FAMAF Facultad de Matemática, Astronomía, Física y Computación

UNC Universidad Nacional de Córdoba

DIPLOMATURA DE DATOS 2024

BOLETÍN OFICIAL

MENTOR: EMMANUEL TASSONE

# Trabajo Práctico 3 - Aprendizaje Supervisado y/o No Supervisado

**Una vez que entendemos nuestros dataset, queremos identificar patrones, predecir movimientos y validar estrategias de mercado. Estas competencias son esenciales para la toma de decisiones informadas en finanzas y altamente valoradas en el mercado laboral.**

---

## Actividades

1. ***Lagging* en series temporales**
   - Cargar el dataset y rellenar los valores faltantes con el método `fillna` de la librería `pandas`.
   - Crear 30 columnas *laggeadas* con la función `shift`. Luego hacer un *drop* y quedarse con las columnas *'close'* + las columnas *laggeadas.*
   - Dividir el dataset restante en un conjunto de entrenamiento y un conjunto de testeo.
   - Entrenar los modelos `LinearRegression`, `RandomForestRegressor` y SVR (*Support Vector Machine*) para predecir el precio de cierre del conjunto de testeo.
   - Evaluar el rendimientos de los modelos utilizando las métricas MAE (*Mean Absolute Error*) y RMSE (*Root Mean Squared Error*).
   - ¿Qué puede decir sobre los modelos y sus prediciones?¿Qué puede decir sobre el mejor modelo?¿Son resultados confiables?
2. **Modelos ARIMA** (*Auto-Regressive Integrated Moving Average*)
   - Explicar el modelo ARIMA.¿Cuantas componentes tiene el modelo? ¿Como actúa cada componente sobre la serie temporal?¿Para qué sirve hacer diferenciación en este contexto?¿Qué representan los parámetros (p,d,q)?
   - ¿Esperaría tener predicciones confiables para la acción TSLA con este modelo? ¿Por qué? ¿Qué tipo de series temporales son ideales para un modelo ARIMA? Demuestre su afirmación por medio de un test de Dickey-Fuller aumentada.
   - Entrene un modelo ARIMA con la serie temporal de precios de cierre y haga una predicción sobre el conjunto de testeo. Relacione los resultados con los dicho previamente.

3. **Modelos con Redes Neuronales : MLP** (*Multilayer Perceptron*)
    - Uno de los modelos más elementales y primitivos de redes neuronales son los perceptrones multicapa. Utilice este modelo para predecir los precios de cierre de la acción de tesla en el conjunto de testeo.
    - ¿Qué numero de neuronas y capas permiten realizar las mejores predicciones?
    - Realice una validación cruzada utilizando la clase `TimeSeriesSplit` con `n_splits=5`. ¿Puede decir que el modelo es sesgado?
4. **Modelos con Redes Neuronales: LSTM** (*Long Short Term Memory*)
    - Normalice la serie de precios de cierre utilizando el MinMaxScaler.
    - Convierta la serie temporal en una matriz de datos utilizando subintervalos de la serie temporal y un valor ventana (Ej. `seq_length=60`) para crear estos subintervalos.
    - Construya una red neuronal utilizando la librería `keras`. La red debe estar compuesta por la siguiente secuencia: `LSTM(50, return_sequences=True)|LSTM(50, return_sequences=False)|Dense(1)`.
    - Graficar la evolución del error cuadrático medio con las épocas.¿Qué puede decir sobre este?
    - Graficar la predicción en el conjunto de testeo recuperando los valores no normalizados de precios de cierre.

## Características que debe cumplir el entregable

- El proyecto debe ser escrito en un *jupyter notebook* (.ipynb), siguiendo las convenciones PEP8 (https://peps.python.org/pep-0008/).
- El Notebook debe ser claro y estar bien organizado: debe contar con un índice, apartados, código fácil de leer, probado y comentado (no abusar de los comentarios).
- El archivo .ipynb debe compartirse a través de un Google Collab con permisos de edición habilitados al mentor. A su vez, una copia del archivo debe enviarse a la casilla de correo emmanuel.tassone@unc.edu.ar
- Tener en cuenta que el archivo entregable debe contener solo las resoluciones pedidas, dejando de lado análisis adicionales que se hayan hecho.
- Es importante que luego de cada código y gráfico **haya una conclusión o interpretación** de lo obtenido.