

Machine learning y Deep learning con Python:

Lectura 2

Ing. Pedro Rotta

Universidad de Piura - Vida Universitaria

Enero-2022

Plotly

Plotly es una librería gráfica de python que permite visualizar gráficas y trazas dinámicas, de manera que se puede visualizar gráficas interactivas en el uso de Ipython notebooks como Google Colab, escalables a una aplicación web interactiva con el framework de python Dash.

La sintaxis de su instalación en cmd es:

```
pip install plotly
```

```
pip install dash
```

Plotly tiene 2 módulos que trabajan con plotly que son express y graph_objects.

Plotly

Para importar el módulo express la sintaxis estándar es:

```
import plotly.express as px
```

Se puede generar una gráfica de dispersión con el método **scatter**

Existen varias formas de generar un gráfico por dispersión mediante plotly. Por medio de listas, por medio de arrays y por medio directo de un dataframe. La sintaxis genérica para este método es:

```
import plotly.express as px
import pandas as pd

fig = px.scatter(data_frame = , x= , y= , color =,
size= )
```

Dispersión

La librería ofrece, al igual que seaborn y que sklearn una variedad de dataframes de prueba, que permiten descargar conjuntos de datos para analizar brevemente.

Estos conjuntos de datos pueden ser visualizados en <https://github.com/plotly/datasets>. La forma de usarlos, directamente desde el código, es mediante el comando **data** y su comando de creación:

```
import plotly.express as px  
  
df_iris = px.data.iris()
```

Documentación: <https://plotly.com/python-api-reference/generated/plotly.express.scatter>
Ver Problema 1

Línea

También es útil definir, los gráficos de línea que se pueden realizar, con ayuda de numpy podemos realizar gráficos interactivos de funciones o con pandas, analizar algunas variables.

La sintaxis de análisis de gráficos de línea es :

```
import plotly.express as px  
  
import numpy as np  
  
px.line(data_frame =, x=, y, labels= )
```

En el siguiente problema, veremos cómo se puede definir un gráfico de línea.

Documentación: <https://plotly.com/python-api-reference/generated/plotly.express.line>

Añadir trazas

Podemos añadir trazas a nuestra figura, para que aparezcan dentro de la misma figura. Generamos una figura nueva con el módulo de graph objects que se importa con la siguiente sintaxis:

```
import plotly.graph_objects as go
import numpy as np
fig = go.Figure()
```

Para añadir una traza, usamos la siguiente sintaxis:

```
fig.add_trace(go.Scatter(x = , y=, marker_color))
```

Ver problema 4

Histogramas y Gráficos de distribución

En plotly también se pueden realizar gráficos de distribución e histogramas

Para ello se utiliza la siguiente sintaxis, teniendo en cuenta el número de bins:

```
fig = px.histogram(df, x= , nbins= )
```

Donde nbins es el número de bins.

Ver problema 5.

Histogramas y Gráficos de distribución

En plotly también se pueden realizar gráficos de distribución e histogramas

Para ello se utiliza la siguiente sintaxis, teniendo en cuenta el número de bins:

```
fig = px.histogram(df, x= , nbins= )
```

Donde nbins es el número de bins.

Ver problema 5.

Para un gráfico de distribución combinado, la sintaxis en plotly cambia y se expresa como:

```
import plotly.figure_factory as ff  
  
ff.create_displot(histdata, grouplabels)
```

Para una distribución y un título. Ver problema 6

Linear Regresion

Es útil también reconocer, que se puede usar plotly para hacer un problema de regresión lineal de una manera bastante gráfica. Con ayuda del método scatter, podemos trazar, la gráfica de una regresión lineal, como se ve en el problema 7

También se puede obtener una regresión aparente con la propiedad trendline dentro de la forma scatter. como se ve en la sintaxis:

```
px.scatter(df,x = ,y= , trendline = 'ols')
```