

Lab 13

R. Harini

18BCE1010

```
C:\Windows\system32\cmd.exe
Server will be initialized using port: 9004

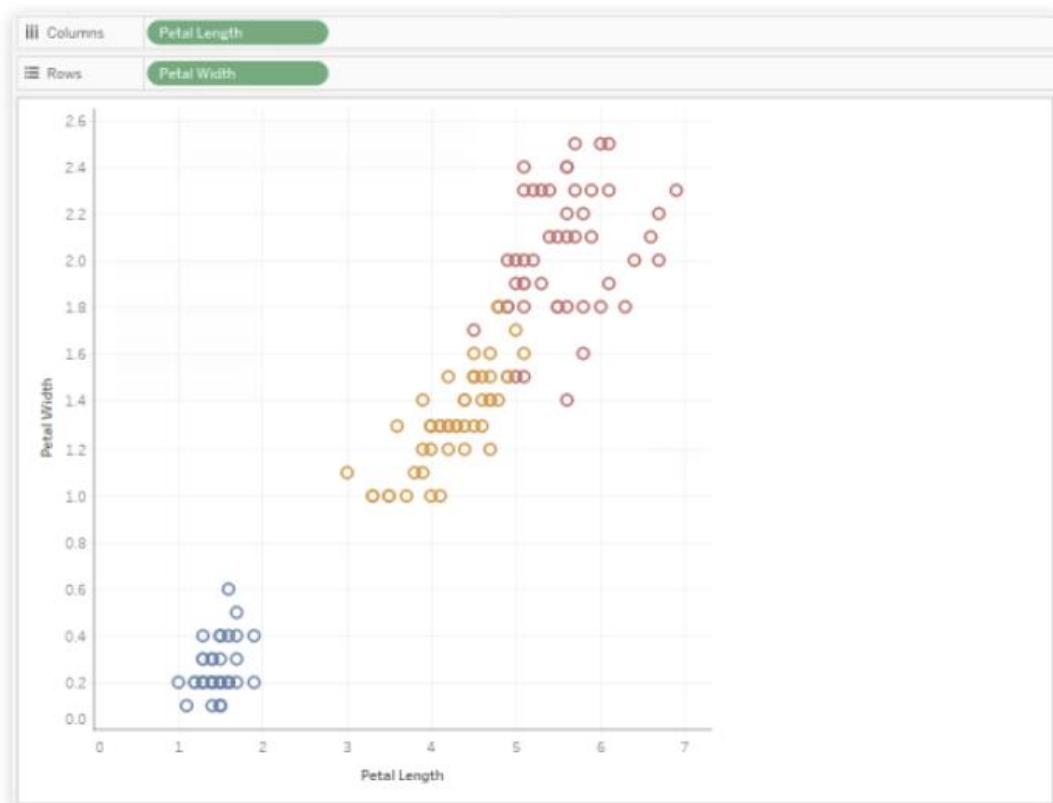
~~~~~ Looking for existing Anaconda installation ~~~~~
INFO: Could not find files for the given pattern(s).
No existing Anaconda installation found.
Looking for a local installer...

~~~~~ Downloading and installing Anaconda ~~~~~
Download completed.
Installing Anaconda...

~~~~~ Activating the environment ~~~~~

The system cannot find the path specified.
No Tableau-Python-Server environment found. Creating one...
WARNING: A space was detected in your requested environment path
'C:\Users\Prakash Kannan\Anaconda3\envs\Tableau-Python-Server'
Spaces in paths can sometimes be problematic.
Source: C:\Users\Prakash Kannan\Anaconda3
Destination: C:\Users\Prakash Kannan\Anaconda3\envs\Tableau-Python-Server
The following packages cannot be cloned out of the root environment:
- defaults::conda-4.7.12-py37_0
- defaults::conda-build-3.18.9-py37_3
- defaults::conda-env-2.6.0-1
Packages: 273
Files: 12
Preparing transaction: done
Verifying transaction: done
Executing transaction: -
```

Using Naïve Bayes algorithm for Iris dataset:



```

SCRIPT_REAL("
import numpy as np
from sklearn.naive_bayes import GaussianNB

# create the model
model = GaussianNB()

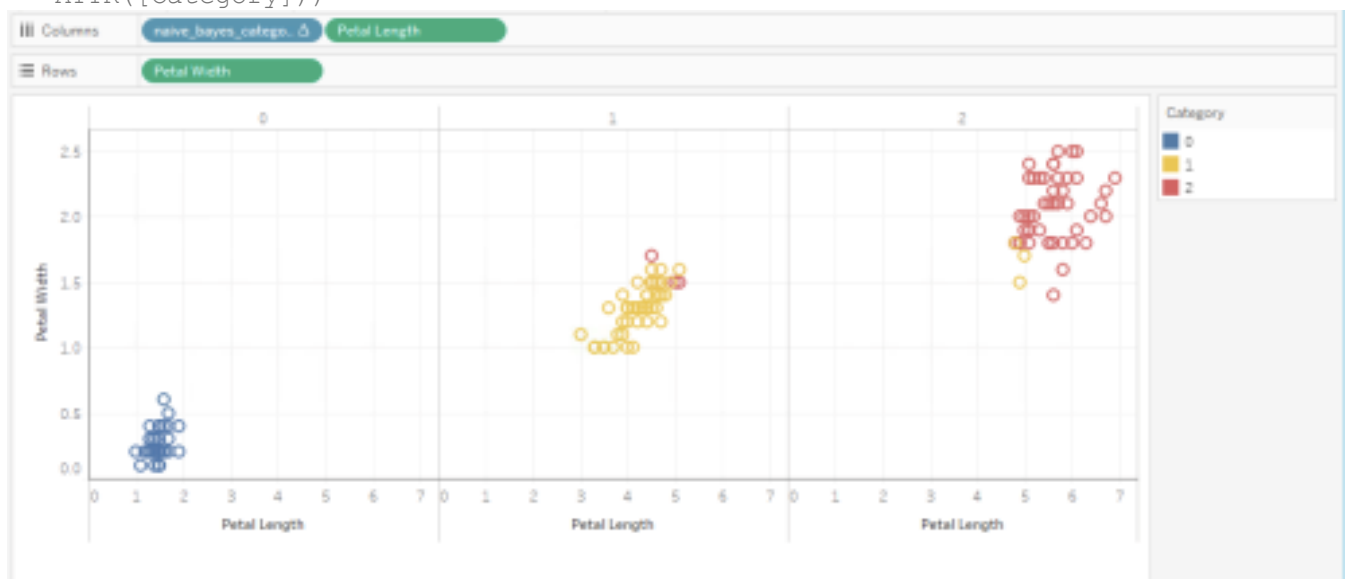
# transform input data
data_x = np.transpose(np.array([_arg1, _arg2, _arg3, _arg4]))
data_y = np.array(_arg5)

# fit the model
model.fit(data_x, data_y)

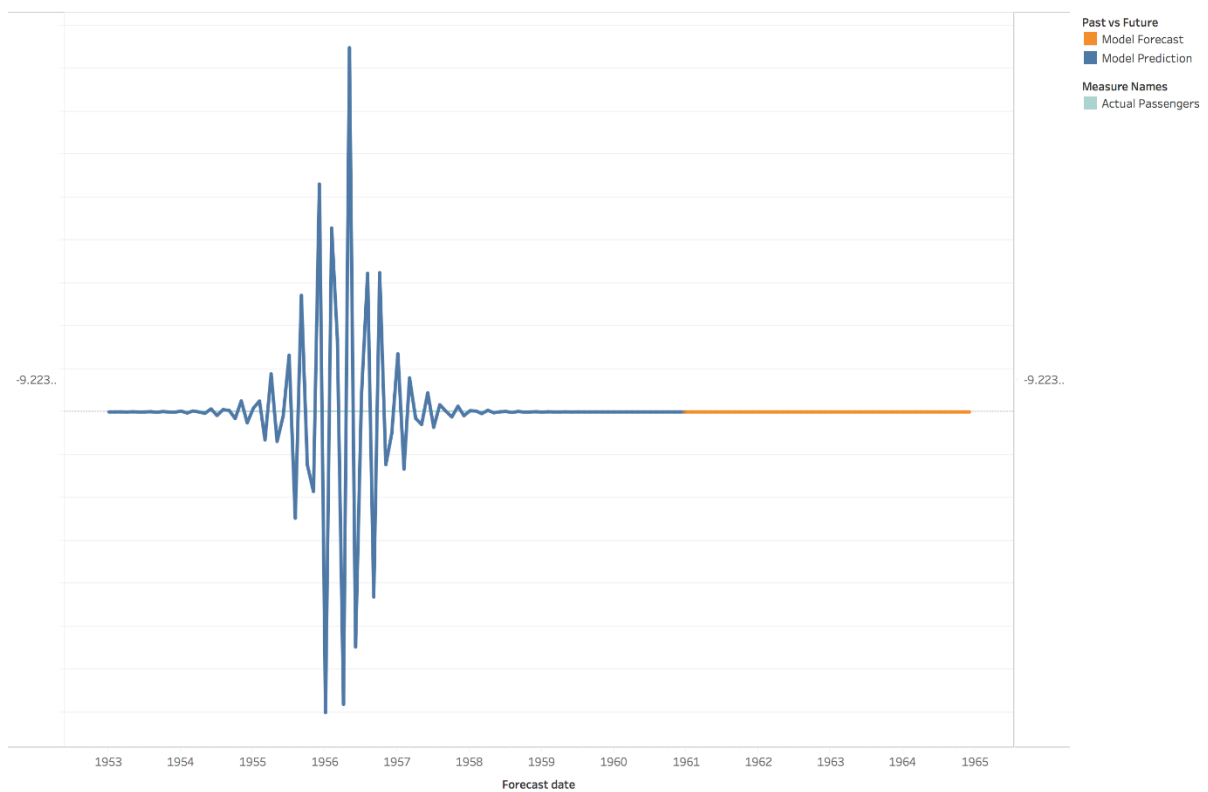
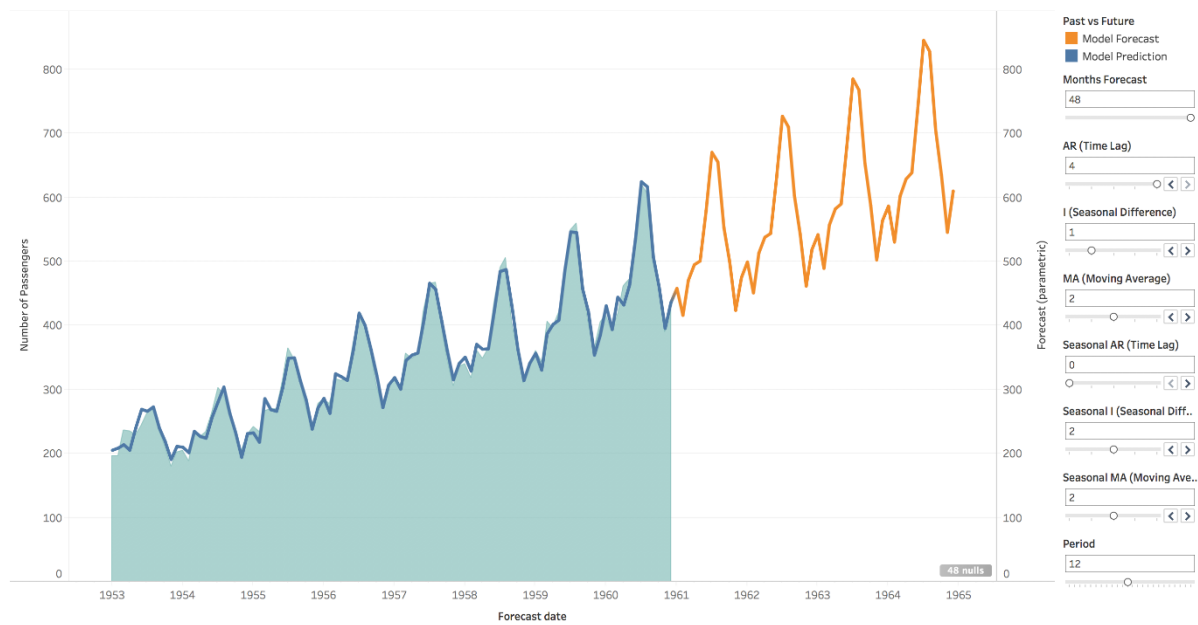
# predict the category for input data
predicted_category = model.predict(data_x)

# transform output
return list(np.round(predicted_category, decimals=2))
", ATTR([Petal Length]),
  ATTR([Petal Width]),
  ATTR([Sepal Length]),
  ATTR([Sepal Width]),
  ATTR([Category]))

```



Regression model for Air Passengers data:



```
SCRIPT_REAL(
,
import pandas as pd
import statsmodels.api as sm

dates = _arg1
passengers = _arg2
order_ar = min(_arg3)
```

```

order_i = min(_arg4)
order_ma = min(_arg5)
months_forecast = min(_arg6)
seasonal_ar = min(_arg7)
seasonal_i = min(_arg8)
seasonal_ma = min(_arg9)
period = min(_arg10)

y = pd.Series(data=passengers, index=dates)

mod = sm.tsa.statespace.SARIMAX(y,
                                order=(order_ar, order_i, order_ma),
                                seasonal_order=(seasonal_ar, seasonal_i,
seasonal_ma, period),
                                enforce_stationarity=False,
                                enforce_invertibility=False)

results = mod.fit()

pred_uc = results.get_forecast(steps=months_forecast)

pred = results.get_prediction(start=(pd.to_datetime("1949-01-01")) +
pd.DateOffset(months=months_forecast), dynamic=False)

data = list(pred.predicted_mean)
data.extend(pred_uc.predicted_mean)

return data
',
ATTR([Month]),
ATTR([#Passengers]),
MIN([AR (Time Lag)]),
MIN([I (Seasonal Difference)]),
MIN([MA (Moving Average)]),
MIN([Months Forecast]),
MIN([Seasonal AR (Time Lag)]),
MIN([Seasonal I (Seasonal Difference)]),
MIN([Seasonal MA (Moving Average)]),
MIN([Period]))

```