

# Practical Machine Learning Lab-1

For this PML lab1 I have create a own sample data as ('ml\_lab1data.csv') which contain month and no.of.accidents.

**Roll no: 205229133**

```
In [18]: import pandas as pd  
data=pd.read_csv('ml_lab1data.csv')
```

```
In [23]: print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 12 entries, 0 to 11  
Data columns (total 2 columns):  
#   Column          Non-Null Count  Dtype  
---  ---  
0   Month           12 non-null    object  
1   No.of.Accidents  12 non-null    int64  
dtypes: int64(1), object(1)  
memory usage: 320.0+ bytes  
None
```

```
In [24]: print(data.shape)
```

```
(12, 2)
```

```
In [25]: print(data.size)
```

```
24
```

```
In [26]: print(data.ndim)
```

```
2
```

## 1.Table Visualization

## Graph Plot

```
In [4]: import pandas as pd
from matplotlib import pyplot as plt

df = pd.read_csv('ml_lab1data.csv')

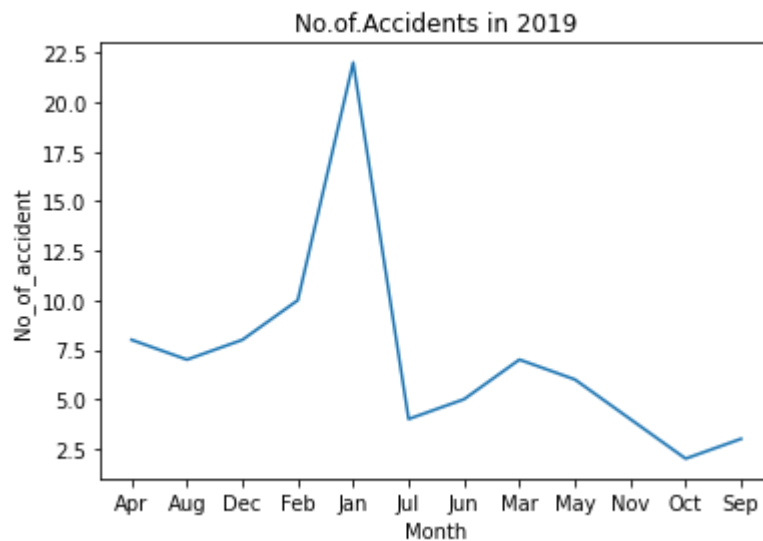
groupedby_acc = df.groupby('Month')[['No.of.Accidents']].sum()

fig, ax = plt.subplots()

ax.plot(groupedby_acc.index, groupedby_acc['No.of.Accidents'])

labels = ax.get_xticklabels()
plt.title('No.of.Accidents in 2019')
plt.xlabel('Month')
plt.ylabel('No_of_accident')

plt.show()
```



## Scatter Plot

```
In [5]: import pandas as pd
from matplotlib import pyplot as plt

df = pd.read_csv('ml_lab1data.csv')

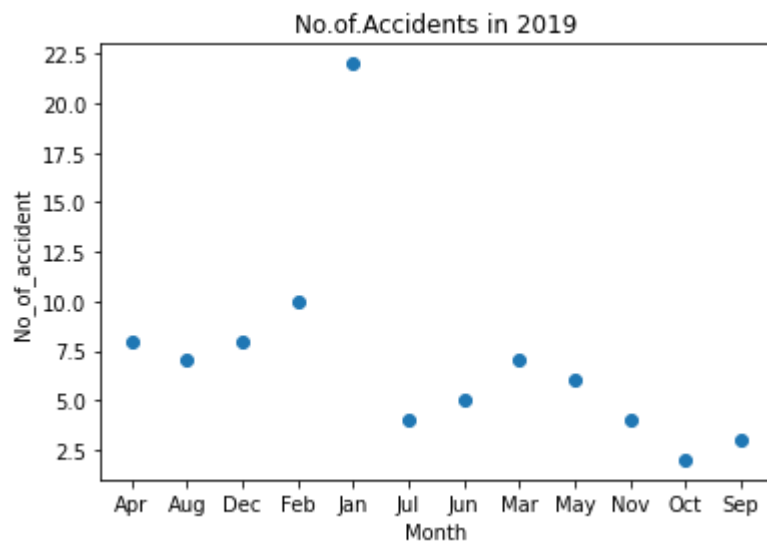
groupedby_acc = df.groupby('Month')[['No.of.Accidents']].sum()

fig, ax = plt.subplots()

ax.scatter(groupedby_acc.index, groupedby_acc['No.of.Accidents'])

labels = ax.get_xticklabels()
plt.title('No.of.Accidents in 2019')
plt.xlabel('Month')
plt.ylabel('No_of_accident')

plt.show()
```



## Bar Chart

```
In [29]: import pandas as pd
from matplotlib import pyplot as plt

df = pd.read_csv('ml_lab1data.csv')

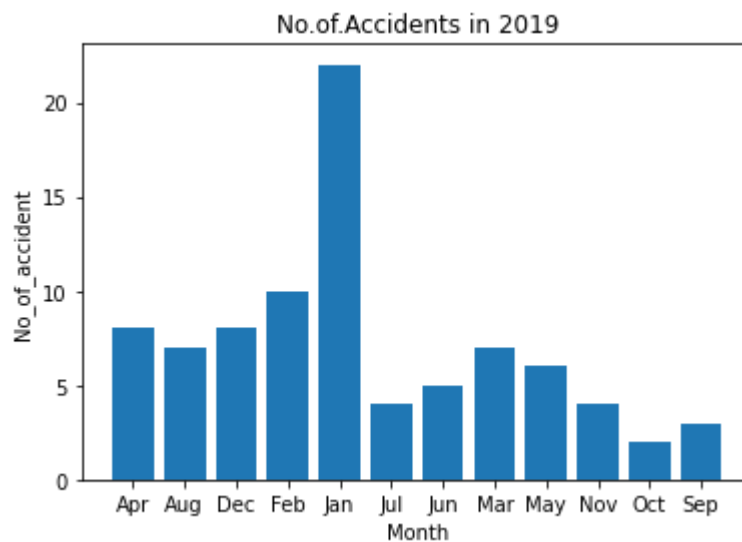
groupedby_acc = df.groupby('Month')[['No.of.Accidents']].sum()

fig, ax = plt.subplots()

ax.bar(groupedby_acc.index, groupedby_acc['No.of.Accidents'])

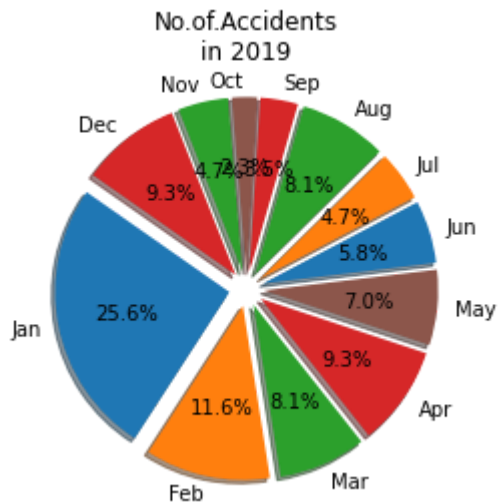
labels = ax.get_xticklabels()
plt.title('No.of.Accidents in 2019')
plt.xlabel('Month')
plt.ylabel('No_of_accident')

plt.show()
```



## Pie Chart

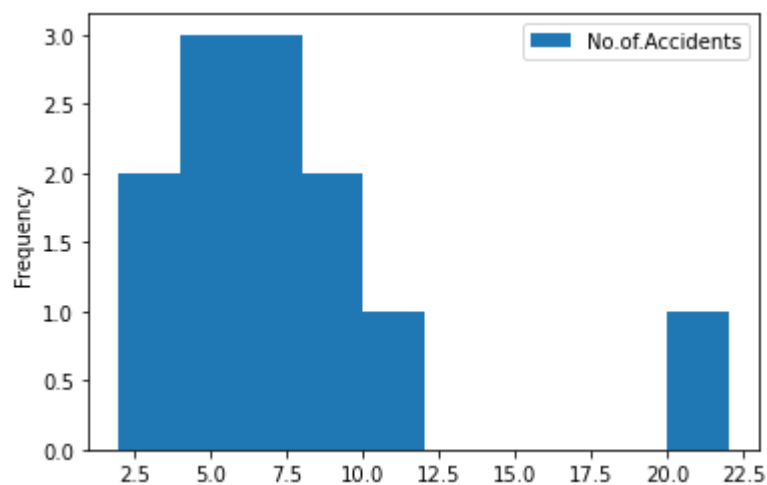
```
In [117]: import matplotlib.pyplot as plt
import pandas as pd
df = pd.read_csv('ml_lab1data.csv')
month_data = df["Month"]
acc_data = df["No.of.Accidents"]
colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#8c564b", "#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#8c564b", "#1f77b4", "#ff7f0e"]
explode = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)
plt.pie(acc_data, labels=month_data, explode=explode, colors=colors,
autopct='%1.1f%%', shadow=True, startangle=145)
plt.title("No.of.Accidents\n"+"in 2019")
plt.show()
```



## Histogram

```
In [136]: import pandas as pd
from matplotlib import pyplot as plt
df = pd.read_csv('ml_lab1data.csv')
df.plot.hist()
```

Out[136]: <AxesSubplot:ylabel='Frequency'>



## 2.Image Visualization

```
In [6]: %matplotlib inline
import imageio
import matplotlib.pyplot as plt
import matplotlib.cbook

pic=imageio.imread('bhc.png')
plt.figure(figsize=(6,6))
plt.imshow(pic)
plt.axis('off')
```

Out[6]: (-0.5, 699.5, 499.5, -0.5)



```
In [7]: %matplotlib notebook
import matplotlib.pyplot as plt

from mpl_toolkits.mplot3d import Axes3D

import numpy as np

from scipy.stats import multivariate_normal

X = np.linspace(-5,5,50)
Y = np.linspace(-5,5,50)
X, Y = np.meshgrid(X,Y)

X_mean = 0; Y_mean = 0

X_var = 5; Y_var = 8

pos = np.empty(X.shape+(2,))

pos[:, :, 0] = X
pos[:, :, 1] = Y

rv = multivariate_normal([X_mean, Y_mean], [[X_var, 0], [0, Y_var]])

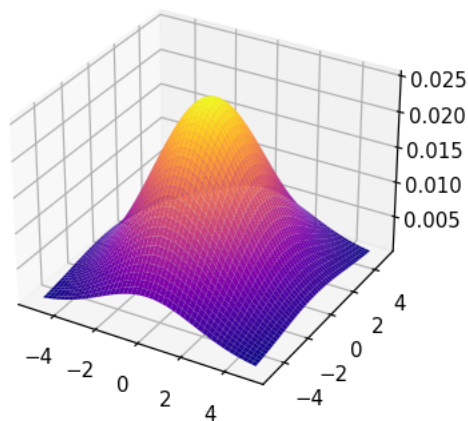
fig = plt.figure()

ax = fig.add_subplot(111, projection='3d')

ax.plot_surface(X, Y, rv.pdf(pos), cmap="plasma")

plt.show()
```

<IPython.core.display.Javascript object>







```
In [3]: # Load imports
import IPython.display as ipd
import librosa
import librosa.display
import matplotlib.pyplot as plt
```

## Audio Player

```
In [4]: ipd.Audio('Jeevamshamayii.m4a')
```

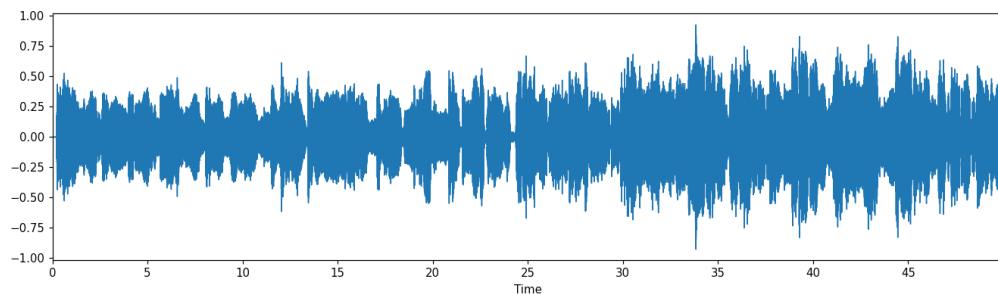
Out[4]:

0:00 / 0:00

## spectrogram

```
In [64]: filename1 = 'Jeevamshamayii.m4a'
plt.figure(figsize=(15,4))
data1,sample_rate1 = librosa.load(filename1, sr=22050, mono=True, offset=0.0, dur
librosa.display.waveplot(data1,sr=sample_rate1, max_points=50000.0, x_axis='time'
```

<IPython.core.display.Javascript object>



Out[64]: <matplotlib.collections.PolyCollection at 0x1f2e3c7c160>

## Text visualization

In [11]: !pip install wordcloud

Collecting wordcloud

Downloading wordcloud-1.8.1-cp38-cp38-win\_amd64.whl (155 kB)

Requirement already satisfied: numpy>=1.6.1 in c:\users\mahesh\anaconda3\lib\site-packages (from wordcloud) (1.19.2)

Requirement already satisfied: pillow in c:\users\mahesh\anaconda3\lib\site-packages (from wordcloud) (8.1.0)

Requirement already satisfied: matplotlib in c:\users\mahesh\anaconda3\lib\site-packages (from wordcloud) (3.3.2)

Requirement already satisfied: certifi>=2020.06.20 in c:\users\mahesh\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2020.12.5)

Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\users\mahesh\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.4.7)

Requirement already satisfied: cycler>=0.10 in c:\users\mahesh\anaconda3\lib\site-packages (from matplotlib->wordcloud) (0.10.0)

Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\mahesh\anaconda3\lib\site-packages (from matplotlib->wordcloud) (1.3.1)

Requirement already satisfied: python-dateutil>=2.1 in c:\users\mahesh\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.8.1)

**Reference** <https://www.geeksforgeeks.org/generating-word-cloud-python/#:~:text=Generating%20Word%20Cloud%20in%20Python.%20Last%20Updated%3A%04-2020.,points%20can%20be%20highlighted%20using%20a%20word%20cloud>  
(<https://www.geeksforgeeks.org/generating-word-cloud-python/#:~:text=Generating%20Word%20Cloud%20in%20Python.%20Last%20Updated%3A%04-2020.,points%20can%20be%20highlighted%20using%20a%20word%20cloud>).

**Reference data set** <https://archive.ics.uci.edu/ml/machine-learning-databases/00380/YouTube-Spam-Collection-v1.zip> (<https://archive.ics.uci.edu/ml/machine-learning-databases/00380/YouTube-Spam-Collection-v1.zip>).

```
In [14]: # Python program to generate WordCloud

# importing all necessary modules
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import pandas as pd

# Reads 'Youtube04-Eminem.csv' file
df = pd.read_csv(r"Youtube04-Eminem.csv", encoding="latin-1")

comment_words = ''
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in df.CONTENT:
    val = str(val)
    tokens = val.split()
# Converts each token into lowercase
for i in range(len(tokens)):
    tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "

wordcloud = WordCloud(width = 800, height = 800,
background_color = 'white',
stopwords = stopwords,
min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

<IPython.core.display.Javascript object>



# i views

# AWESOME

In [16]: `!pip install plotly`

```
Collecting plotly
  Downloading plotly-4.14.3-py2.py3-none-any.whl (13.2 MB)
Collecting retrying>=1.3.3
  Downloading retrying-1.3.3.tar.gz (10 kB)
Requirement already satisfied: six in c:\users\mahesh\anaconda3\lib\site-packages (from plotly) (1.15.0)
Building wheels for collected packages: retrying
  Building wheel for retrying (setup.py): started
  Building wheel for retrying (setup.py): finished with status 'done'
  Created wheel for retrying: filename=retrying-1.3.3-py3-none-any.whl size=11429 sha256=cab58ae52f4edf04f59a45252096be9e097e53fabee195befe362fb063d8bd76
  Stored in directory: c:\users\mahesh\appdata\local\pip\cache\wheels\c4\47\48\0a434133f6d56e878ca511c0e6c38326907c0792f67b476e56
Successfully built retrying
Installing collected packages: retrying, plotly
Successfully installed plotly-4.14.3 retrying-1.3.3
```

**Reference** <https://plotly.com/python/bubble-charts/> (<https://plotly.com/python/bubble-charts/>)

```
In [27]: import plotly.graph_objects as go

fig = go.Figure(data=[go.Scatter(
    x=[1, 2, 3, 4], y=[10, 11, 12, 13],
    text=['A<br>size: 10', 'B<br>size: 20', 'C<br>size: 30', 'D<br>size: 40'],
    mode='markers',
    marker=dict(
        color=['rgb(93, 164, 214)', 'rgb(255, 144, 14)', 'rgb(44, 160, 101)', 'r
        size=[10, 20, 30, 40],
    )
)])

fig.show()
```

In [ ]:

