

**El presente análisis de series temporales se basa en el volumen de venta de Diezmillo Choice en el periodo que comprende del 01-01-2015 al 22-09-2020.**



**Presenta:** Carlos Alberto Rábago Figueroa

**Exp. 204200919**

## Descripción de la base de datos

---

La base se compone de 2047 registros.

Las características son la Fecha y el volumen de venta.

## Importando Librerías necesarias

```
In [31]: #Core Keras Libraries
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.layers import LSTM
from keras.layers import Bidirectional

#For data conditioning
from scipy.ndimage import gaussian_filter1d
from scipy.signal import medfilt

# Make results reproducible
from numpy.random import seed
seed(1)
import tensorflow
tensorflow.random.set_seed(1)
```

```
In [32]: # Other essential libraries
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from numpy import array

# Make our plot a bit formal
font = {'family' : 'Calibri',
        'weight' : 'normal',
        'size'   : 10}
plt.rc('font', **font)
```

## Número de entrada de marcas de tiempo y días de entrenamiento

```
In [33]: # Set input number of timestamps and training days

n_timestamp = 10
train_days = 1535 # number of days to train from
testing_days = 512 # number of days to be predicted
n_epochs = 50
filter_on = 1

model_type = 1
```

### Se importa el dataset

```
In [34]: dataset = pd.read_csv("datos.csv")
if filter_on == 1:
    dataset['Volumen'] = medfilt(dataset['Volumen'], 3)
    dataset['Volumen'] = gaussian_filter1d(dataset['Volumen'], 1.2)
```

```
In [35]: dataset.head(1500)
```

Out[35]:

	fecha	Volumen
0	2015-01-02	28.151975
1	2015-01-03	30.689112
2	2015-01-04	28.253833
3	2015-01-05	20.036879
4	2015-01-06	12.873568
...	...	...
1495	2019-03-15	14.985238
1496	2019-03-16	16.266488
1497	2019-03-17	16.629583
1498	2019-03-18	15.196382
1499	2019-03-19	12.556375

### Se establece el conjunto de datos de entrenamiento y de prueba

```
In [36]: # Set number of training and testing data

train_set = dataset[0:train_days].reset_index(drop=True)
test_set = dataset[train_days: train_days+testing_days].reset_index(drop=True)
training_set = train_set.iloc[:, 1:2].values
testing_set = test_set.iloc[:, 1:2].values
```

### Normalización de datos

```
In [37]: # Normalize data first

sc = MinMaxScaler(feature_range = (0, 1))
training_set_scaled = sc.fit_transform(training_set)
testing_set_scaled = sc.fit_transform(testing_set)
```

### Se dividen los datos en el registro de tiempo

```
In [38]: # Split data into n_timestamp

def data_split(sequence, n_timestamp):
    x = []
    y = []
    for i in range(len(sequence)):
```

## Se dividen los datos en el registro de tiempo

```
In [38]: # Split data into n_timestamp

def data_split(sequence, n_timestamp):
    X = []
    y = []
    for i in range(len(sequence)):
        end_ix = i + n_timestamp
        if end_ix > len(sequence)-1:
            break
        # i to end_ix as input
        # end_ix as target output
        seq_x, seq_y = sequence[i:end_ix], sequence[end_ix]
        X.append(seq_x)
        y.append(seq_y)
    return array(X), array(y)

X_train, y_train = data_split(training_set_scaled, n_timestamp)
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
X_test, y_test = data_split(testing_set_scaled, n_timestamp)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)

X_train.shape
```

```
Out[38]: (1525, 10, 1)
```

## Primer configuración red neuronal LSTM normal, activación ReLU, Dropout 0.5

### Configuración de la red neuronal (Normal LSTM normal, Bidireccional y Stacked)

```
In [39]: if model_type == 1:
# Single cell LSTM
model = Sequential()
model.add(LSTM(units = 50, activation='relu', input_shape = (X_train.shape[1], 1)))
model.add(Dense(units = 1))
model.add(Dropout(0.5))
if model_type == 2:
# Stacked LSTM
model = Sequential()
model.add(LSTM(50, activation='relu', return_sequences=True, input_shape=(X_train.shape[1], 1)))
model.add(LSTM(50, activation='relu'))
model.add(Dense(1))
model.add(Dropout(0.5))
if model_type == 3:
# Bidirectional LSTM
model = Sequential()
model.add(Bidirectional(LSTM(50, activation='relu'), input_shape=(X_train.shape[1], 1)))
model.add(Dense(1))
model.add(Dropout(0.5))
```

## Inicia el entrenamiento

In [40]: *# Start training*

```
model.compile(optimizer = 'adam', loss = 'mean_squared_error')
history = model.fit(X_train, y_train, epochs = n_epochs, batch_size = 32)
loss = history.history['loss']
epochs = range(len(loss))

48/48 [=====] - 1s 13ms/step - loss: 0.0501
Epoch 4/50
48/48 [=====] - 1s 13ms/step - loss: 0.0522
Epoch 5/50
48/48 [=====] - 1s 13ms/step - loss: 0.0470
Epoch 6/50
48/48 [=====] - 1s 13ms/step - loss: 0.0532
Epoch 7/50
48/48 [=====] - 1s 14ms/step - loss: 0.0536: 0s - loss:
Epoch 8/50
48/48 [=====] - 1s 14ms/step - loss: 0.0511
Epoch 9/50
48/48 [=====] - 1s 13ms/step - loss: 0.0500
Epoch 10/50
48/48 [=====] - 1s 13ms/step - loss: 0.0512
Epoch 11/50
48/48 [=====] - 1s 14ms/step - loss: 0.0523
Epoch 12/50
48/48 [=====] - 1s 15ms/step - loss: 0.0468
Epoch 13/50
```



### Obtener predicción

```
In [41]: # Get predicted data

y_predicted = model.predict(X_test)

# 'De-normalize' the data

y_predicted_descaled = sc.inverse_transform(y_predicted)
y_train_descaled = sc.inverse_transform(y_train)
y_test_descaled = sc.inverse_transform(y_test)
y_pred = y_predicted.ravel()
y_pred = [round(yx, 2) for yx in y_pred]
y_tested = y_test.ravel()
```

### Mostrar resultados

```
In [42]: # Show results

plt.figure(figsize=(8,7))

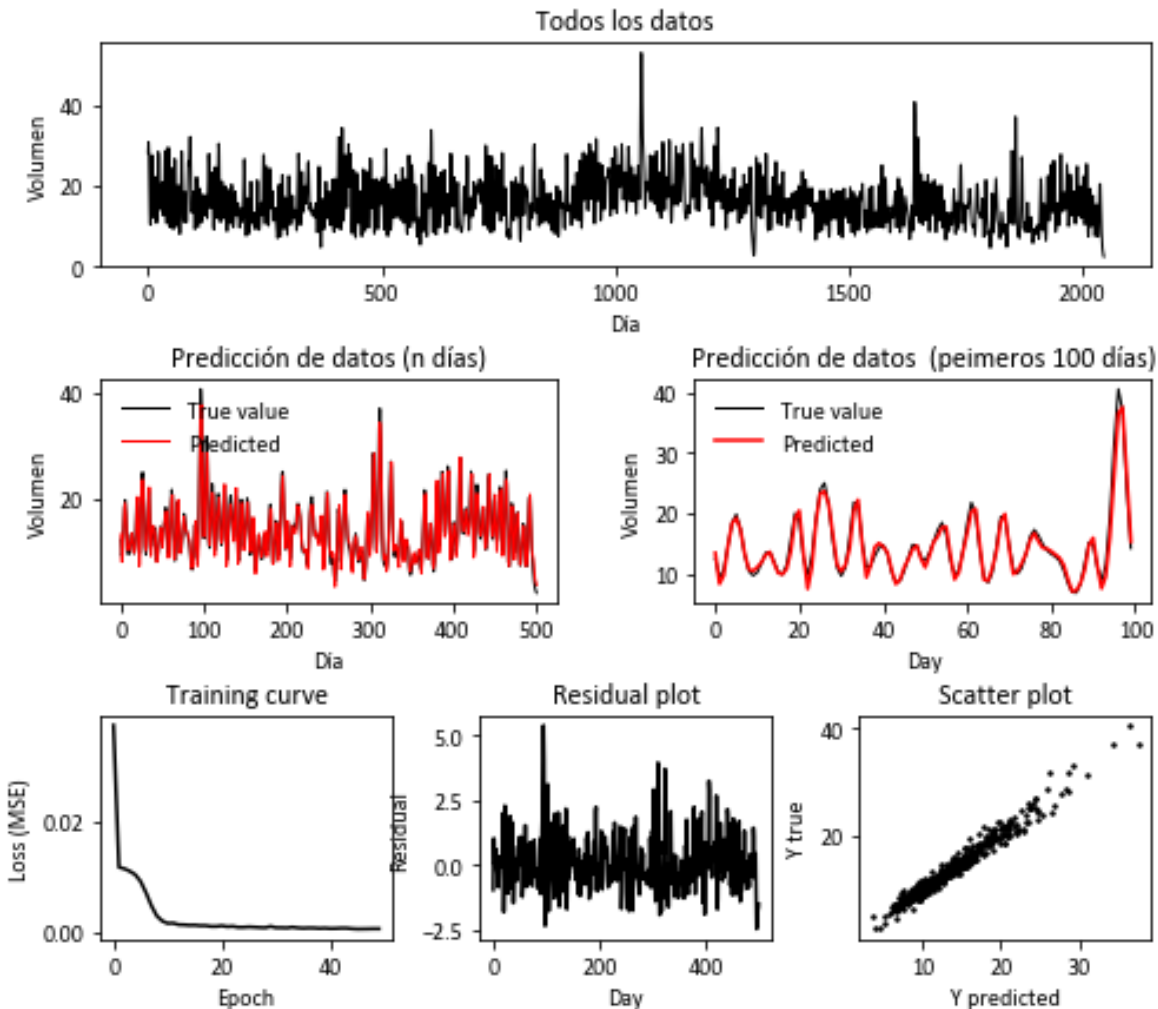
plt.subplot(3, 1, 1)
plt.plot(dataset['Volumen'], color = 'black', linewidth=1, label = 'True value')
plt.ylabel("Volumen")
plt.xlabel("Día")
plt.title("Todos los datos")

plt.subplot(3, 2, 3)
plt.plot(y_test_descaled, color = 'black', linewidth=1, label = 'True value')
plt.plot(y_predicted_descaled, color = 'red', linewidth=1, label = 'Predicted')
plt.legend(frameon=False)
plt.ylabel("Volumen")
plt.xlabel("Día")
plt.title("Predicción de datos (n días)")

plt.subplot(3, 2, 4)
plt.plot(y_test_descaled[0:100], color = 'black', linewidth=1, label = 'True value')
plt.plot(y_predicted_descaled[0:100], color = 'red', label = 'Predicted')
plt.legend(frameon=False)
plt.ylabel("Volumen")
plt.xlabel("Day")
plt.title("Predicción de datos (peimeros 100 días)")

plt.subplot(3, 3, 7)
plt.plot(epochs, loss, color='black')
plt.ylabel("Loss (MSE)")
plt.xlabel("Epoch")
```

## Primer configuración red neuronal LSTM normal, activación ReLU



mse=1.12  
r2=0.96

La base se dividió en 1535 datos para entrenamiento. 512 datos para prueba, numero de iteraciones 50, tipo de modelo 1 (LSTM normal)

Batch Size 32

La grafica Principal muestra la variación en la venta de día 0 a 2000 días con volúmenes de en 1 y 40 kg.

La grafica de comparación del valor verdadero vs el valor predicho se ve muy pareja (0 a 500 días y con volúmenes 1 a 40 kg), la grafica contigua es similar pero a menor escala también se observa muy pareja (0 a 100 días y con volúmenes 1 a 40 kg).

La grafica de curva de error va decreciendo a 0 y baja poco a poco, la grafica de correlación se muestra la tendencia a una línea recta con dispersión hacia el extremo superior derecho.

Tiene un error cuadrático medio bajo lo que indica que el modelo es relativamente bueno, el coeficiente de determinación es cercano a 1.

## Segunda configuración red neuronal LSTM normal, activación ReLU, Dropout 0.5

La base se dividió en 1535 datos para entrenamiento. 512 datos para prueba, numero de iteraciones 50, tipo de modelo 1 (LSTM normal)

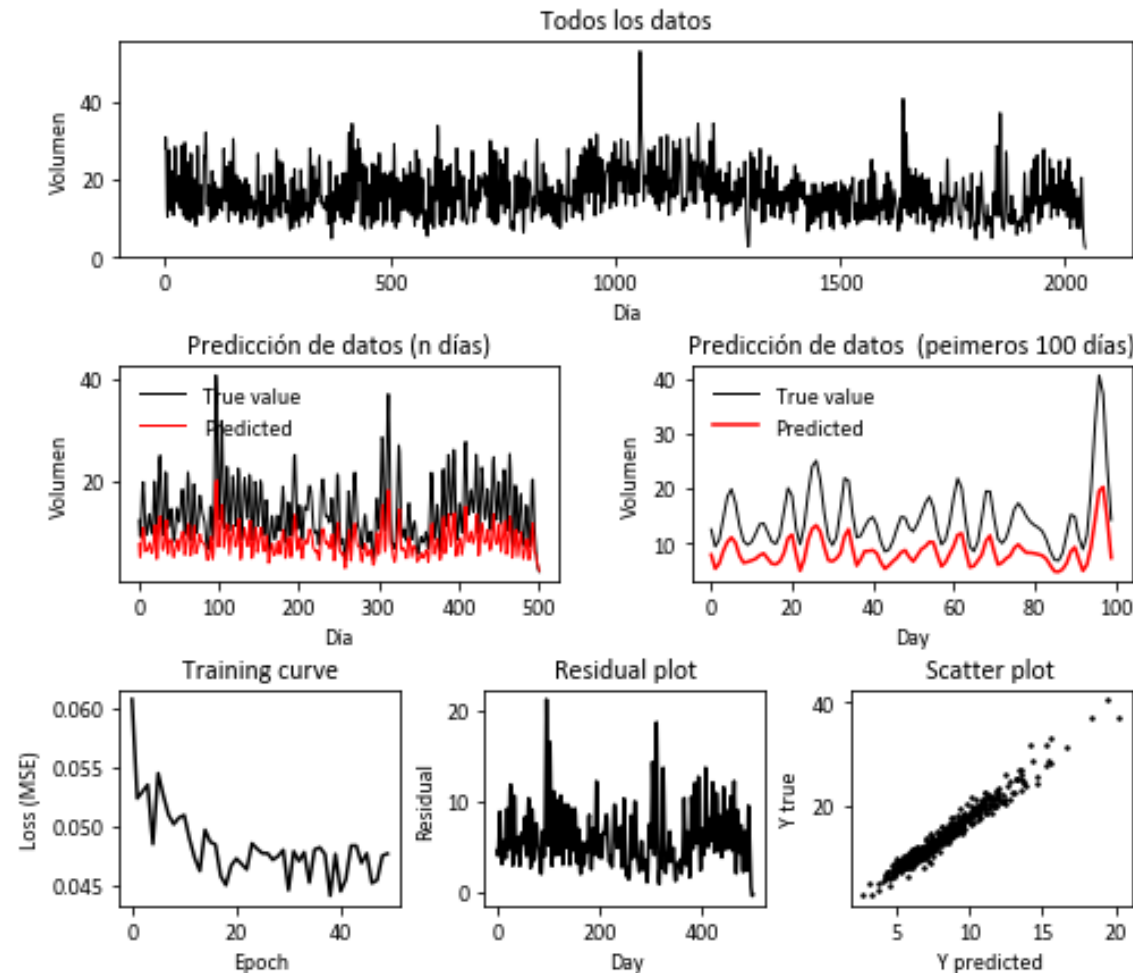
Batch Size 32

La grafica Principal muestra la variación en la venta de día 0 a 2000 días con volúmenes de en 1 y 40 kg.

La grafica de comparación del valor verdadero vs el valor predicho se ve diferencias (0 a 500 días y con volúmenes 1 a 40 kg), la grafica contigua es similar pero a menor escala también se observa también diferencias (0 a 100 días y con volúmenes 1 a 40 kg).

La grafica de curva de error denota variación tiende a subir y bajar, la grafica de correlación se muestra la tendencia a una línea recta con dispersión hacia el extremo superior derecho.

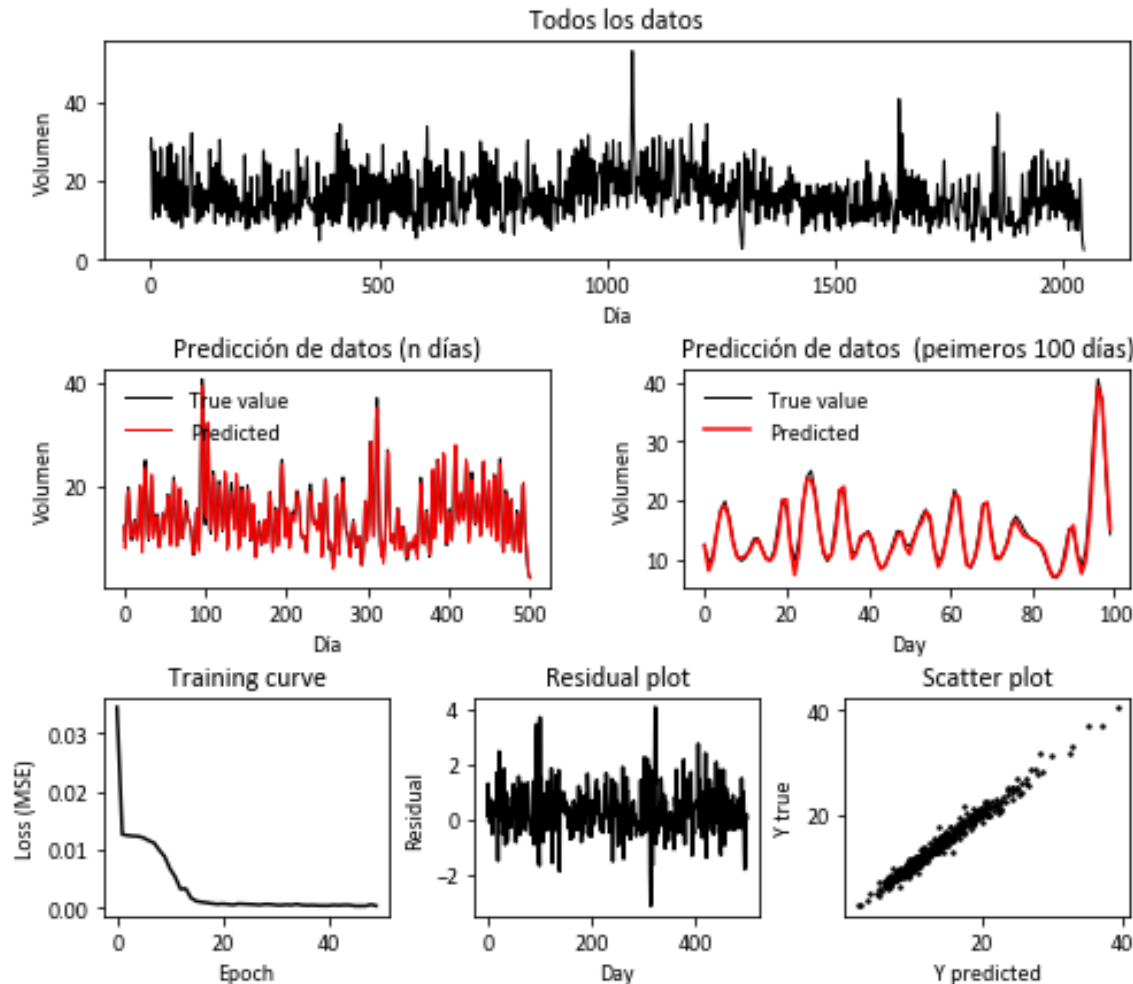
Tiene un error cuadrático medio muy alto lo que indica que el modelo no es bueno, el coeficiente de determinación es cercano a 0.



mse=43.69

r2=-0.47

## Segunda configuración red neuronal LSTM Stacked, activación ReLU



mse=0.81  
r2=0.97

La base se dividió en 1535 datos para entrenamiento. 512 datos para prueba, numero de iteraciones 50, tipo de modelo 2 (LSTM Stacked)

Batch Size 32

La grafica Principal muestra la variación en la venta de día 0 a 2000 días con volúmenes de en 1 y 40 kg.

La grafica de comparación del valor verdadero vs el valor predicho se ve muy pareja (0 a 500 días y con volúmenes 1 a 40 kg), la grafica contigua es similar pero a menor escala también se observa muy pareja (0 a 100 días y con volúmenes 1 a 40 kg).

La grafica de curva de error va decreciendo a 0 y baja poco a poco, la grafica de correlación se muestra la tendencia a una línea recta con dispersión hacia el extremo superior derecho.

Tiene un error cuadrático medio bajo lo que indica que el modelo se debe verificar, el coeficiente de determinación es cercano a 1.

## Segunda configuración red neuronal LSTM Bidirectional, activación ReLU

La base se dividió en 1535 datos para entrenamiento. 512 datos para prueba, numero de iteraciones 50, tipo de modelo 3 (LSTM Bidireccional)

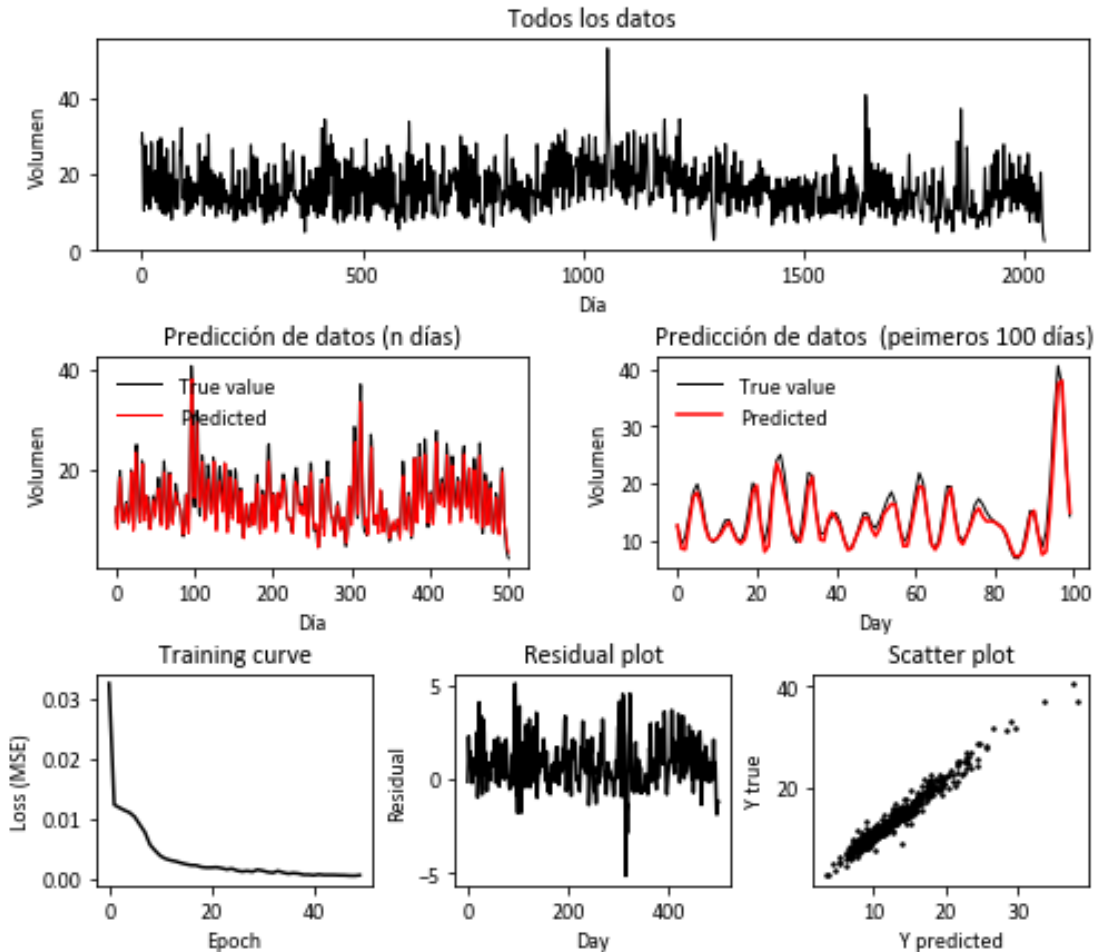
Batch Size 32

La grafica Principal muestra la variación en la venta de día 0 a 2000 días con volúmenes de en 1 y 40 kg.

La grafica de comparación del valor verdadero vs el valor predicho se ve casi parejo (0 a 500 días y con volúmenes 1 a 40 kg), la grafica contigua es similar pero a menor escala también se observa también que casi hay superposición (0 a 100 días y con volúmenes 1 a 40 kg).

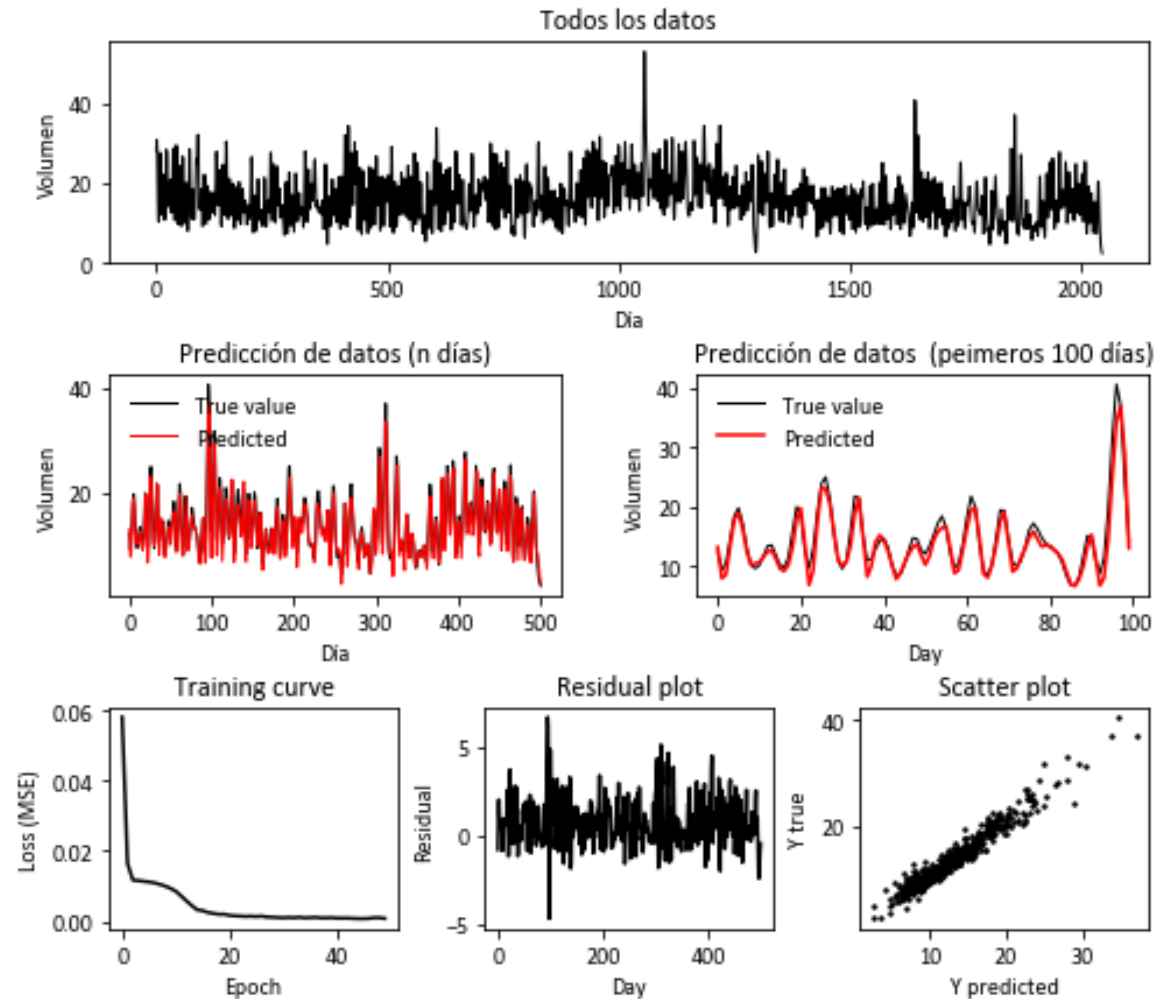
La grafica de curva de error va decreciendo a 0 y baja poco a poco, la grafica de correlación se muestra la tendencia a una línea recta con dispersión hacia el extremo superior derecho.

Tiene un error cuadrático medio es bajo lo que indica que el modelo es relativamente bueno, el coeficiente de determinación es cercano a 1.



mse=2.25  
r2=0.92

## Primer configuración red neuronal LSTM normal, activación ReLU



mse=2.42  
r2=0.92

La base se dividió en 1535 datos para entrenamiento. 512 datos para prueba, numero de iteraciones 50, tipo de modelo 1 (LSTM normal)

Batch Size 64

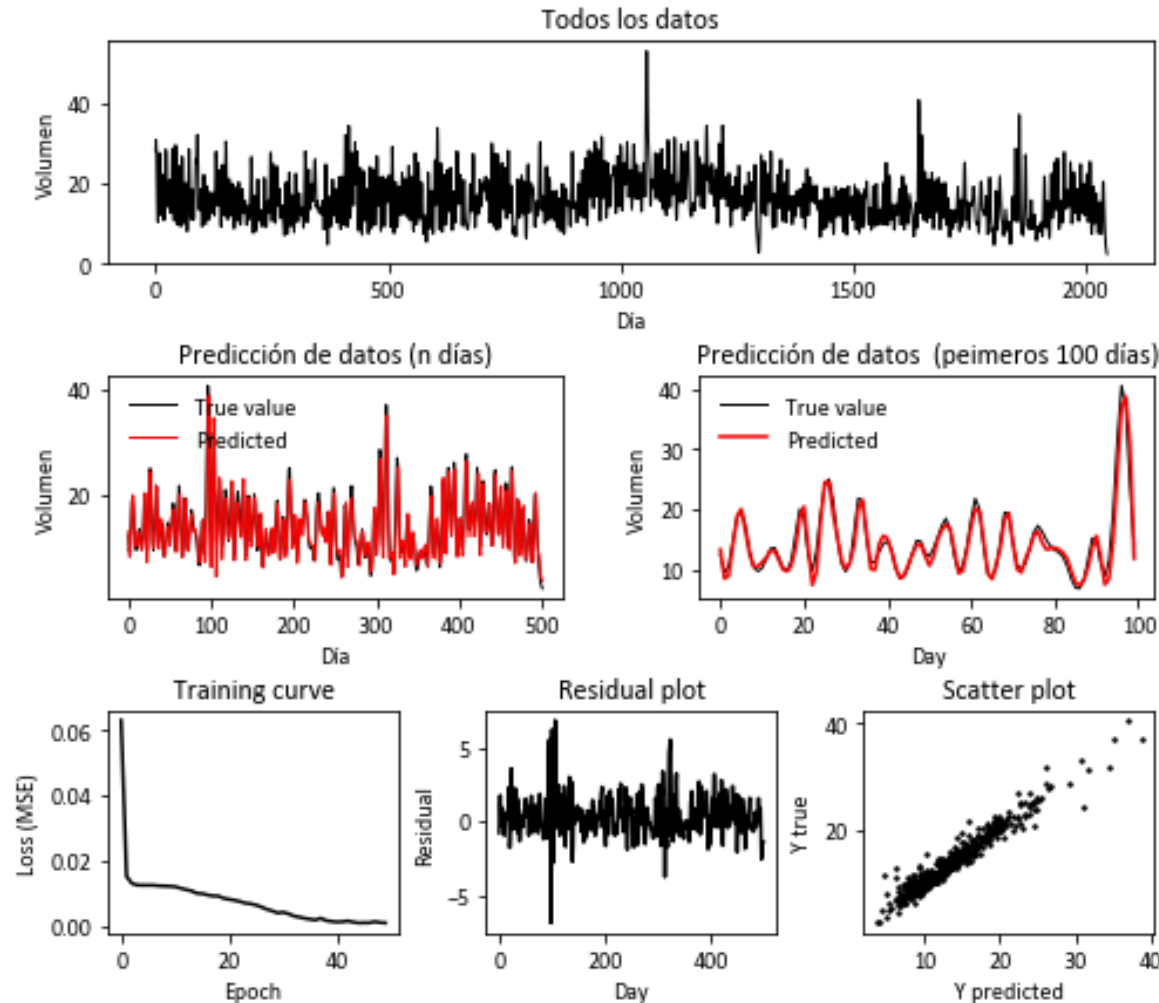
La grafica Principal muestra la variación en la venta de día 0 a 2000 días con volúmenes de en 1 y 40 kg.

La grafica de comparación del valor verdadero vs el valor predicho se ve muy pareja (0 a 500 días y con volúmenes 1 a 40 kg), la grafica contigua es similar pero a menor escala también se observa muy pareja (0 a 100 días y con volúmenes 1 a 40 kg).

La grafica de curva de error va decreciendo a 0 y baja poco a poco, la grafica de correlación se muestra la tendencia a una línea recta con dispersión hacia el extremo superior derecho.

Tiene un error cuadrático medio bajo lo que indica que el modelo es relativamente bueno, el coeficiente de determinación es cercano a 1.

## Segunda configuración red neuronal LSTM Stacked, activación ReLU



mse=2.02  
r2=0.93

La base se dividió en 1535 datos para entrenamiento. 512 datos para prueba, numero de iteraciones 50, tipo de modelo 2 (LSTM Stacked)

Batch Size 64

La grafica Principal muestra la variación en la venta de día 0 a 2000 días con volúmenes de en 1 y 40 kg.

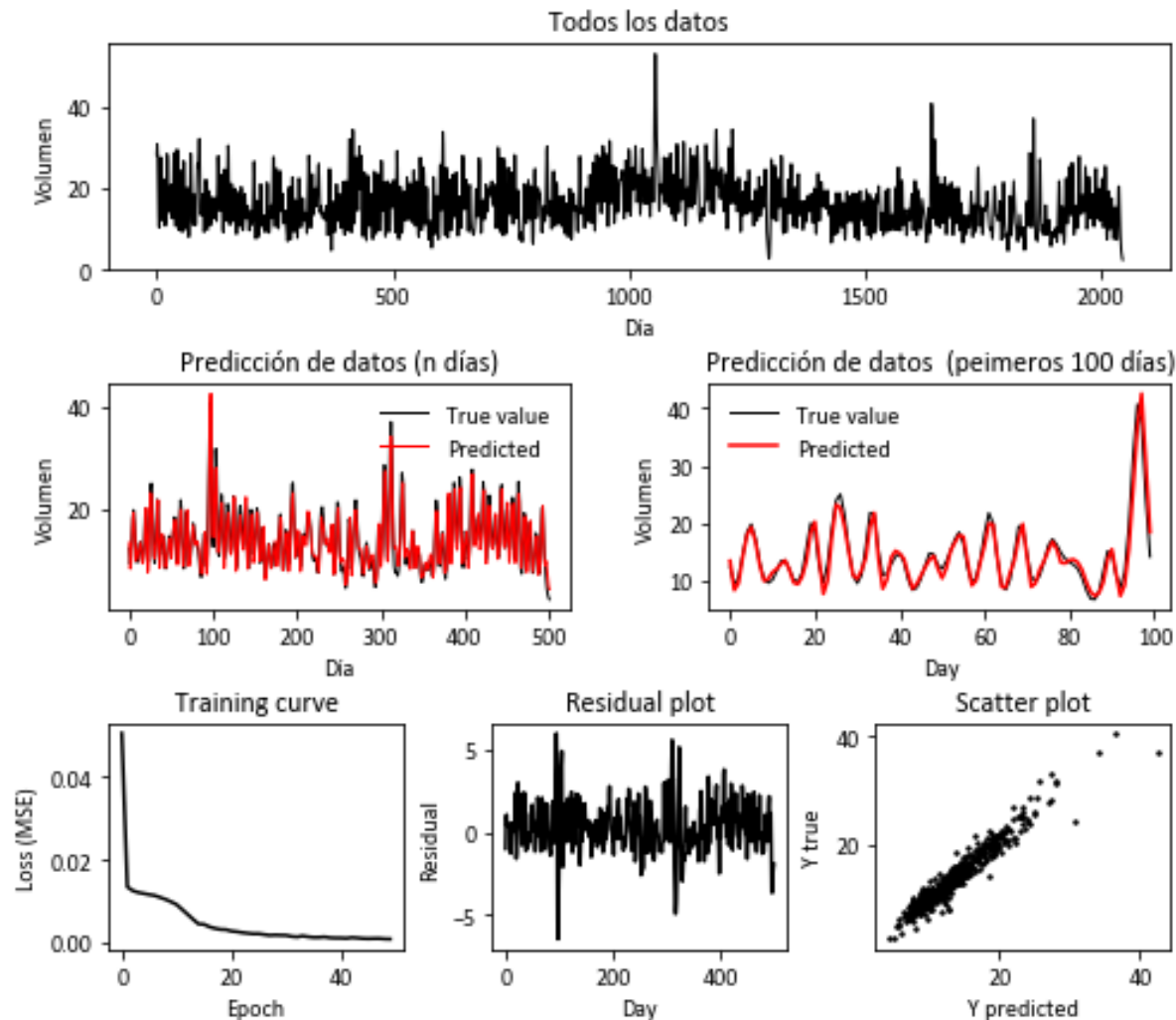
La grafica de comparación del valor verdadero vs el valor predicho se ve muy pareja (0 a 500 días y con volúmenes 1 a 40 kg), la grafica contigua es similar pero a menor escala también se observa muy pareja (0 a 100 días y con volúmenes 1 a 40 kg).

La grafica de curva de error va decreciendo a 0 y baja poco a poco, la grafica de correlación se muestra la tendencia a una línea recta con dispersión hacia el extremo superior derecho.

Tiene un error cuadrático medio bajo lo que indica que el modelo es relativamente bueno, el coeficiente de determinación es cercano a 1.



## Segunda configuración red neuronal LSTM Bidirectional, activación ReLU



mse=2.35  
r2=0.92

La base se dividió en 1535 datos para entrenamiento. 512 datos para prueba, numero de iteraciones 50, tipo de modelo 3 (LSTM Bidireccional)

Batch Size 64

La grafica Principal muestra la variación en la venta de día 0 a 2000 días con volúmenes de en 1 y 40 kg.

La grafica de comparación del valor verdadero vs el valor predicho se ve muy pareja (0 a 500 días y con volúmenes 1 a 40 kg), la grafica contigua es similar pero a menor escala también se observa muy pareja (0 a 100 días y con volúmenes 1 a 40 kg).

La grafica de curva de error va decreciendo a 0 y baja poco a poco, la grafica de correlación se muestra la tendencia a una línea recta con dispersión hacia el extremo superior derecho.

Tiene un error cuadrático medio bajo lo que indica que el modelo es relativamente bueno, el coeficiente de determinación es cercano a 1.