

Semana 3 – Predicción en bases de datos.

Ejemplo de mi github:

https://github.com/mayraberrones94/FCFM/blob/master/Semana_3_Prediccion.ipynb

Ejemplos para tener en cuenta:

Market basket análisis:

<https://www.kaggle.com/code/nandinibagga/apriori-algorithm>

<https://www.kaggle.com/code/benroshan/market-basket-analysis>

Series de tiempo:

<https://www.kaggle.com/code/jagangupta/time-series-basics-exploring-traditional-ts>

<https://www.kaggle.com/code/dimitreoliveira/deep-learning-for-time-series-forecasting>

<https://www.kaggle.com/code/neelkudu28/covid-19-visualizations-predictions-forecasting>

Suport vector machine:

<https://www.kaggle.com/code/faressayah/support-vector-machine-pca-tutorial-for-beginner>

Arboles de decisión:

<https://www.kaggle.com/code/faressayah/decision-trees-random-forest-for-beginners>

<https://www.kaggle.com/code/dmilla/introduction-to-decision-trees-titanic-dataset>

Reglas de Asociación:

<https://www.kaggle.com/code/anurag2405/association-rules>

Visualización:

<https://www.kaggle.com/code/kanncaa1/statistical-learning-tutorial-for-beginners>

<https://matplotlib.org/stable/gallery/index.html>

<https://www.python-graph-gallery.com/seaborn/>

<https://plotly.com/python/>

Análisis de datos:

<https://www.kaggle.com/code/subinium/storytelling-with-data-netflix-ver>

<https://www.kaggle.com/code/nayansakhiya/heart-fail-analysis-and-quick-prediction>

<https://www.kaggle.com/code/shwetagoval4/spotify-song-eda>

<https://www.kaggle.com/code/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets>

<https://www.kaggle.com/code/ruchi798/movies-and-tv-shows-eda>

Chatbot example:

<https://towardsdatascience.com/how-to-create-a-chatbot-with-python-deep-learning-in-less-than-an-hour-56a063bdfc44>

Tarea 3: Utilizando una base de datos de interés (pueden incluso seguir usando la que han ido modificando en las tareas anteriores) y utilizando las técnicas mencionadas en clase, realizar una visualización, predicción o clasificación de sus datos. Al finalizar el modelo, establecer cuál es la métrica adecuada para evaluarlo, y mostrar sus resultados.

El trabajo será entregado de la siguiente manera:

- Archivo en formato PDF en la plataforma de Teams.
- Archivo PDF, documento de código e imágenes adjuntas, en su repositorio personal de GitHub.

Recuerden que en el archivo PDF deben de agregar en las referencias el URL de su GitHub personal y el URL de la página de donde sacaron su base de datos.

Se evaluará:

- En los resultados que tengan es necesario una discusión y análisis apropiado de cada grafica o tabla que generen.
- En caso de utilizar un modelo, explicar correctamente porque se utilizo ese modelo, sus ventajas y desventajas.
- Agregar que métricas podemos usar para evaluar el modelo y discutir los resultados.

NOTA: Recuerden que la próxima semana empezamos a discutir el tema del mini proyecto, por lo que les pido que ya tengan en mente con quien les gustaría trabajar, y que base de datos o técnica les gustaría usar.

Puntuación:

Las tareas individuales pueden llegar a tener hasta 6 puntos.

6 puntos - Excelente: Sobrepasa la expectativa del trabajo. Se agregaron elementos de interés a la tarea, como orientar el trabajo con elementos que sirvan para su proyecto de tesina.

5 – 4 puntos – Bueno: Cumple con lo que se espera de la tarea. Baja un poco si se tiene faltas de ortografía o inconsistencias en la discusión de resultados.

3 – 2 puntos – Regular: Faltan puntos que se comentaron para la tarea. Tiene algo que puede ser rescatable, pero falta mucho contenido.

1 punto – Malo: Se intento hacer algo, pero al final no tiene mucho sentido.

0 puntos: No entrego la tarea.

NOTA: Recuerden que la puntuación se evalúa a la mitad de los puntos que pudieron haber obtenido si lo entregan a destiempo por una semana. Después de la primera semana todos los trabajos se evalúan a 0.