



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CARRERA DE SOFTWARE**  
**ASIGNATURA: INTELIGENCIA ARTIFICIAL NIVEL: 06**

|                                    |  |
|------------------------------------|--|
| <b>Nombres completos:</b>          | William Sánchez                          |
| <b>Fecha:</b>                      | 15/02/2022                               |
| <b>Tema:</b>                       | Informe de la evaluación de supletorio   |
| <b>Objetivo de esta actividad:</b> | Realizar las evidencias de la evaluación |

### Contenido

|                                    |          |
|------------------------------------|----------|
| <b>INDICACION .....</b>            | <b>2</b> |
| <b>DESARROLLO.....</b>             | <b>3</b> |
| Descargar el Data Set.....         | 3        |
| Carga de datos en python .....     | 4        |
| Preprocesamiento de datos.....     | 4        |
| Entrenamiento del modelo .....     | 5        |
| Red Neuronal .....                 | 5        |
| Matriz de confusion .....          | 6        |
| Curva ROC .....                    | 8        |
| Graficas del Training y Test ..... | 9        |

## INDICACION

Utilizando Python elabore una Red Neuronal Multicapa, MLP, (no se acepta otro algoritmo) para predecir la autenticación de billetes utilizando 5 atributos.

Realice el preprocesamiento de datos. Justifique técnicamente cada uno de los parámetros e hiperparámetros utilizados en el entrenamiento de la red.

Muestre las métricas y gráficas de rendimiento obtenidas: accuracy, precisión, sensibilidad, especificidad, F1-score, área bajo la curva (AUC), gráfico de la curva ROC, learning curves.

Evaluación:

- • Funcionalidad: 4
- • Justificación técnica: 2
- • Métricas: 4

Nota: Realice un informe en Word o pdf. Trabaje en grupo de 2 integrantes o individual.

Exámenes similares serán

Autoguardado

Vista protegida

Buscar (Alt+Q)

William Sanchez

Windows icons

Archivo Inicio Insertar Disposición de página Fórmulas Datos Revisar Vista Ayuda

VISTA PROTEGIDA Tenga cuidado: los archivos de Internet pueden contener virus. Si no tiene que editarlo, es mejor que siga en Vista protegida.

Habilitar edición

F7

X V f

| A  | B        | C        | D        | E        | F | G | H | I | J | K | L | M | N | O | P | Q |
|----|----------|----------|----------|----------|---|---|---|---|---|---|---|---|---|---|---|---|
| 1  | 36.216   | 86.661   | -28.073  | -0.44699 | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 2  | 45.459   | 81.674   | -24.586  | -14.621  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 3  | 3.866    | -26.383  | 19.242   | 0.10645  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 4  | 34.566   | 95.228   | -40.112  | -35.944  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 5  | 0.32924  | -44.552  | 45.718   | -0.9888  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 6  | 43.684   | 96.718   | -39.606  | -31.625  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 7  | 35.912   | 30.129   | 0.72888  | 0.56421  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 8  | 20.922   | -6.81    | 84.636   | -0.60216 | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 9  | 32.032   | 57.588   | -0.75345 | -0.61251 | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 10 | 15.356   | 91.772   | -22.718  | -0.73535 | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 11 | 12.247   | 87.779   | -22.135  | -0.80647 | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 12 | 39.899   | -27.066  | 23.946   | 0.86291  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 13 | 18.993   | 76.625   | 0.15394  | -31.108  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 14 | -15.768  | 10.843   | 25.462   | -29.362  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 15 | 3.404    | 87.261   | -29.915  | -0.57242 | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 16 | 46.765   | -33.895  | 34.896   | 14.771   | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 17 | 26.719   | 30.646   | 0.37158  | 0.58619  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 18 | 0.80355  | 28.473   | 43.439   | 0.6017   | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 19 | 14.479   | -48.794  | 83.428   | -21.086  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 20 | 52.423   | 110.272  | -4.353   | -41.013  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 21 | 57.867   | 78.902   | -26.196  | -0.48708 | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 22 | 0.3292   | -44.552  | 45.718   | -0.9888  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 23 | 39.362   | 101.622  | -38.235  | -40.172  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 24 | 0.93584  | 88.855   | -16.831  | -16.599  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 25 | 44.338   | 9.887    | -46.795  | -37.483  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 26 | 7.0057   | -54.981  | 83.368   | -28.715  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 27 | 11.432   | -37.413  | 55.777   | -0.63578 | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 28 | -0.38214 | 83.909   | 21.624   | -37.405  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 29 | 65.633   | 98.187   | -44.113  | -32.258  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 30 | 48.906   | -33.584  | 34.202   | 10.905   | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 31 | -0.24811 | -0.17797 | 49.068   | 0.15429  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 32 | 14.884   | 36.274   | 3.308    | 0.48921  | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 33 | 42.969   | 7.617    | -23.874  | -0.96164 | 0 |   |   |   |   |   |   |   |   |   |   |   |
| 34 | -0.96511 | 94.111   | 17.305   | -48.629  | 0 |   |   |   |   |   |   |   |   |   |   |   |

data\_banknote\_authentication

## Carga de datos en python

Cargamos los datos utilizando un lenguaje de python(interprete) y en el ambiente de spyder

Para la carga de datos utilizamos las importaciones de pandas para leer archivos tipo excel, posteriormente los guardamos en variables para que esten separados de mayor manera para el identificar cuales seran nuestras training\_data y target\_data, relacionadas con ls columnas y filas respectivas

```
WilliamSanchezSuple.py X
1  # -*- coding: utf-8 -*-
2  """
3  Created on Tue Feb 15 11:14:53 2022
4
5  @author: HP
6  """
7
8  import numpy as np
9  from keras import models
10 from keras.layers.core import Dense
11 from sklearn import preprocessing
12 import pandas as pd
13
14
15 """Cargar el dataset"""
16 admissions = pd.read_excel('banknote.xlsx')
17 datos = admissions.values
18 entrada = datos[:,0:4]
19 salida = datos[:,4]
20
21
22 training_data = np.array(entrada, "float32")
```

## Preprocesamiento de datos

Para el preprocesamiento de datos utilice el StandardScaler() ya que me dio datos cercanos a 1 y 0, ademas Tambien le intente con el MinMaxScaler pero no me daba resultados favorables ya que me daban valores 4578.0 por ejemplo asi que el mayor Metodo para normalizer este data set es StandardScaler

```
23
24 """Normalizando Los datos con el metodo/min_max_scaler"""
25 from sklearn import preprocessing
26 #returns a numpy array
27 #min_max_scaler = preprocessing.MinMaxScaler()
28 #x_scaled = min_max_scaler.fit_transform(training_data)
29 #training_data = pd.DataFrame(training_data)
30 #training_data
31
32
33 """Normalizando Los datos con StandarScaler"""
34 training_data = preprocessing.StandardScaler().fit_transform(training_data)
35 training_data = pd.DataFrame(training_data)
36 training_data
37 # ground truth
```

## Entrenamiento del modelo

Para el entrenamiento del modelo se realiza que un 0.2 por ciento sea fijado para el entrenamiento

```
39 """Entrenamos el modelo"""
40 from sklearn.model_selection import train_test_split
41 X_train,X_test, y_train,y_test = train_test_split(training_data,target_data,test_size=0.2,random_state=0)
42
```

## Red Neuronal

Para realizar el entrenamiento de la red neuronal

Se entreno con 33 neuronas, la function de activacion fue sigmoid ya que es Buena para trabajar con clasificacion binaria y una neurona de salida,ademas de que utilizamos la function relu para las capas ocultas, para la función de Perdida se utilize binary\_crossentropy ya que es buenas para determinar y trabajar con clasificacion binaria(2 clases) y el optimizador adam. Entrenamos el modelo internamente con 20 epocas

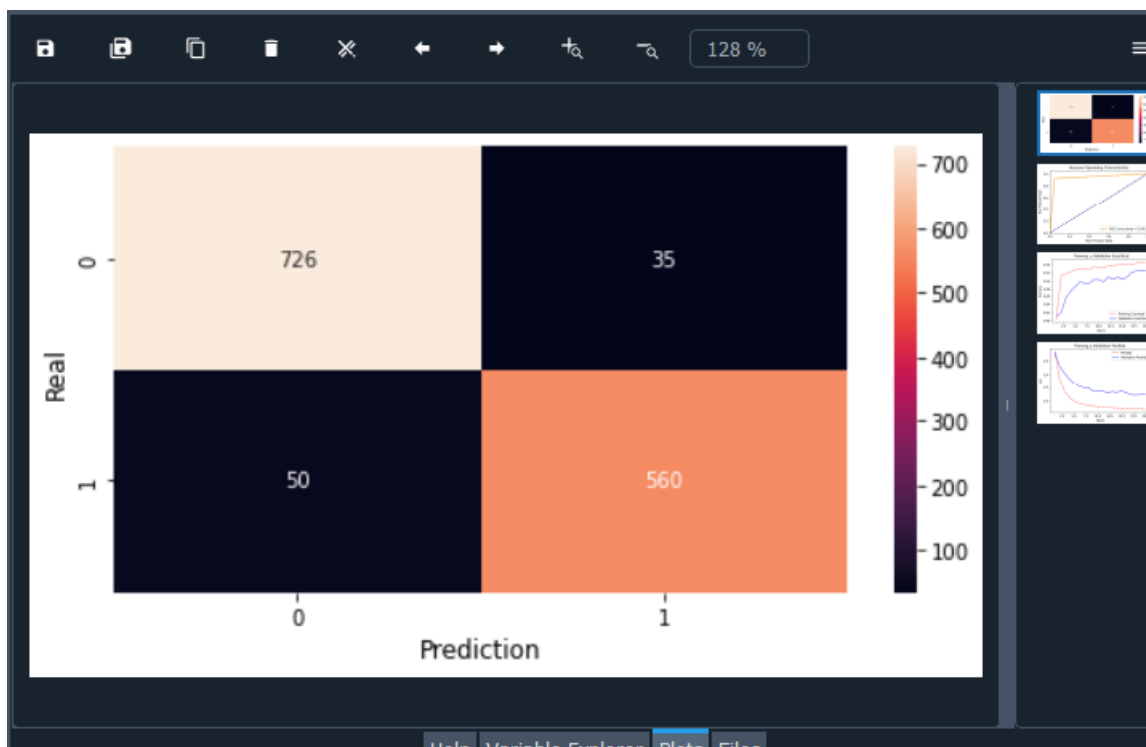
```
42 X_train,X_test, y_train,y_test = train_test_split(training_data,target_data,test_size=0.2,random_state=0)
43
44 """RED NEURONAL"""
45 model = models.Sequential()
46 model.add(Dense(25, activation='relu', input_dim=4))
47 model.add(Dense(1, activation='sigmoid'))
48 model.compile(loss='binary_crossentropy',
49               optimizer='adam',
50               metrics=['accuracy'])
51 modelo=model.fit(training_data, target_data, epochs=20, batch_size=8, validation_split=0.25)
52
```

Determinamos los parametros para la evaluacion del modelo de train y del test

```
53
54 """Evaluando el modelo train y test"""
55
56 print("Evaluación del modelo test")
57 scores_test = model.evaluate(X_test, y_test)
58 print("\%s: %.4f" % (model.metrics_names[0], scores_test[0])) # loss
59 print("\n%s: %.2f%%" % (model.metrics_names[1], scores_test[1]*100)) # accuracy
60
61 print("Evaluación del modelo training")
62 scores = model.evaluate(training_data,target_data)
63 print("\%s: %.4f" % (model.metrics_names[0], scores[0])) # loss
64 print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100)) # accuracy
65
66 test_loss, test_accuracy = model.evaluate(X_test, y_test)
67
68 print("Exactitud del Entrenamiento", test_accuracy)
69 print("Pérdida del Entrenamiento", test_loss)
70
71
```

## Matriz de confusión

```
76
77 """Matriz de confusión"""
78 from sklearn.metrics import confusion_matrix
79 cm = confusion_matrix(y_true, y_pred)
80 print("Matriz de confusión:")
81 print(cm)
82 import matplotlib.pyplot as plt
83 import seaborn as sns
84 plt.figure(figsize = (8,4))
85 sns.heatmap(cm, annot=True, fmt='d')
86 plt.xlabel('Prediction', fontsize = 12)
87 plt.ylabel('Real', fontsize = 12)
88 plt.show()
89
```



Rendimiento obtenidas: accuracy, precisión, sensibilidad, especificidad, F1-score, área bajo la curva (AUC),

```
89
90
91 """Exactitud:"""
92 from sklearn.metrics import accuracy_score
93 acc = accuracy_score(y_true, y_pred)
94 print('acc:', acc)
95
96 """Sensibilidad:"""
97 from sklearn.metrics import recall_score
98 rec = recall_score(y_true, y_pred)
99 print('recall:', rec)
100
101 """Precisión:"""
102 from sklearn.metrics import precision_score
103 prec = precision_score(y_true, y_pred)
104 print('precision:', prec)
105
106 """Puntuación F1:"""
107 from sklearn.metrics import f1_score
108 f1 = f1_score(y_true, y_pred)
109 print('F1:', f1)
110
111 """Área bajo la curva:"""
112 from sklearn.metrics import roc_auc_score
113 auc = roc_auc_score(y_true, y_pred)
114 print('AUC:', auc)
115
116 """ R Score:(R^2 coefficient of determination)"""
117 from sklearn.metrics import r2_score
118 r2 = r2_score(y_true, y_pred)
119 print('R2:', r2)
```

\loss: 0.1637

accuracy: 93.80%

9/9 [=====] - 0s 1ms/step - loss: 0.1386 - accuracy: 0.9382

Exactitud del Entrenamiento 0.9381818175315857

Perdida del Entrenamiento 0.13861769437789917

Matriz de confusion:

[[726 35]

[ 50 560]]

acc: 0.9380014587892049

recall: 0.9180327868852459

precision: 0.9411764705882353

F1: 0.9294605809128631

AUC: 0.9360203356239634

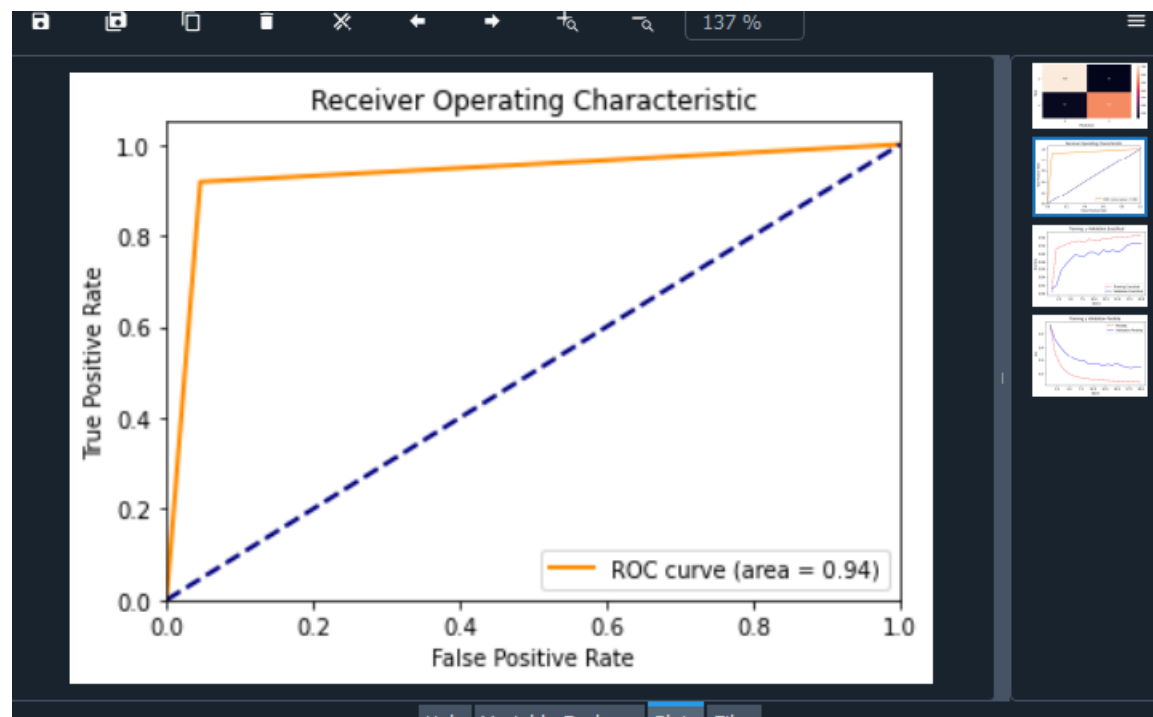
R2: 0.7489606103494615

<Figure size 432x288 with 0 Axes>

In [61]:

## Curva ROC

```
120
121 """Curva ROC"""
122 from sklearn.metrics import roc_curve
123 lw = 2
124 plt.plot(roc_curve(y_true, y_pred)[0], roc_curve(y_true, y_pred)[1],
125          color='darkorange', lw=lw, label='ROC curve (area = %0.2f)' % auc)
126 plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
127 plt.xlim([0.0, 1.0])
128 plt.ylim([0.0, 1.05])
129 plt.xlabel('False Positive Rate')
130 plt.ylabel('True Positive Rate')
131 plt.title('Receiver Operating Characteristic')
132 plt.legend(loc="lower right")
133 plt.show()
134
```





## Gráficas del Training y Test

```
134
135 #-----GRAFICAS-----
136 """Gráficas de Training y Test  epocas /accuracy"""
137
138 acc = modelo.history['accuracy']
139 val_acc = modelo.history['val_accuracy']
140 loss = modelo.history['loss']
141 val_loss = modelo.history['val_loss']
142 epochs = range(1, len(acc)+1, 1)
143 plt.plot ( epochs, acc, 'r--', label='Training Exactitud' )
144 plt.plot ( epochs, val_acc, 'b', label='Validation Exactitud')
145 plt.title ('Training y Validation Exactitud')
146 plt.ylabel('Accuracy')
147 plt.xlabel('epocs')
148 plt.legend()
149 plt.figure()
150
151 """Entrenamiento epocas /perdida"""
152 plt.plot ( epochs, loss, 'r--', label='Perdida' )
153 plt.plot ( epochs, val_loss, 'b', label='Validation Perdida')
154 plt.title ('Training y Validation Perdida ' )
155 plt.ylabel('Loss') # corregir por perdida
156 plt.xlabel('epocs')
157 plt.legend()
158 plt.figure()
159 plt.show()
160
```



