Lab Report

In lab Task 1:

```
!wget
https://www.dropbox.com/s/veak3ugc4wj9luz/Alumni%20Giving%20Regression%20%28Edited%29.csv?dl=1
```

In lab Task 2:

```
!ls
```

Code output:

```
'Alumni Giving Regression (Edited).csv?dl=1'   sample_data
```

In lab Task 3:

```
import pandas as pd

# Replace 'data.csv' with your file name
data = pd.read_csv('Alumni Giving Regression (Edited).csv?dl=1')
```

```
data.head()
```

Code output:

	A	B	C	D	E	F
0	24	0.42	0.16	0.59	0.81	0.08
1	19	0.49	0.04	0.37	0.69	0.11
2	18	0.24	0.17	0.66	0.87	0.31
3	8	0.74	0.00	0.81	0.88	0.11
4	8	0.95	0.00	0.86	0.92	0.28

In lab Task 4:

```
from keras.models import Sequential
from keras.layers import Dense, Dropout
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import numpy as np
from sklearn import linear_model
from sklearn import preprocessing
from sklearn import tree
from sklearn.ensemble import RandomForestRegressor
import pandas
```

```
data.describe()
```

Code output:

	A	B	C	D	E	F
count	123.000000	123.000000	123.000000	123.000000	123.000000	123.000000
mean	17.772358	0.403659	0.136260	0.645203	0.841138	0.141789
std	4.517385	0.133897	0.060101	0.169794	0.083942	0.080674
min	6.000000	0.140000	0.000000	0.260000	0.580000	0.020000
25%	16.000000	0.320000	0.095000	0.505000	0.780000	0.080000
50%	18.000000	0.380000	0.130000	0.640000	0.840000	0.130000
75%	20.000000	0.460000	0.180000	0.785000	0.910000	0.170000
max	31.000000	0.950000	0.310000	0.960000	0.980000	0.410000

```
corr=data.corr(method ='pearson')
corr
```

	A	B	C	D	E	F
A	1.000000	-0.691900	0.414978	-0.604574	-0.521985	-0.549244
B	-0.691900	1.000000	-0.581516	0.487248	0.376735	0.540427

C	0.414978	-0.581516	1.000000	0.017023	0.055766	-0.175102
D	-0.604574	0.487248	0.017023	1.000000	0.934396	0.681660
E	-0.521985	0.376735	0.055766	0.934396	1.000000	0.647625
F	-0.549244	0.540427	-0.175102	0.681660	0.647625	1.000000

In lab Task 6:

```
Y_POSITION = 5
model_1_features = [i for i in range(0,Y_POSITION)]
X = data.iloc[:,model_1_features]
Y = data.iloc[:,Y_POSITION]
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.20,
random_state=2020)
```

Explanantion

Task 1: Downloading the Dataset

I used the wget command to download the dataset from a Dropbox link. This was necessary to acquire the data I needed for my analysis and modeling. An alternative option would be to manually download the dataset from Dropbox to your local machine and then upload it to your Colab environment.

Task 2: Importing Libraries

I imported various libraries, including Keras, scikit-learn, and numpy. These libraries are essential for machine learning, data analysis, and building predictive models. An alternate option is to install the required libraries in your Colab environment using !pip install library_name if they are not already installed.

Task 3: Reading the Dataset

After downloading the dataset in Task 1, I used the pd.read_csv function to read it into a Pandas DataFrame. This step made the data easily accessible for my analysis and modeling. An alternate option would be to manually upload the dataset to your Colab environment and then load it using Pandas.

Task 4: Calculating Correlations

To understand the relationships between variables in the dataset, I calculated Pearson correlation coefficients using df.corr(method='pearson'). This helped me identify strong relationships between features. An alternative option is to use other correlation methods like Spearman or Kendall tau, depending on the nature of your data and the relationships you want to explore.

Task 5: Preparing Data for Regression

In this task, I prepared the data for regression analysis, including splitting it into training and testing sets. An alternative option would be to use different ratios for the training and testing sets or consider other methods like k-fold cross-validation for model evaluation.

Task 6: Training and Evaluating a Linear Regression Model

I created a linear regression model and assessed its performance using RMSE. An alternative option is to explore different regression algorithms such as Lasso, Ridge, or Decision Trees and compare their performance to select the best model for your specific dataset and problem.

Code output:

```
D:\Python37\python.exe D:\AI_Labs\FA21BCE084_AI_LAB\lab2\task4a.py
3.25
1
16
8

Process finished with exit code 0
```

Task4_b

```
evenNumbers = [2, 4, 6, 8]
oddNumbers = [1, 3, 5, 7]

allNumbers = oddNumbers + evenNumbers

print(allNumbers)
```

Code output:

```
D:\Python37\python.exe D:\AI_Labs\FA21BCE084_AI_LAB\lab2\task4b.py
[1, 3, 5, 7, 2, 4, 6, 8]

Process finished with exit code 0
```

Task4_c

```
x = object()
y = object()

x_list = [x,x,x,x,x,x,x,x,x,x]
y_list = [y,y,y,y,y,y,y,y,y,y]

concat_list = x_list + y_list

print("X List contains {} objects" .format(len(x_list)))
print("Y List contains {} objects" .format(len(y_list)))
print("Concat List contains {} objects" .format(len(concat_list)))
```

Code output:

```
D:\Python37\python.exe D:\AI_Labs\FA21BCE084_AI_LAB\lab2\task4c.py
X List contains 10 objects
Y List contains 10 objects
Concat List contains 20 objects

Process finished with exit code 0
```

In lab Task 5:

```
hello = "Hello"
world = "World"
print("{}, {}" .format(hello, world))

print("{} {} {} {} {}" .format(hello, hello, hello, hello, hello))
```

Code output:

```
D:\Python37\python.exe D:\AI_Labs\FA21BCE084_AI_LAB\lab2\task5.py
Hello, World
Hello Hello Hello Hello Hello

Process finished with exit code 0
```

In lab Task 6:

```
x = 3
print (x == 3)
print (x == 2)
print (x < 5)
name = "saira"
print(name == "saira" and x == 2)
print(name == "saira" or name == "John")

print(name in ["John", "Jill", "Jess"])
```

Code output:

```
D:\Python37\python.exe D:\AI_Labs\FA21BCE084_AI_LAB\lab2\task6.py
True
False
True
False
True
False

Process finished with exit code 0
```

In lab Task 7:

```
x = 2
y = 10

if x>2:
    print("x > 2")
elif x == 2 and y >50:
    print("x == 2 and y > 50")
elif x < 10 or y > 50:
    print("x < 10 or y > 50")
else:
    print("Nothing happened")

nameList1 = ["John", "Jill"]
nameList2 =["John", "Jill"]

print(not(nameList1 == nameList2))

print(nameList1 == nameList2)
print(nameList1 is nameList2)
```

Code output:

```
D:\Python37\python.exe D:\AI_Labs\FA21BCE084_AI_LAB\lab2\task7.py
x < 10 or y > 50
False
True
False

Process finished with exit code 0
```

In lab Task 8:

```
numbers = [10, 20, 30, 40, 50]

for number in numbers:
    result = number * 2
    print(result)

text = "Hello World"

for char in text:
    print(char)

newText = ""

for char in text:
    if char.isalpha():
        newText += char.upper()
    else:
        newText += char

print(newText)

numericData = []

for i in range(1, 11):
    numericData.append(i)
```

```
print(numericData)
```

Code output:

```
D:\Python37\python.exe D:\AI_Labs\FA21BCE084_AI_LAB\lab2\task8.py
20
40
60
80
100
H
e
l
l
o

W
o
r
l
d
HELLO WORLD
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Process finished with exit code 0
```

In lab Task 9:

```
count = 1
while count < 5:
    print(count)
    count += 1

text = "Hello"
index = 0
while index < len(text):
    print(text[index])
    index += 1

grades = {"saira": 92, "ali": 79, "Ahmad": 98}
keys = list(grades.keys())
index = 0
while index < len(keys):
    key = keys[index]
    value = grades[key]
    print(f"{key}: {value}")
    index += 1
```

Code output:

```
D:\Python37\python.exe D:\AI_Labs\FA21BCE084_AI_LAB\lab2\task9.py
1
2
3
4
H
e
l
l
o
saira: 92
ali: 79
Ahmad: 98

Process finished with exit code 0
```