

# SyP 500



Jorge Saenz de Miera Marzo  
Nicolas Vega Muñoz

# ÍNDICE

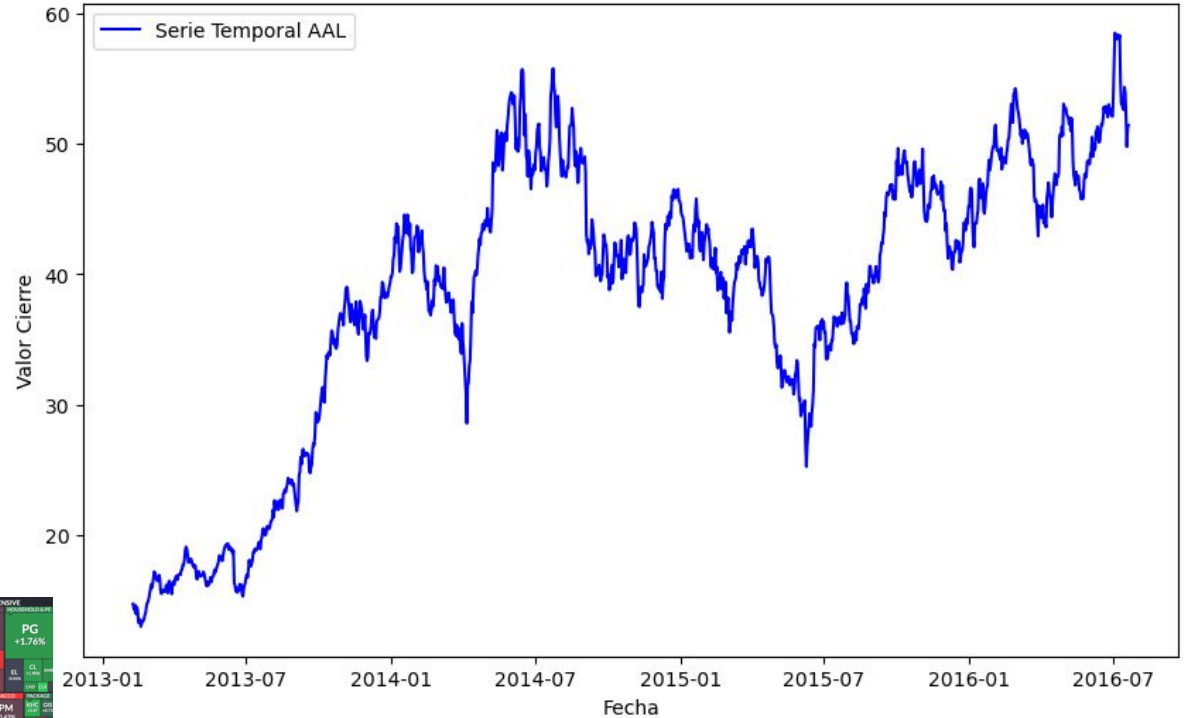
1. Dominio
2. Preprocesamiento
3. Regresión
4. Clasificación
5. Simulación
6. Resultados



# DOMINIO

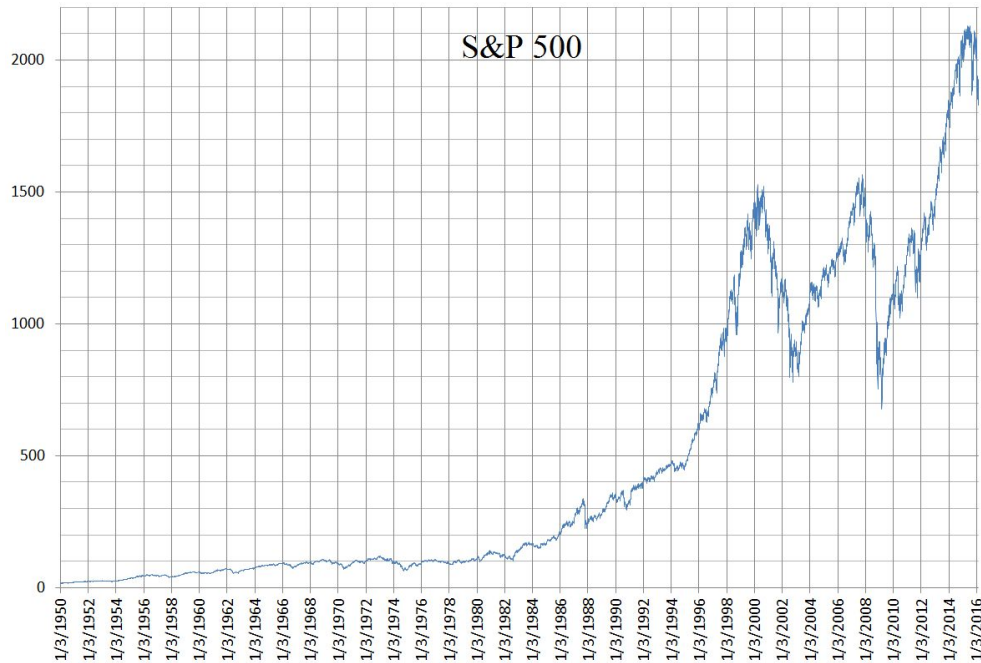
- Bolsa
- Bursátil
- Aleatorio

Serie Temporal AAL

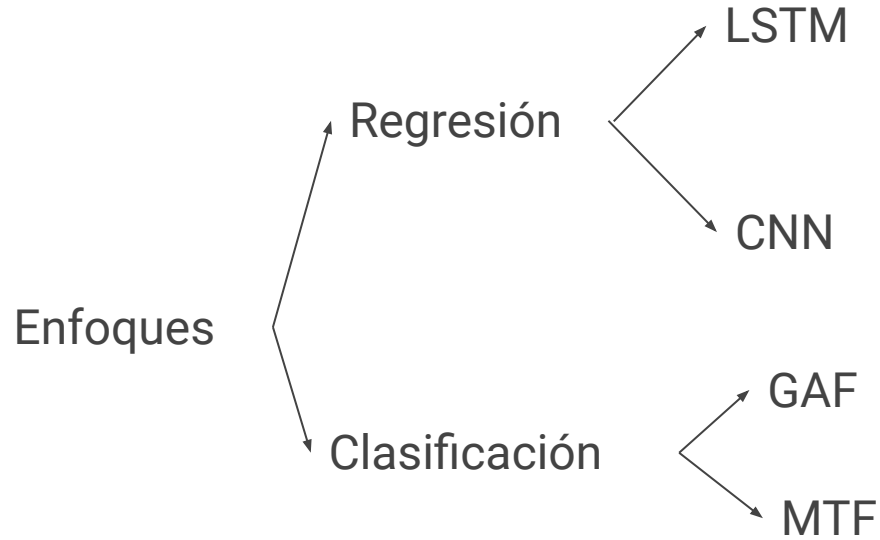


# PREPROCESAMIENTO

1. Series temporales incompletas ❌
2. Normalización ❌
3. Normalización dinámica ✅

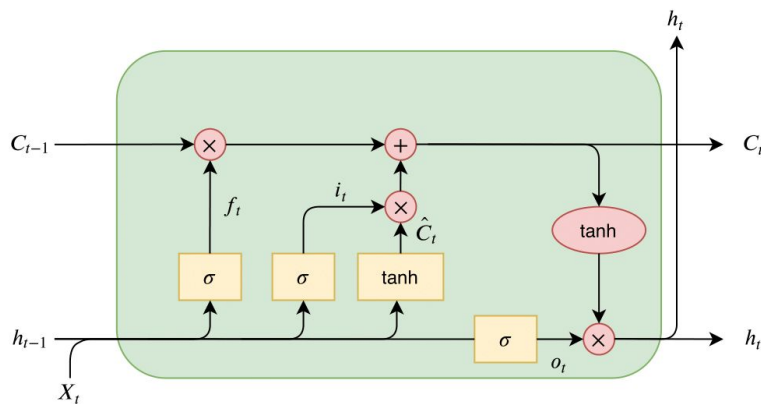


# OBJETIVO: PREDECIR

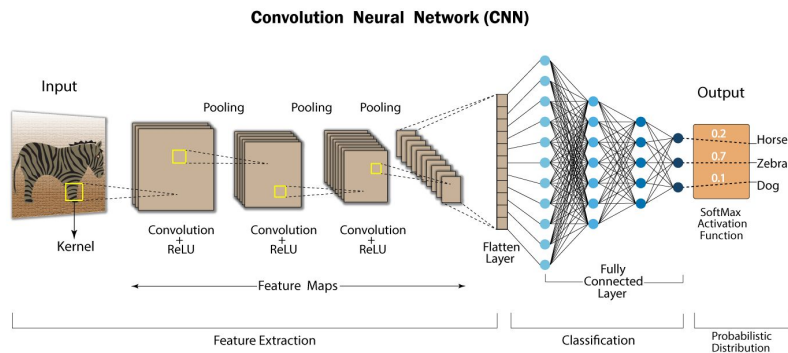


# REGRESIÓN

## 1. LSTM



## 2. CNN



# REGRESIÓN: LSTM

- Sin normalización:

```
lstm = Sequential()  
lstm.add(LSTM(50, input_shape=(X_y_train_regresion[0].shape[1], 1)))  
lstm.add(Dense(1))  
lstm.compile(optimizer='adam', loss='mse')  
lstm.fit(X_y_train_regresion[0], X_y_train_regresion[1], epochs=50, batch_size=32,  
lstm.save('lstm.h5')
```

- Normalizado:

```
lstm_norm = Sequential()  
lstm_norm.add(LSTM(50, input_shape=(X_y_train_regresion_norm[0].shape[1], 1)))  
lstm_norm.add(Dense(1))  
lstm_norm.compile(optimizer='adam', loss='mse')  
lstm_norm.fit(X_y_train_regresion_norm[0], X_y_train_regresion_norm[1], epochs=50, batch_size=32,  
lstm_norm.save('lstm_norm.h5')
```

# EVALUACIÓN: LSTM

- Sin normalización:

## **Errores LSTM:**

Mean Squared Error (MSE): 8225.327508136937  
Root Mean Squared Error (RMSE): 90.69359132891881  
Mean Absolute Error (MAE): 44.70350518466598  
R-squared ( $R^2$ ): 6.388196922213485e-05

- Normalizado:

## **Errores LSTM normalizado:**

Mean Squared Error (MSE): 1.9890293810518618  
Root Mean Squared Error (RMSE): 1.4103295292419646  
Mean Absolute Error (MAE): 1.1559882517338047  
R-squared ( $R^2$ ): 0.00030020314506706836



# REGRESIÓN: CNN

- Sin normalización:

```
cnn = Sequential()  
cnn.add(Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(X_y_train[0].shape[1],)))  
cnn.add(MaxPooling1D(pool_size=2))  
cnn.add(Flatten())  
cnn.add(Dense(50, activation='relu'))  
cnn.add(Dense(1))  
cnn.compile(optimizer='adam', loss='mse')  
cnn.fit(X_y_train_regresion[0], X_y_train_regresion[1], epochs=50, batch_size=32,  
cnn.save('cnn.h5')
```

- Normalizado:

```
cnn_norm = Sequential()  
cnn_norm.add(Conv1D(filters=64, kernel_size=3, activation='relu', input_shape=(X_y_train_regresion_norm[0].shape[1],)))  
cnn_norm.add(MaxPooling1D(pool_size=2))  
cnn_norm.add(Flatten())  
cnn_norm.add(Dense(50, activation='relu'))  
cnn_norm.add(Dense(1))  
cnn_norm.compile(optimizer='adam', loss='mse')  
cnn_norm.fit(X_y_train_regresion_norm[0], X_y_train_regresion_norm[1], epochs=50, batch_size=32,  
cnn_norm.save('cnn_norm.h5')
```

# EVALUACIÓN: CNN

- Sin normalización:

## Errores CNN:

Mean Squared Error (MSE): 8256.562607997897  
Root Mean Squared Error (RMSE): 90.86562940957322  
Mean Absolute Error (MAE): 47.57789044116947  
R-squared ( $R^2$ ): -0.0037333047653334006

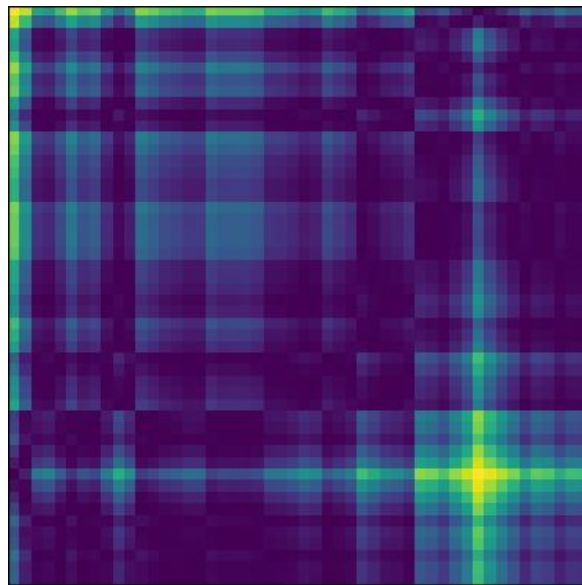
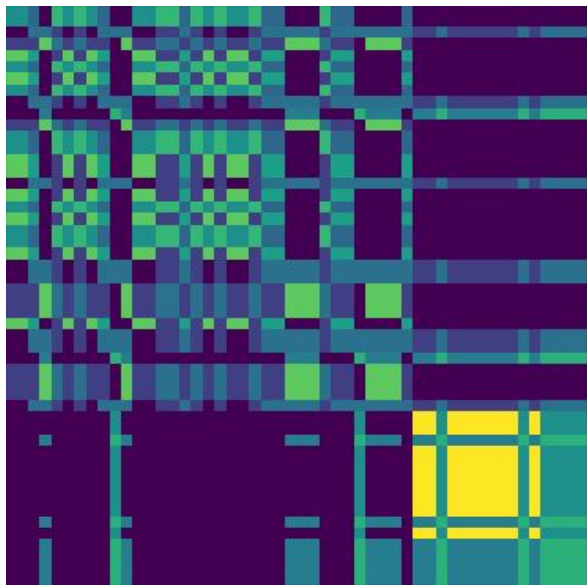
- Normalizado:

## Errores CNN normalizado:

Mean Squared Error (MSE): 3.204628761330171  
Root Mean Squared Error (RMSE): 1.7901476926025324  
Mean Absolute Error (MAE): 1.4167694709673142  
R-squared ( $R^2$ ): -0.6106683753475006

# CLASIFICACIÓN

- Gramian Angular Field (GAF)

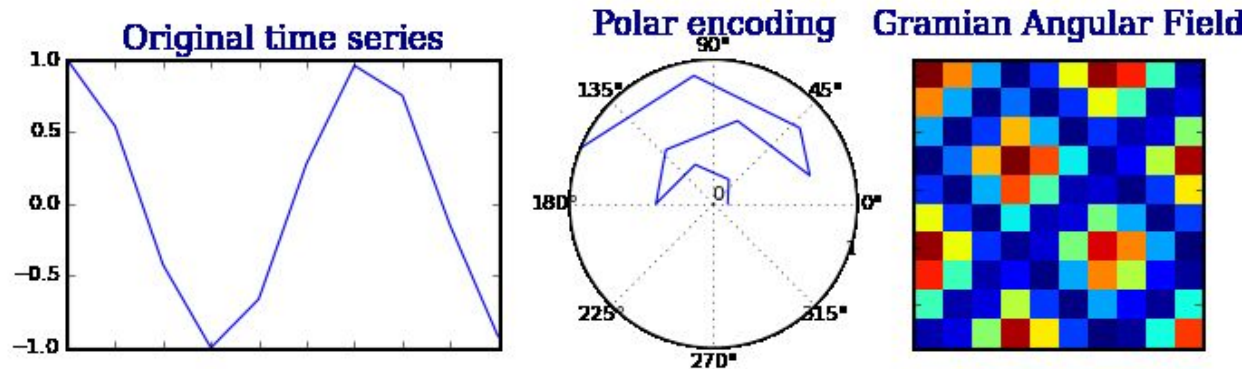


- Markov Transition Field(MTF)

# CLASIFICACIÓN: GAF

1. Normalizar a  $[-1,1]$
2. Coordenadas polares
3. Matriz de Gram

$$G(x_1, \dots, x_n) = \begin{vmatrix} \langle x_1, x_1 \rangle & \langle x_1, x_2 \rangle & \dots & \langle x_1, x_n \rangle \\ \langle x_2, x_1 \rangle & \langle x_2, x_2 \rangle & \dots & \langle x_2, x_n \rangle \\ \vdots & \vdots & & \vdots \\ \langle x_n, x_1 \rangle & \langle x_n, x_2 \rangle & \dots & \langle x_n, x_n \rangle \end{vmatrix}$$



# CLASIFICACIÓN: GAF

- CNN

```
# Crear el modelo CNN
```

```
cnn_gaf = models.Sequential()  
cnn_gaf.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(window_size, window_size, 1)))  
cnn_gaf.add(layers.MaxPooling2D((2, 2)))  
cnn_gaf.add(layers.Conv2D(64, (3, 3), activation='relu'))  
cnn_gaf.add(layers.MaxPooling2D((2, 2)))  
cnn_gaf.add(layers.Conv2D(64, (3, 3), activation='relu'))
```

```
cnn_gaf.add(layers.Flatten())  
cnn_gaf.add(layers.Dense(64, activation='relu'))  
cnn_gaf.add(layers.Dense(1, activation='sigmoid'))
```

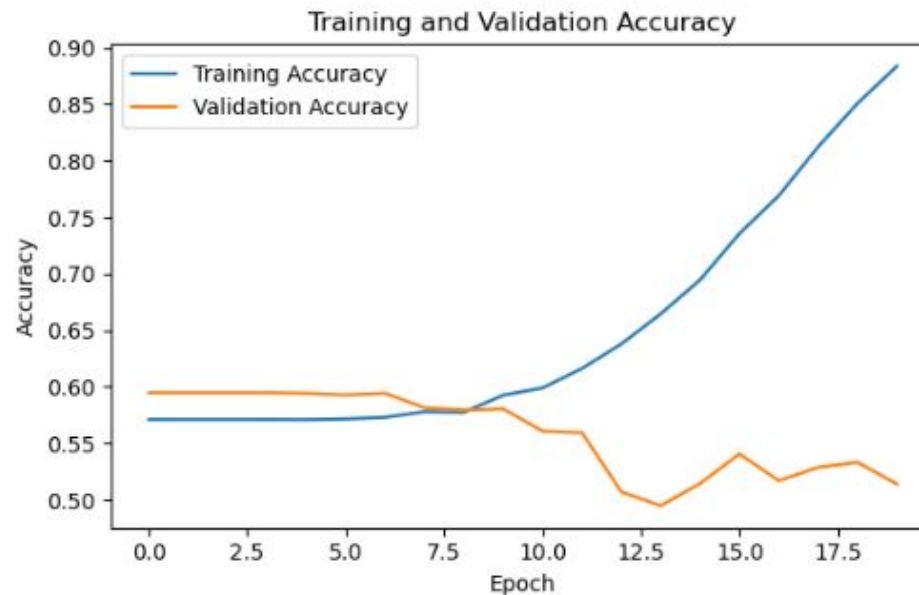
```
# Compilar el modelo
```

```
cnn_gaf.compile(optimizer='adam',  
               loss='binary_crossentropy',  
               metrics=['accuracy'])
```

```
# Entrenar el modelo
```

```
cnn_gaf_history = cnn_gaf.fit(gaf_train, X_y_train_clasificacion[1], epochs=10, batch_size=128, validation_split=0.2)  
cnn_gaf.save('cnn_gaf.h5')
```

# EVALUACIÓN: GAF

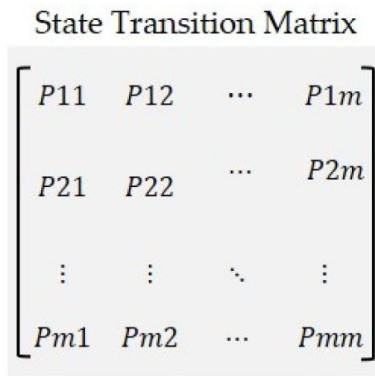
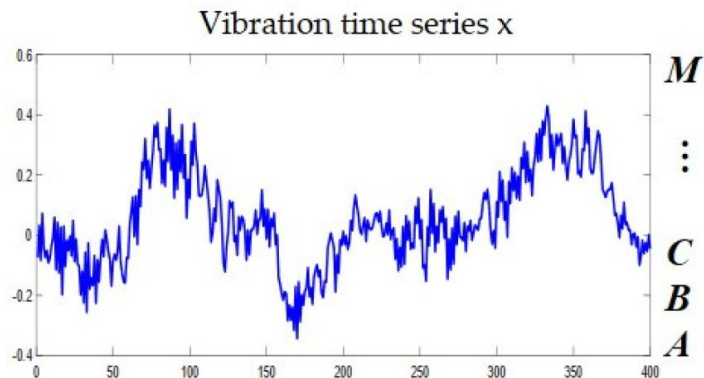


# EVALUACIÓN: GAF

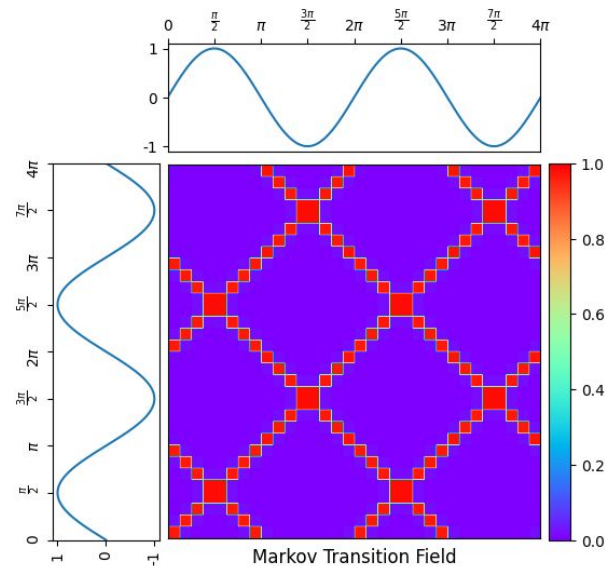
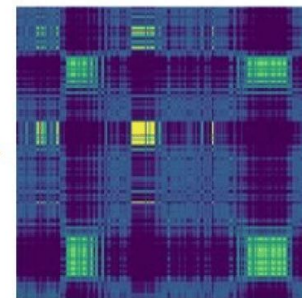
-----Confusion Matrix-----				
	precision	recall	f1-score	support
False	0.58	0.66	0.62	633
True	0.47	0.39	0.42	495
accuracy			0.54	1128
macro avg	0.52	0.52	0.52	1128
weighted avg	0.53	0.54	0.53	1128

# CLASIFICACIÓN: MTF

1. Dividimos en cuantiles
2. Construir matriz ( $A_{ij}=P(st=j|st-1=i)$ )



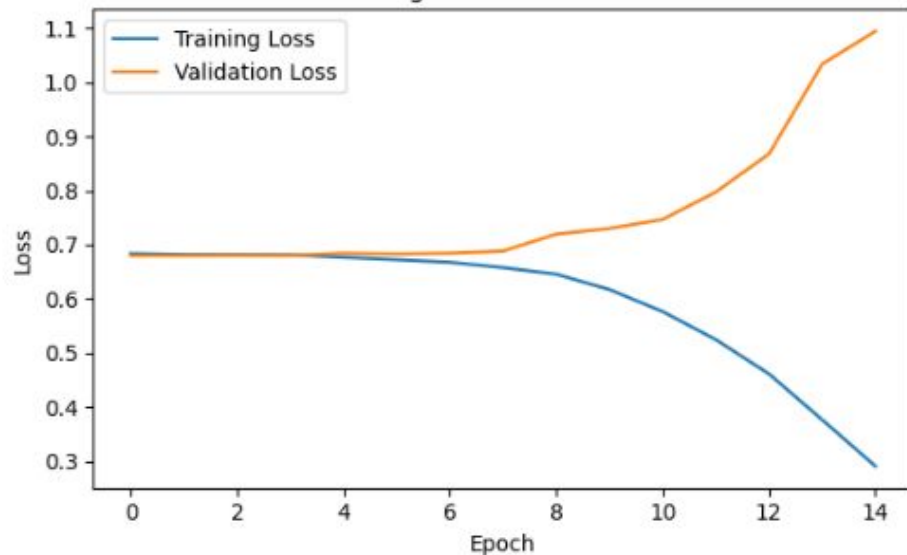
Markov transition field



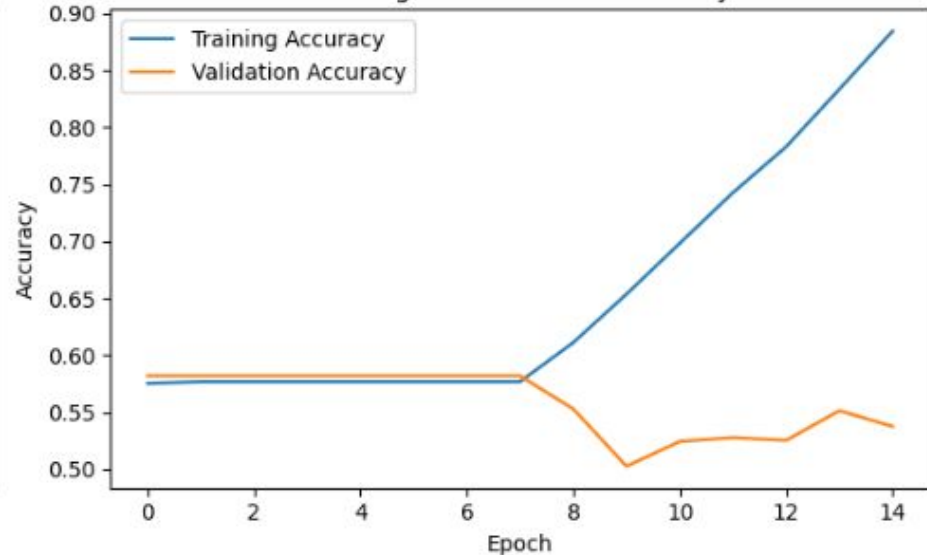


# EVALUACIÓN: MTF

Training and Validation Loss



Training and Validation Accuracy



# EVALUACIÓN: MTF

-----Confusion Matrix-----				
	precision	recall	f1-score	support
False	0.57	0.60	0.59	633
True	0.46	0.43	0.44	495
accuracy			0.53	1128
macro avg	0.52	0.52	0.52	1128
weighted avg	0.52	0.53	0.52	1128

# SIMULACIÓN

	AAL	AAPL	AAP
2017-02-08	45.06	132.040	162.73
2017-02-09	46.30	132.420	164.83
2017-02-10	46.45	132.120	164.05
2017-02-13	47.41	133.290	162.39
2017-02-14	46.57	135.020	164.37
2017-02-15	47.54	135.510	163.88
2017-02-16	46.97	135.345	160.99
2017-02-17	46.91	135.720	161.58
2017-02-21	46.81	136.700	161.15
2017-02-22	46.32	137.110	159.68

PREDECIR

Empresas que van a subir:

['AGN', 'AMG', 'HAS', 'IDXX', 'MTD', 'PCLN', 'TDG']

COMPRAR

Total inversión día 22/02/2017: (\$)  
2939.07

VENDEMOS TODAS AL DIA SIGUIENTE

Total ventas día 23/02/2017: (\$)  
2950.3799999999997

ANALIZAMOS EL BENEFICIO

Beneficio 23/02/2017: (\$)  
11.309999999999949

# SIMULACIÓN

- Aleatoria

El balance final es de 3743.25 \$  
Con una inversión total de 6055074.67 \$  
El día con la mayor inversión fueron 29928.24 \$  
El día con mayor beneficio fueron 390.21 \$,  
El día con mayores pérdidas fueron -908.69 \$,  
Se han comprado un total de 59314 acciones, con una media de 235.37 por día

Evolución del balance

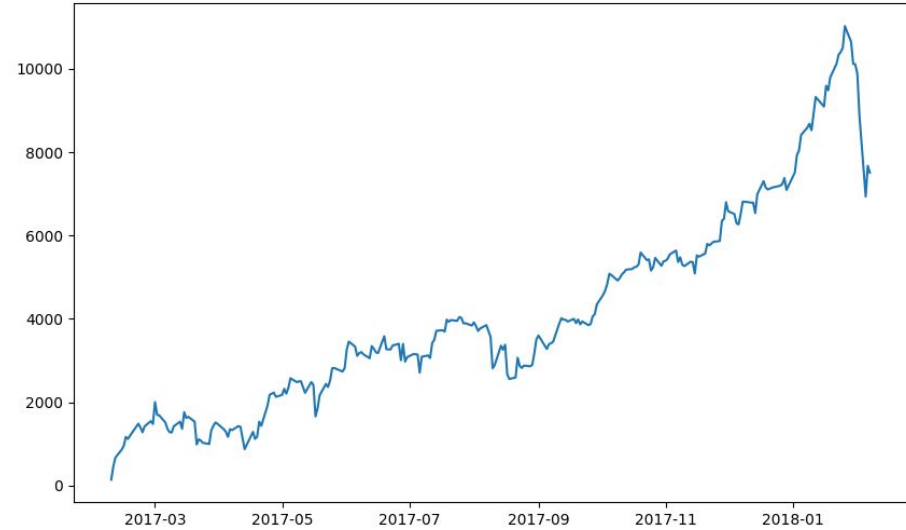


# SIMULACIÓN

- Comprando siempre

El balance final es de 7506.39 \$  
Con una inversión total de 12104427.01 \$  
El día con la mayor inversión fueron 54997.28 \$  
El día con mayor beneficio fueron 739.55 \$,  
El día con mayores pérdidas fueron -1938.37 \$,  
Se han comprado un total de 118440 acciones, con una media de 470.0 por día

Evolución del balance

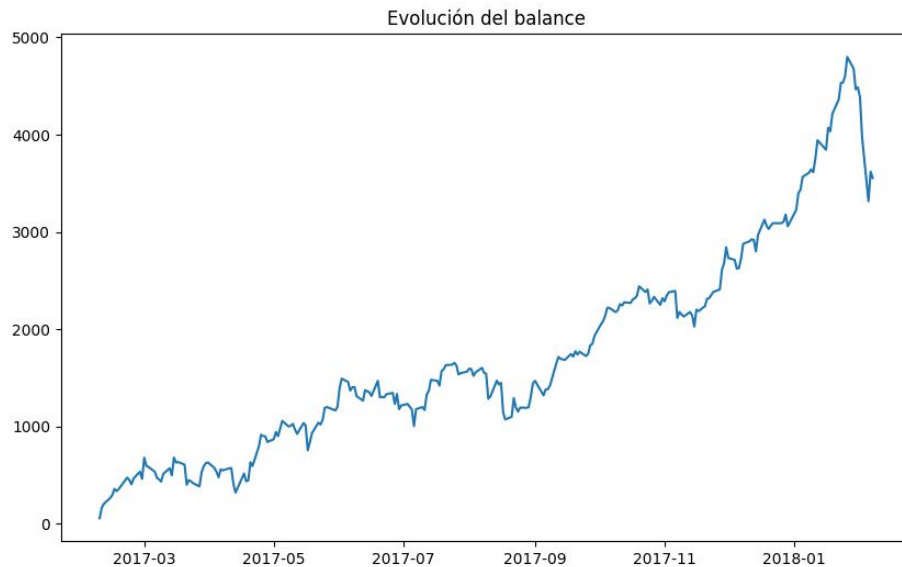


# SIMULACIÓN

- Con Clasificación CNN con MTF

2 horas de ejecución

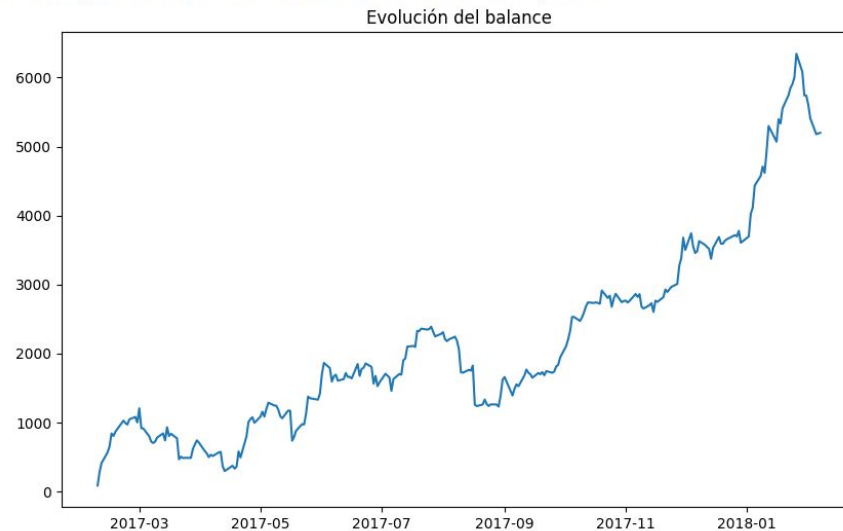
El balance final es de 3553.23 \$  
Con una inversión total de 4974887.41 \$  
El día con la mayor inversión fueron 28574.44 \$  
El día con mayor beneficio fueron 303.33 \$,  
El día con mayores pérdidas fueron -654.45 \$,  
Se han comprado un total de 48561 acciones, con una media de 192.7 por día



# SIMULACIÓN

El balance final es de 5198.2 \$  
Con una inversión total de 7020170.01 \$  
El día con la mayor inversión fueron 44915.74 \$  
El día con mayor beneficio fueron 363.67 \$,  
El día con mayores pérdidas fueron -568.02 \$,  
Se han comprado un total de 66206 acciones, con una media de 262.72 por día

- Con Reglas sencillas



# CONCLUSIÓN







# GRACIAS POR SU ATENCION

Jorge Saenz de Miera Marzo  
Nicolas Vega Muñoz