upGrad

Assessment Profile™

Candidate ID: **3888430**

Candidate Name: **Kaliraj Balakrishnan**

Date: **21-Mar-2021**

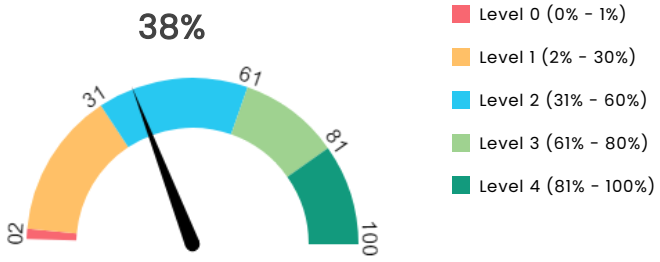
Assessment Name: **ML_01_ML1_Senior**

Client Name: **Upgrad**

SCORE ANALYSIS

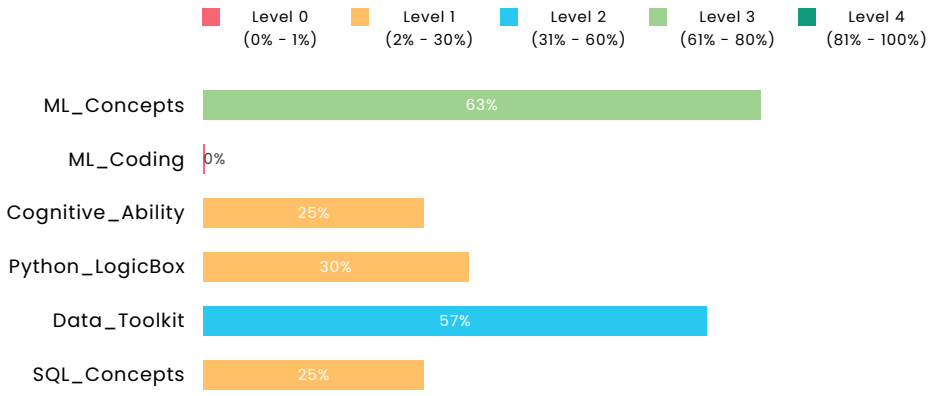
Score: **25/ 66**

Time Taken:
65 min 54 sec / 81 min



Kaliraj Balakrishnan scored **38%** and completed assessment in **81%** of the alloted time

SECTION SCORE ANALYSIS



Sections	Questions	Q Appeared	Correct Answer	Section Time	Time Taken	Score
ML_Concepts	16	16	10	17 min	11 min 51 sec	10/ 16
ML_Coding	1	1	0	15 min	15 min	0/ 10
Cognitive_Ability	12	12	3	12 min	11 min 6 sec	3/ 12
Python_LogicBox	2	2	1	15 min	12 min 4 sec	3/ 10
Data_Toolkit	14	14	8	18 min	12 min 34 sec	8/ 14
SQL_Concepts	4	4	1	4 min	3 min 17 sec	1/ 4

DIFFICULTY LEVEL ANALYSIS

Level	Number of Questions	Correct Attempts	Correctness

Level	Number of Questions	Correct Attempts	Correctness
Easy	19	7	36.84 %
Medium	18	12	66.67 %
Hard	12	4	33.33 %

PROCTORING ANALYSIS

Window Violation: **0**

Time Violation: **0 sec**

0 images captured. Image violation detected in 0 images

QUESTION DETAILS

Question: #1	Type: Coding	Skill: Machine Learning Coding	Status: Answered	Result: Wrong
Level: Easy	Time Taken: 15 min 2 sec	Score: 0 / 10	Window Violation: 0 times	Time Violation: 0 sec

Question #1

You are a data science professor.

You are teaching linear regression to your students.

As part of the lesson, you want to show the students the line with the least-squares fit.

Thus, given some values for the independent variable '***X***', and the corresponding dependent variable '***Y***', ***output the sum of the slope and the intercept of the linear regression fit, rounded off to the second decimal place.***

Note

Train the model on default hyperparameters.

Round off the output number to 2 digits after the decimal. Example: 2.33, 5.88

Input Format

Line 1: A space-separated list of floating-point numbers denoting the ***independent variables***.

Line 2: A space-separated list of floating-point numbers denoting the ***dependent variables.***

Output Format

Print a single floating point number denoting the sum of the slope and intercept of the fitted linear regression rounded up to the second decimal place.

Sample Input

```
-1.053347195170761 -0.07047909428095181 0.09683910047859905
-1.7673884778971813 -1.2470643567216502 -1.598828633884807
-1.470326613668779 0.06373640328726708 0.8313079349445928
0.810428363101334
-8.787747649931456 -0.4283737851271292 1.0784334837561802
-14.868094220695752 -9.860690691161116 -14.132472220335451
-13.185944016121688 3.7540726760233563 8.180097232403666 11.623121566237993
```

Sample Output

```
11.24
```

Explanation

The sum of the slope and the intercept of the linear regression fit rounded off to the second decimal place is ***11.24.***

Answer:

Coding Language: Python with ML

Candidate Code:

```
'''IMPORTANT: click on "Compile & Run" to generate an output to the written code
ONLY this will generate scores for your code!
Also this will help you to check correctness of your code against test cases'''

import numpy as np

# INPUT [uncomment & modify if required]
#input for independent variables
x = list(map(float, input().split()))

#input for dependent variables
y = list(map(float, input().split()))

m,c = np.polyfit(x,y,1);
#this is default OUTPUT. You are allowed to modify it.
result=0
size = len(x);
for i in range(0,size-1):
    result += (y[i]-(m*x[i]+c)) ** 2

#NOTE: The variable "x" contains list of independent variables and variable "y" contains list of dependent variables

#WRITE YOUR LOGIC HERE:



# OUTPUT [uncomment & modify if required]
print(format(result,'.2f'))
```

Compilation Summary:

Compilation Status: **Compile Successfully**	No Of Compilations: **6**

Defualt Input:

-1.053347195170761 -0.07047909428095181 0.0968391 9.48
0047859905 -1.7673884778971813 -1.24706435672165
02 -1.598828633884807 -1.470326613668779 0.06373
640328726708 0.8313079349445928 0.8104283631013
34
-8.787747649931456 -0.4283737851271292 1.07
84334837561802 -14.868094220695752 -9.86069069
1161116 -14.132472220335451 -13.185944016121688 3.75
40726760233563 8.180097232403666 11.623121566237
993

Candidate Output:

Test Case Summary:

Test Case: 1 Status: Fail Score:0

Test Case Input	Expected Output	Actual Output
-1.053347195170761 -0.07047909428095181 0.096 83910047859905 -1.7673884778971813 -1.247064 3567216502 -1.598828633884807 -1.47032661366 8779 0.06373640328726708 0.831307934944592 8 0.810428363101334 -8.787747649931456 -0.4283737851271292 1.0784 334837561802 -14.868094220695752 -9.860690 691161116 -14.132472220335451 -13.18594401612168 8 3.7540726760233563 8.180097232403666 11.62 3121566237993	11.24	9.48

Test Case: 2 Status: Fail Score:0

Test Case Input	Expected Output	Actual Output
-0.6462928194729174 1.5112420039891516 0.91552 70919790113 -0.3820530494472467 0.7123023785 388861 0.8775648991069698 1.38290586266108 2 -0.604871381953642 -0.8006928770429605 -0. 7204982871528259 -8.995515118674142 25.855562169066932 14.8449 9797104278 -2.5143299737862206 11.0592526873 74283 15.247973938374962 21.45398824691205 8 -7.114398260701733 -10.581363839097492 -11.51 8875613934384	16.63	9.77

Test Case: 3 Status: Fail Score:0

Test Case Input	Expected Output	Actual Output

Test Case Input	Expected Output	Actual Output
0.20076599207795404 -0.36597173416618545 0.11509494440343183 -1.7011514131966492 -0.7822901409936784 -0.2837055622647406 1.985715985505638 -0.3308663426558487 -0.10587206117721278 -0.42926503507535413 5.819207450683875 -3.476493792097122 -1.2662709189285595 -2.097499402204781 0.0049607913843068285 0.21525959653513693 17.986541769721853 1.6683158455592788 1.3727527106931485 -1.205243043763006	8.64	86.07

Test Case: **4** Status: **Fail** Score:**0**

Test Case Input	Expected Output	Actual Output
-0.5754432391087806 0.4684332567381314 -0.2107152388066264 1.0883807349571344 0.9971886718733006 1.3446395082373481 -2.1298971293183593 1.1289929362444253 0.7067615732992125 -1.433364558370002 -35.13644158837215 43.21440716831026 -14.45036549817872 82.97168563468632 75.32329288222742 100.07665745412869 -150.1521220889344 84.52175584458142 54.964219180648506 -101.90656032702331	76.37	49.26

Test Case: **5** Status: **Fail** Score:**0**

Test Case Input	Expected Output	Actual Output
-0.4160074130286227 0.6087003482876278 0.7442195446742945 -1.0098081275365054 -0.03403845911276064 0.8794794841101103 0.460117265273801 -0.7322152001681816 -0.38837438621798004 -0.18226962845813346 -24.300180054813847 32.725970021001146 41.901227594232346 -51.67089004925092 3.629362713503361 52.825993532876815 32.18556987087824 -43.52459351180278 -21.326959040444187 -1.3356919373799139	58.76	92.50

Test Case: **6** Status: **Fail** Score:**0**

Test Case Input	Expected Output	Actual Output

Test Case Input	Expected Output	Actual Output
0.5946041914990825 -0.7184336320243087 -0.04466439181020556 0.9966053681404196 -0.3248000524298263 -1.4252832666338595 -1.156176314065439 -0.09027141096077067 -0.1090052827670868 -0.6753238528196934 9.32630404749119 -8.660303417481598 0.9766230036260993 12.448790897780574 -3.4089722825904394 -16.437084531080185 -13.07157098806054 -0.7496544096406333 0.36927157988483406 -5.459929801090279	13.31	4.14

Question: #2 Type: AI-LogicBox Skill: Machine Learning Status: Answered Result: Correct

Level: Easy Time Taken: 7 min 39 sec Score: 3 / 5 Window Violation: 0 times Time Violation: 0 sec

Question #2

The scikit-learn library in Python is a well-known library used to solve modeling problems. You have the following dataset at hand:

Throttle Speed (m/s)	Throttle Acceleration (m/s^2)	Steering Angle (radians)
5	1.5	0
6.5	0.4	1.08
10	3.3	0.06
3	2.1	0.22

In relation to the modeling of the above dataset (Throttle Speed and Acceleration being the independent variables, Steering Angle being the dependent variable).

Complete the code as per the instructions given for each blank:

At Blank 1: Which paradigm of Machine Learning Problems does the above scenario belong to?

At Blank 2: Write the code to import the relevant libraries required to model the problem.

At Blank 3: Write the code to split the data into 80:20 train/test sets and a random state equal to 123.

At Blank 4: Write the code to fit a Linear Regression model on the training set.

Sample Script:

Name the ML Paradigm associated with the problem:

Blank 1: Write your answer here

```
import numpy as np
import pandas as pd
```

Blank 2: Write your code here

```
X = data.features
y = data.target
X_train, X_test, y_train, y_test =
```

Blank 3: Write your code here

```
lr = LinearRegression()
```

Blank 4: Write your code here

```
r2_Score =
```

Blank 5: Write your code here

At Blank 5: Write the code to calculate the R^2 score on the test set.

Answer:

Answer2 ✗ **Wrong** **Points:** 1 **Score:** 0

from sklearn.model_selection import train_test_split from sklearn.model_selection import r2_score

Answer3 ✓ **Correct** **Points:** 1 **Score:** 1

train_test_split(x,y)

Answer4 ✓ **Correct** **Points:** 1 **Score:** 1

lr.fit(X_train,y_train)

Answer5 ✓ **Correct** **Points:** 1 **Score:** 1

r2_score(lr.fit(X_test,y_test))

Question: #3 Type: AI-LogicBox Skill: Machine Learning Status: Answered Result: Wrong

Level: Medium Time Taken: 4 min 25 sec Score: 0 / 5 Window Violation: 0 times Time Violation: 0 sec

Question #3

The scikit-learn library in Python is a well-known library used to solve modelling problems. You have the following dataset at hand:

Transaction Region	Transaction Amount ($)	Transaction Status (1/0/2 for Malicious/Normal/Cannot Say)
APAC	334500	0
EU	23345	1
USA	345678	2
CN	23456	0

In relation to the modelling of the above dataset (Transaction Region and Amount being the independent variables, Transaction Status being the dependent variable), fill in the blanks below:

At Blank 1: Which loss is minimized when the above data is modelled using Logistic Regression?

At Blank 2: Write the code to import the relevant libraries required to model the problem using Logistic Regression.

At Blank 3: Write the code to convert the `Transaction Region` column into dummy variables using pandas.

At Blank 4: Write the code to fit a Logistic

Sample Script:

The loss that is minimized when the above data is modelled using Logistic Regression:

Blank 1: Write your answer h

```
import numpy as np
import pandas as pd
from sklearn.model_selection import
train_test_split
```

Blank 2: Write your code here

```
X = data.features
y = data.target
```

```
X_transformed =
```

Blank 3: Write your code here

```
clf = SVC()
```

Blank 4: Write your code here

```
r2_score=
```

Blank 5: Write your code here

Regression Classifier on the transformed set.

At Blank 5: Write the code to find the R2 score on the transformed set.

Answer:

Answer1 ✗ **Wrong** **Points:** 1 **Score:** 0

Loss

Answer2 ✗ **Wrong** **Points:** 1 **Score:** 0

from sklearn.model_selection import LogisticRegression

Answer3 ✗ **Wrong** **Points:** 1 **Score:** 0

pd.get_dummies('Transaction Region')

Answer4 ✗ **Wrong** **Points:** 1 **Score:** 0

clf.fit(X_transformed, y)

Answer5 ✗ **Wrong** **Points:** 1 **Score:** 0

r2_score(clf)