Gladson Goms

6th Sem

Electronics and communication

Vidyavardhaka college of engineering

MAJOR PROJECT1

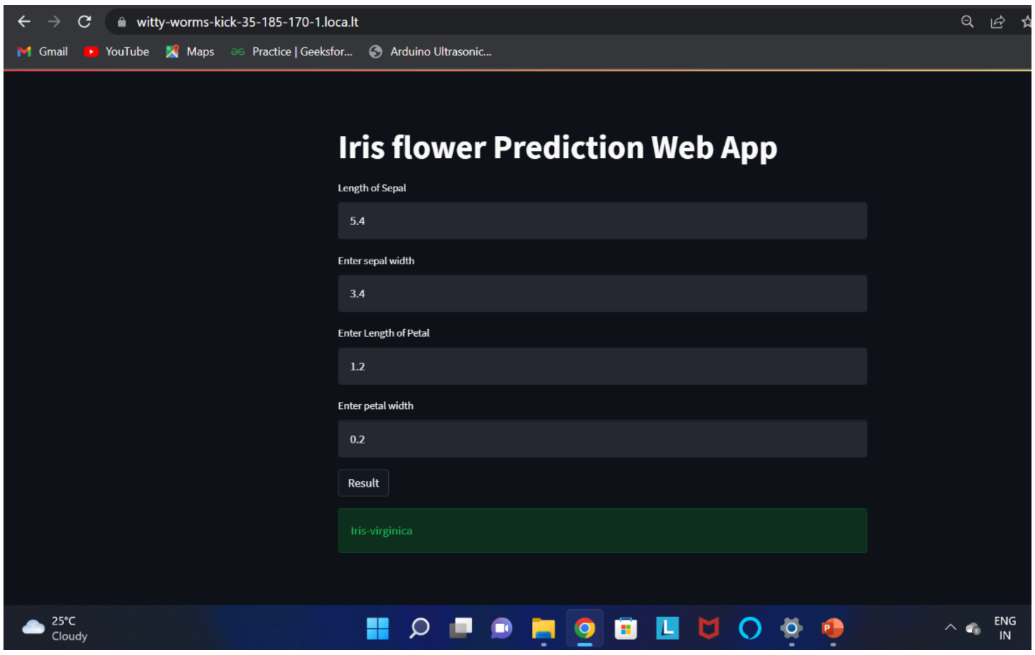
Analysis of dataset on Iris flower and deployed it using heroku and Streamlit

Python code for ML technique(Logistic Regression):

```
import pandas as pd
df=pd.read_csv('/content/IRIS.csv')
df

df.info()

df.shape

df['species'].value_counts()

#to consider i/p & o/p
x=df.iloc[:,0:4].values
x

y=df.iloc[:,4].values
y

#train_test_split
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,random_state=0)

print(x.shape) #150 rows & 4 cols
print(x_train.shape)   #75% is used to training data _____112 rows & 
4 cols
print(x_test.shape) #25% in testing_____38 rows & 4 cols
print(y.shape)
print(y_train.shape)
print(y_test.shape)

#Scaling
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler()

x_train=scaler.fit_transform(x_train)
x_test=scaler.fit_transform(x_test)

#Classifier
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()

model.fit(x_train,y_train)

y_pred=model.predict(x_test)
y_predy_pred=model.predict(x_test)
y_pred

y_test

from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)*100

#individual Prediction
model.predict([[22,3.5,4,5]])

model.predict([[ 5.1, 3.5,  1.4,  0.2        ]])

#Accuracy
```

```
from sklearn.metrics import accuracy_score
accuracy_score(y_pred,y_test)*100

#Joblib
import joblib
joblib.dump(model,'Iris')
#We are creating a new file & dumping the model inside it.
```

Deployed in Heroku & streamlit.



Iris flower Prediction Web App

Length of Sepal: 5.4

Enter sepal width: 3.4

Enter Length of Petal: 1.2

Enter petal width: 0.2

Result

Iris-virginica

Link:https://iris7.herokuapp.com/

Here in this webpage is implemented Heroku cloud using GitHub and streamlit , it takes Sepal length, sepal width, petal length and petal width and tell which type of Iris flower –Iris setose, Iris -versicolor, Iris-virginica based on dataset. An example as shown in the above image that it belongs to Iris virginica family.

Major project -2

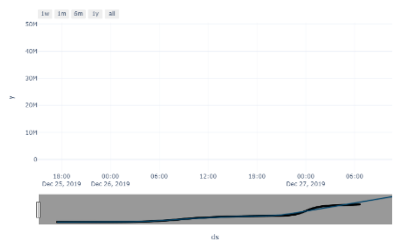
Implemented dataset and performed Exploratory Data Analysis(EDA) on COVID-19 data frame:

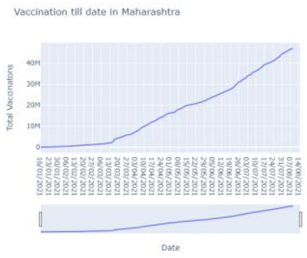
Python code:

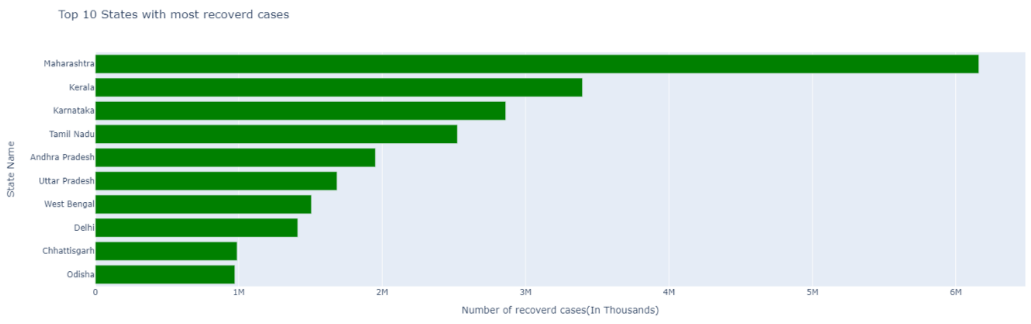
```
import numpy as np

import pandas as pd



df = pd.read_csv('/content/covid_19_india.csv')

df


df.head()


import seaborn as sns

import matplotlib.pyplot as plt

import plotly.graph_objects as go

%matplotlib inline

import warnings

warnings.filterwarnings('ignore')


df.isnull().sum()



df.info()



df = df.drop(['Sno','ConfirmedIndianNational','ConfirmedForeignNational'],axis=1)
```

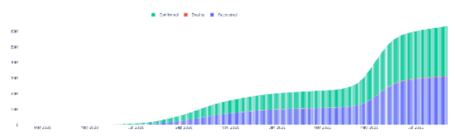
```
df.head()


df['Active'] = df['Confirmed'] - df['Cured'] - df['Deaths']
df.tail()


df['Date'] = pd.to_datetime(df['Date'])


df.info()


india_cases = df[df['Date'] == df['Date'].max()].copy().fillna(0)
india_cases.index = india_cases['State/UnionTerritory']
india_cases = india_cases.drop(['State/UnionTerritory','Time','Date'],axis=1)


india_cases.head()


dff = pd.DataFrame(pd.to_numeric(india_cases.sum())).transpose()
dff.style.background_gradient(cmap='BuGn',axis=1)



Trend = df.groupby(['Date'])['Confirmed','Deaths','Cured',].sum().reset_index()


Trend.head()


fig = go.Figure(go.Bar(x = Trend.Date, y = Trend.Cured, name = 'Recovered'))
fig.add_trace(go.Bar(x = Trend.Date, y = Trend.Deaths, name = 'Deaths'))
fig.add_trace(go.Bar(x = Trend.Date, y = Trend.Confirmed, name = 'Confirmed'))
fig.update_layout(barmode='stack', legend_orientation="h", legend=dict(x=0.3,y=1.1),
            paper_bgcolor='white',
            plot_bgcolor="white")
```

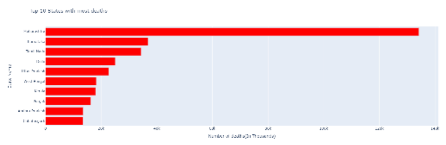
```
fig.show()

import plotly.express as px



def horizontal_bar_chart(dff, x, y, title, x_label, y_label, color):
  fig = px.bar(dff, x=x, y=y, orientation='h', title=title,
            labels={x.name:x_label,
                    y.name:y_label}, color_discrete_sequence=[color])
  fig.update_layout(yaxis={'categoryorder': 'total ascending'})
  fig.show()

top_10_death_states = india_cases.sort_values('Deaths', ascending = False)[:10]

horizontal_bar_chart(top_10_death_states, top_10_death_states.Deaths,
top_10_death_states.index,
                   'Top 10 States with most deaths', 'Number of deaths(In Thousands)','State
Name','red')

top_10_confirmed_states = india_cases.sort_values('Confirmed', ascending = False)[:10]

horizontal_bar_chart(top_10_confirmed_states, top_10_confirmed_states.Confirmed,
top_10_confirmed_states.index,
                   'Top 10 States with most confirmed cases', 'Number of confirmed cases(In
Thousands)','State Name','orange')

top_10_recoverd_states = india_cases.sort_values('Cured', ascending = False)[:10]

horizontal_bar_chart(top_10_recoverd_states, top_10_recoverd_states.Cured,
top_10_recoverd_states.index,
                'Top 10 States with most recoverd cases', 'Number of recoverd cases(In
Thousands)','State Name','green')


vaccination = pd.read_csv('/content/covid_vaccine_statewise.csv.zip')


vaccination.tail()


vaccination.head()


vaccination['Total Vaccinatons'] = vaccination['First Dose
Administered']+vaccination['Second Dose Administered']


#Renaming columns
vaccination.rename(columns = {'Updated On':'Date'}, inplace = True)


Maharashtra = vaccination[vaccination["State"]=="Maharashtra"]


fig = px.line(Maharashtra,x="Date",y="Total Vaccinatons",title="Vaccination till date in
Maharashtra")
fig.update_xaxes(rangeslider_visible=True)


from fbprophet import Prophet
from fbprophet.plot import plot_plotly, add_changepoints_to_plot
from plotly.offline import iplot, init_notebook_mode


model = Prophet()


Confirmed = Trend.loc[:,['Date', 'Confirmed']]
Confirmed.tail()
Confirmed.columns = ['ds', 'y']


model.fit(Confirmed)


future = model.make_future_dataframe(periods=60)

future.tail()


forecast_india_conf = model.predict(future)

forecast_india_conf


fig = plot_plotly(model, forecast_india_conf)

fig.update_layout(template='plotly_white')

iplot(fig)
```

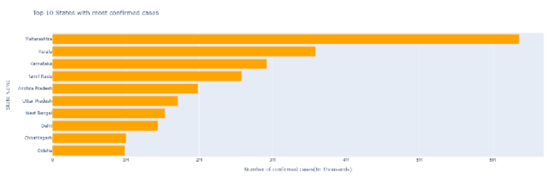












Reference:

github: https://github.com/gladsongoms/Iris

Colab sheet:

Covid-19 -- https://colab.research.google.com/drive/1XwRnq4YweZNW0nGx-l_8-1UCUyQM4q74#scrollTo=cMG-ILRdA0gB

Iris ---https://colab.research.google.com/drive/13_xEwW9pVd_o3QoWgfY-rVTZnpiwpTch#scrollTo=lWJ9EwmijTsg

Iris (streamlit file)--
https://colab.research.google.com/drive/1JjqzQrfARcrcfNs84sUrHGl264MTgkOw#scrollTo=lkADehRFBjy5