

Juan Manuel Sanchez Pareja -1010107723

```
Computación Blanda - C X | web_traffic.csv - Google X | Mi unidad - Google Drive X | Home X | Untitled - Jupyter Note X | Untitled - Jupyter Note X | localhost:8889/Files/De X | + X
localhost:8889/notebooks/Descargas/Untitled.ipynb
Jupyter Untitled Last Checkpoint: hace 8 minutos (autosaved)
File Edit View Insert Cell Kernel Widgets Help
In [2]: #EJEMPLO EN CLASE
#Primer Programa Machine Learning

#Juan Manuel Sanchez
import numpy as np
print('Presentado Por : Juan Manuel Sanchez')

data = np.genfromtxt("web_traffic.tsv", delimiter="\t")
print(data[:20], "\n")
# Número de datos
print(data.shape)

Presentado Por : Juan Manuel Sanchez
[[1.000e+00 2.272e+03]
 [2.000e+00 nan]
 [3.000e+00 1.386e+03]
 [4.000e+00 1.365e+03]
 [5.000e+00 1.488e+03]
 [6.000e+00 1.337e+03]
 [7.000e+00 1.883e+03]
 [8.000e+00 2.283e+03]
 [9.000e+00 1.335e+03]
 [1.000e+01 1.025e+03]]
(743, 2)

In [5]: # La primera columna es: Número de horas
# La segunda columna es: Número de tareas ejecutadas
data = np.genfromtxt("web_traffic.tsv", delimiter="\t")
print(data[:20], "\n")
# Número de datos
print(data.shape)

[[1.000e+00 2.272e+03]
 [2.000e+00 nan]
 [3.000e+00 1.386e+03]
 [4.000e+00 1.365e+03]
 [5.000e+00 1.488e+03]
 [6.000e+00 1.337e+03]
 [7.000e+00 1.883e+03]
 [8.000e+00 2.283e+03]
 [9.000e+00 1.335e+03]
 [1.000e+01 1.025e+03]
 [1.100e+01 1.139e+03]
 [1.200e+01 1.477e+03]]
```

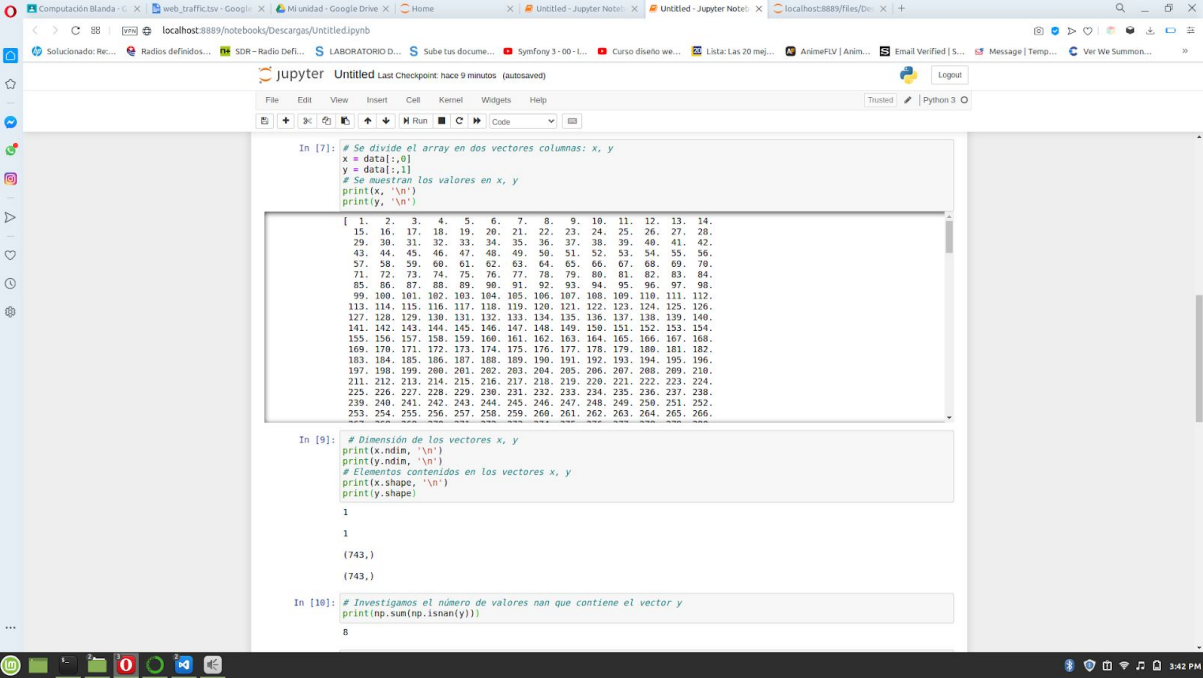
```
Computación Blanda - C X | web_traffic.csv - Google X | Mi unidad - Google Drive X | Home X | Untitled - Jupyter Note X | Untitled - Jupyter Note X | localhost:8889/Files/De X | + X
localhost:8889/notebooks/Descargas/Untitled.ipynb
Jupyter Untitled Last Checkpoint: hace 8 minutos (autosaved)
File Edit View Insert Cell Kernel Widgets Help
In [5]: # La primera columna es: Número de horas
# La segunda columna es: Número de tareas ejecutadas
data = np.genfromtxt("web_traffic.tsv", delimiter="\t")
print(data[:20], "\n")
# Número de datos
print(data.shape)

[[1.000e+00 2.272e+03]
 [2.000e+00 nan]
 [3.000e+00 1.386e+03]
 [4.000e+00 1.365e+03]
 [5.000e+00 1.488e+03]
 [6.000e+00 1.337e+03]
 [7.000e+00 1.883e+03]
 [8.000e+00 2.283e+03]
 [9.000e+00 1.335e+03]
 [1.000e+01 1.025e+03]
 [1.100e+01 1.139e+03]
 [1.200e+01 1.477e+03]
 [1.300e+01 1.203e+03]
 [1.400e+01 1.311e+03]
 [1.500e+01 1.299e+03]
 [1.600e+01 1.494e+03]
 [1.700e+01 1.159e+03]
 [1.800e+01 1.365e+03]
 [1.900e+01 1.272e+03]
 [2.000e+01 1.246e+03]]
(743, 2)

In [7]: # Se divide el array en dos vectores columnas: x, y
x = data[:,0]
y = data[:,1]
# Se muestran los valores en x, y
print(x, "\n")
print(y, "\n")

[ 1.  2.  3.  4.  5.  6.  7.  8.  9. 10. 11. 12. 13. 14.
 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28.
 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42.
 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56.
 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70.
 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84.
 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98.
 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112.
 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126.
 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140.]
```

Juan Manuel Sanchez Pareja -1010107723



The screenshot shows a Jupyter Notebook interface with three code cells. The first cell contains a comment in Spanish and a list of 255 integers. The second cell contains code to print the dimensions and shapes of the data. The third cell contains code to print the number of NaN values in the data.

```
In [7]: # Se divide el array en dos vectores columnas: x, y
x = data[:,0]
y = data[:,1]
# Se muestran los valores en x, y
print(x, '\n')
print(y, '\n')
```

```
[ 1.  2.  3.  4.  5.  6.  7.  8.  9. 10. 11. 12. 13. 14.
15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28.
29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42.
43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56.
57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70.
71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84.
85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98.
99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112.
113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126.
127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140.
141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154.
155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168.
169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182.
183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196.
197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210.
211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224.
225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238.
239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252.
253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266.]
```

```
In [9]: # Dimensión de los vectores x, y
print(x.ndim, '\n')
print(y.ndim, '\n')
# Elementos contenidos en los vectores x, y
print(x.shape, '\n')
print(y.shape)
```

```
1
1
(743,)
(743,)
```

```
In [10]: # Investigamos el número de valores nan que contiene el vector y
print(np.sum(np.isnan(y)))
```

```
8
```

```

In [9]: # Dimension de los vectores x, y
print(x.ndim, '\n')
print(y.ndim, '\n')
# Elementos contenidos en los vectores x, y
print(x.shape, '\n')
print(y.shape)
1
1
(743,)
(743,)

In [10]: # Investigamos el número de valores nan que contiene el vector y
print(np.sum(np.isnan(y)))
8

In [11]: # Número de elementos en x, y, antes de ser comprimidos
print(x.shape, '\n')
print(y.shape, '\n')
# Se eliminan los elementos nan tanto de x como de y
x = x[~np.isnan(y)]
y = y[~np.isnan(y)]
# Se cuenta el número de elementos tanto de x como de y
print(x.shape, '\n')
print(x.shape, '\n')
(743,)
(743,)
(735,)
(735,)

In [16]: # Se importa la librería para graficar
import matplotlib.pyplot as plt
# Dibuja los puntos (x,y) con círculos de tamaño 10
plt.scatter(x, y, s=10)
# Títulos de la gráfica
plt.title("Tráfico Web en el último mes")
plt.xlabel("Tiempo")
plt.ylabel("Accesos/Hora")
# Se modifican los rangos para notar que sucede

```

