Name:UdayKumar Gadikar PRN: 2019BTECS00090
Name:Koustubh Tatikondawar PRN: 2019BTECS00091
Class: TYCSE Batch: T8
SET Lab Assignment-2

Q. 1. Node

1. **Original author:** Ryan Dahl
2. **Developers:** OpenJs Foundation
3. **Initial release: 27 May 2009**
4. **Stable release:** 17.4.0
5. **Preview release:** 16.14.0
6. **Repository (with cloud support):** https://github.com/nodejs/node
7. **Written in (Languages):** C, C++, JavaScript
8. **Operating System support:** z/OS, Linux, macOS, Microsoft Windows, SmartOS, FreeBSD, OpenBSD, IBM AIX
9. **Platform, portability:** All Platforms and portable
10. **Available in (Total languages):** Javascript only
11. **List of languages supported:** Javascript only
12. **Type (Programming tool, integrated development environment etc.):** Javascript runtime environment
13. **Website:** https://nodejs.org/
14. **Features:**

A. Cross-platform compatibility.
B. The convenience of using one coding language.
C. V8 Engine.
D. Facilitates quick deployment and microservice development.
E. Scalable.
F. Commendable data processing ability.
G. Active open-source community.

H. Additional functionality of NPM.

15. **Size (in MB, GB etc.):** 436MB
16. **Privacy and Security:** The basis of Node is javascript so it is secure as compared to others but also it is susceptible to security exposures.
17. **Type of software (Open source/License):** Open source
18. **If License- Provide details:** **https://github.com/nodejs/node/blob/master/LICENSE**
19. **Latest version:** 17.5.0
20. **Cloud support (Yes/No):** Yes
21. **Applicability:** Used in designing backend applications
22. **Drawbacks:**
    I. Reduces performance when handling Heavy Computing Tasks.
    II. Node.js invites a lot of code changes due to Unstable API.
    III. Node.js Asynchronous Programming Model makes it difficult to maintain code.
    IV. Choose Wisely – Lack of Library Support can Endanger your Code.

Implement linear regression problem using Google colab (Perform preprocessing, training and testing):

DataSet Used:
https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset

Implementation:

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

```
[5] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    from sklearn.linear_model import LinearRegression
    from sklearn.model_selection import train_test_split
```

```
[3] dataset=pd.read_csv("/content/drive/MyDrive/hour.csv")
```

```
dataset.head(10)
```

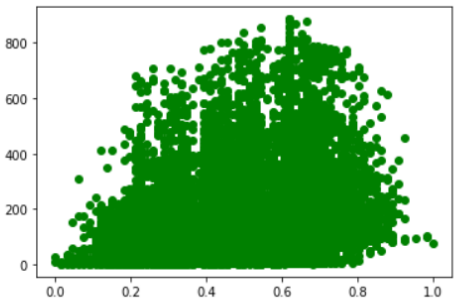
	instant	dteday	season	yr	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	0	6	0	1	0.24	0.2879	0.81	0.0000	3	13	16
1	2	2011-01-01	1	0	1	1	0	6	0	1	0.22	0.2727	0.80	0.0000	8	32	40
2	3	2011-01-01	1	0	1	2	0	6	0	1	0.22	0.2727	0.80	0.0000	5	27	32
3	4	2011-01-01	1	0	1	3	0	6	0	1	0.24	0.2879	0.75	0.0000	3	10	13
4	5	2011-01-01	1	0	1	4	0	6	0	1	0.24	0.2879	0.75	0.0000	0	1	1
5	6	2011-01-01	1	0	1	5	0	6	0	2	0.24	0.2576	0.75	0.0896	0	1	1
6	7	2011-01-01	1	0	1	6	0	6	0	1	0.22	0.2727	0.80	0.0000	2	0	2
7	8	2011-01-01	1	0	1	7	0	6	0	1	0.20	0.2576	0.86	0.0000	1	2	3
8	9	2011-01-01	1	0	1	8	0	6	0	1	0.24	0.2879	0.75	0.0000	1	7	8

```
print('Defining Variables')
X = dataset['atemp']
y = dataset['registered']
```

```
Defining Variables
```

```
[8] plt.scatter(X,y,color='green')
```

```
<matplotlib.collections.PathCollection at 0x7fe74f8e22d0>
```




```
[9] print('Splitting into hour')
    X_hour,X_test,y_hour,y_test=train_test_split(X,y,random_state=0)
```

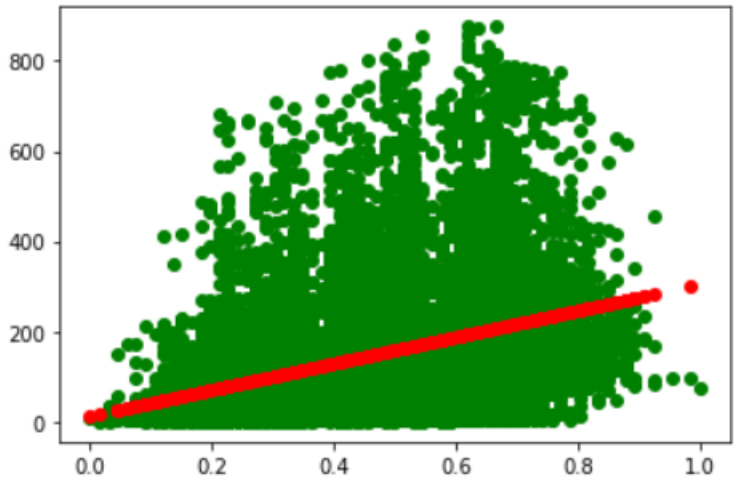
```
Splitting into hour
```

```
[10] plt.scatter(X_hour,y_hour,color='green')
```

```
[16] print('Predicting using trained model X_hour')
     y_pred=lr.predict(X_test.values.reshape(-1,1))
     print(y_pred)
```

```
Predicting using trained model X_hour
[169.41927848 142.95631446 142.95631446 ...  63.5674224  134.13532646
 116.49335044]
```

```
[17] plt.scatter(X_hour,y_hour,color='green')
     plt.scatter(X_test,y_pred,color='red')
     plt.xticks()
     plt.yticks()
     plt.show()
```




```
[19] print('finding intercept and coeff')
     print('Intercept',lr.intercept_)
     print('Coefficient',lr.coef_)
     print(lr.coef_,'x +',lr.intercept_)
```

```
finding intercept and coeff
Intercept 15.037432279963298
Coefficient [291.12171639]
[291.12171639] x + 15.037432279963298
```