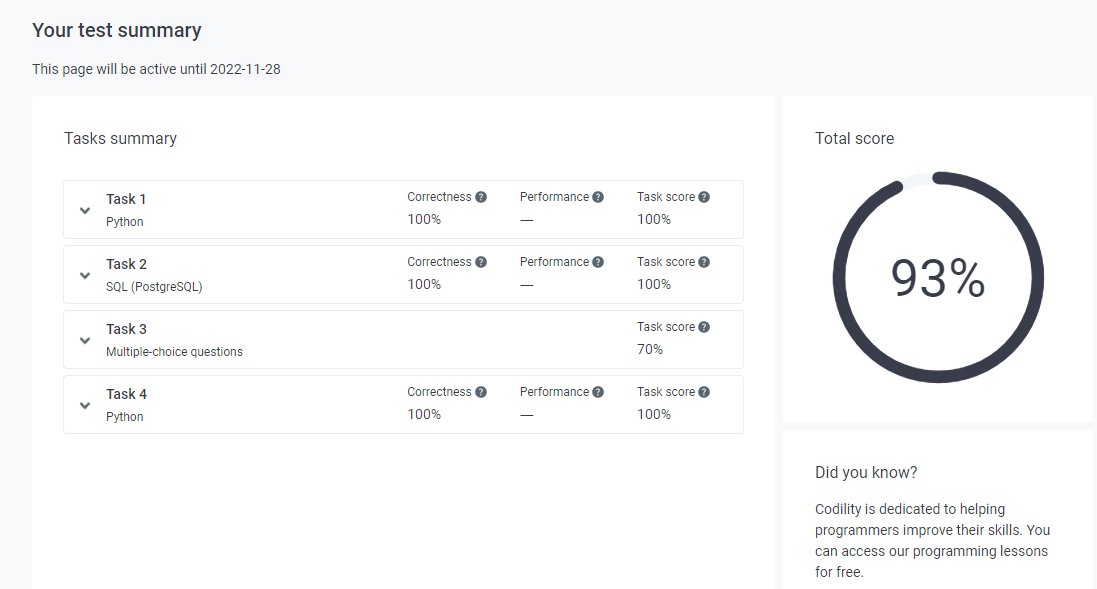
**SQL**

-- write your code in PostgreSQL 9.4

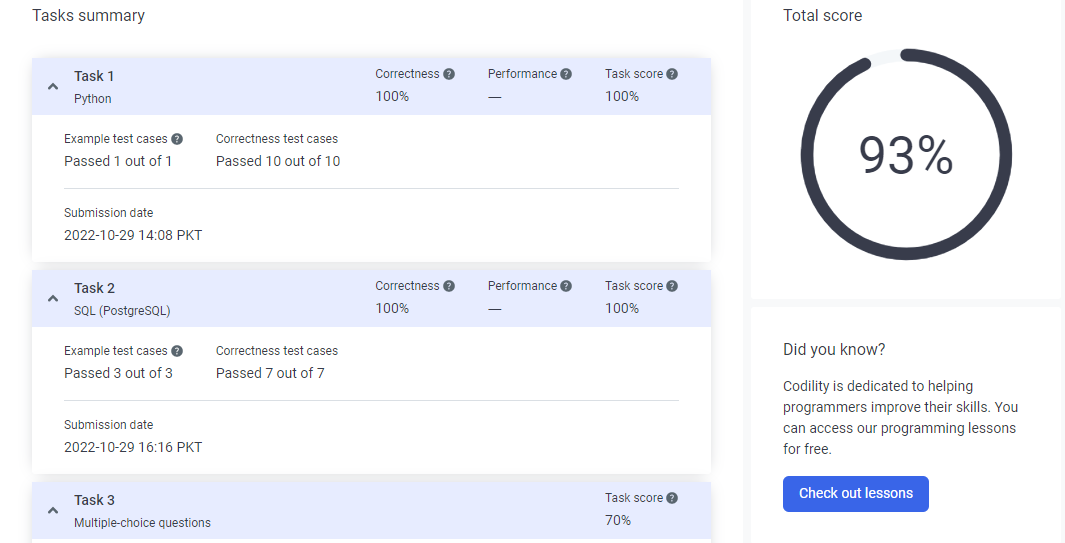
--SELECT user\_id from logs where login\_time > "2020-02-20 11:22:00";

SELECT user\_id from logs where login\_time > '2020-02-20 11:22:00' group by user\_id  having(count(\*)) >=1 order by user\_id;

**And 10 technical Machine Learning MCQS**

****

[**https://app.codility.com/c/feedback/J96SWM-T4E/**](https://app.codility.com/c/feedback/J96SWM-T4E/)

****