Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Tecnicatura en Inteligencia Artificial

Aprendizaje Automático 1

Trabajo Práctico: Predicción de lluvia en Australia.

Objetivos

Familiarizarse con la biblioteca scikit-learn y las herramientas que brinda para el pre-procesamiento de datos, la implementación de modelos y la evaluación de métricas, con TensorFlow para el entrenamiento de redes neuronales y con streamlit para la puesta en producción del modelo seleccionado como el más adecuado, entre otras.

Dataset

El dataset se llama weatherAUS.csv y contiene información climática de Australia de los últimos diez años, incluyendo si para el día siguiente llovió o no y la cantidad de lluvia en las columnas '**RainTomorrow**' y '**RainfallTomorrow**'. El objetivo es la predicción de estas dos variables en función del resto de las características que se consideren adecuadas.

Tiene una columna 'Location' que indica la ciudad y el objetivo es predecir la condición de lluvia en las ciudades de **Adelaide, Canberra, Cobar, Dartmoor, Melbourne, MelbourneAirport, MountGambier, Sydney y SydneyAirpor**t. Pueden considerarse como una única ubicación. **Descartar el resto de los datos.**

Para todos los ítems, incorporar una cantidad de texto adecuado en forma de comentarios, ya sea para la comprensión del código (usualmente una línea de comentario por cada celda) como para explicar las decisiones tomadas a lo largo del trabajo (por ejemplo, la justificación de la imputación de valores faltantes, la elección de las métricas adecuadas, entre otros). Mantener la coherencia con los comentarios.

Consignas

1. Armar grupos de hasta dos personas para la realización del trabajo práctico. Dar aviso al cuerpo docente del equipo. En caso de no tener compañero, informar al cuerpo docente. **Se recomienda que al menos un integrante haya aprobado Fundamentos de Ciencias de Datos.**

2. Crear un repositorio que se llame "AA1-TUIA-Apellido1-Apellido2" en GitHub.

3. Realizar un análisis descriptivo, que ayude a la comprensión del problema, de cada una de las variables involucradas en el problema detallando características, comportamiento y rango de variación.
   Debe incluir:
   - Análisis y decisión sobre datos faltantes.
   - Visualización de datos (por ejemplo histogramas, scatterplots entre variables, diagramas de caja)
   - ¿Está *balanceado* el dataset? ¿Por qué cree que hacemos esta pregunta?
   - Codificación de variables categóricas (si se van a utilizar para predicción).
   - Matriz de correlación de variables.
   - Estandarización de datos.

- Validación cruzada train - test. Realizar una división del conjunto de datos en conjuntos de entrenamiento y prueba (y si se quiere, se puede incluir validación, que luego será útil) **en el MOMENTO donde usted lo crea adecuado.**

4. Implementar la solución del problema de regresión con regresión lineal múltiple.
    - Probar con el método **LinearRegression**.
    - Probar con métodos de **gradiente descendiente**. ¿Algún cambio?
    - Probar con métodos de regularización (**Lasso, Ridge, Elastic Net**).
    - Obtener las **métricas adecuadas** (entre R2 Score, MSE, RMSE, MAE, MAPE, elegir) tanto para entrenamiento como para prueba. **¿Por qué para ambos conjuntos?**
    - **¿Creen que han conseguido un buen fitting?**

5. Implementar la solución del problema de clasificación con regresión logística.
    - Obtener las **métricas adecuadas** (entre Accuracy, precision, recall, F1 Score, entre otras, ¡investiguen adicionales!). Graficar matrices de confusión para cada modelo. Analizar “falsos negativos” y “falsos positivos”, ¿qué significa cada uno?
    - Trazar curvas ROC para cada modelo entrenado (donde hayan utilizado distintos conjuntos de entrenamiento o distintos hiperparámetros). Comente cuáles serían los umbrales adecuados a utilizar; ¿cómo podría calcular el mejor umbral? ¿Es 0.5 el mejor?
    - **¿Creen que han conseguido un buen fitting?**

6. Implementar un modelo base para clasificación y uno para regresión.

7. Optimizar la selección de hiperparámetros.
    - Probar validación cruzada k-folds, si corresponde.
    - Utilizar grid search, random search u optuna. **Justificar su uso. Justificar los hiperparámetros que se están optimizando.**

8. Implementar explicabilidad del modelo.
    - Utilizar SHAP o similar. Implementar al menos dos gráficas a nivel local y dos gráficas a nivel global. **¡Escribir lo que se observa!**
    - ¿Cuáles son las variables más importantes? ¿Cuáles son las menos? ¿Coinciden en ambos modelos (regresión/clasificación)?

9. Implementar las soluciones con una red neuronal.
    - Obtener las métricas adecuadas.
    - Repetir los pasos 7 y 8 para las redes neuronales. ¿Qué diferencias hay con los modelos de regresión lineal y logística?

10. Comparación de modelos.
    - Incluyan en su análisis una comparación de modelos: de todos los modelos de regresión, ¿cuál es el mejor? **Escoger una métrica adecuada para poder compararlos**. Lo mismo con los de clasificación.

11. MLOps
    - Realizar un script app.py donde se utilice streamlit para la puesta en producción, donde el modelo para predecir debe ser el que se elija en el ítem 10.
    - Se valorará incorporación de pipeline con clases y funciones que se encarguen de hacer las transformaciones de datos, aunque no es necesario.
    - El script debe permitir incorporar datos en un frontend de streamlit y debe mostrar la predicción para dichos valores. Se valora incorporar manejo de errores. Se valora el trabajo realizado para mejorar el frontend de la aplicación; sin embargo es suficiente con una interfaz que sea útil.
    - A su vez, se especifica el requerimiento de un script que tenga funciones para tomar datos de entrada (que estén en el mismo formato del dataset)

12. Escribir una conclusión del trabajo.

13. **Preparar una defensa del trabajo práctico**. La presentación de forma oral se hace sobre el notebook de trabajo y la aplicación de streamlit, en un total de 20 minutos, donde deben destacar lo que consideren más relevante, ya que no es tiempo suficiente para mostrar absolutamente todo. **¡Planificar la gestión de tiempos es parte de la calidad de la presentación del trabajo realizado**! Se recomienda previa práctica.

Entregas parciales y condiciones
Las entregas parciales se realizan mediante GitHub (suben el código y nos devuelven el link del repositorio).

El repositorio debe llamarse “AA1-TUIA-Apellido1-Apellido2” **sin excepciones. No crear carpetas dentro que contengan alguno de los entregables.**

Respetar los siguientes nombres:

Notebook de trabajo: TP-integrador-AA1.ipynb
Script con función para predecir a partir de datos tipo-dataset: predict.py
Script que corre aplicación en Streamlit: app.py

Se deben hacer commits con el asunto “Entrega hasta ítem x” (se pueden hacer commits parciales -de hecho se recomienda-).

Hasta el 22/03: ítem 1. Aviso por correo.
Hasta el 19/04: ítems 2, 3 y 4. Enviar link al repositorio con el notebook por mail. Un correo electrónico por grupo. Poner en copia a todos los integrantes del grupo.
Hasta el 24/05: ítem 5, 6, 7 y 8. Enviar link al repositorio con el notebook por mail. Un correo electrónico por grupo. Poner en copia a todos los integrantes del grupo.
Hasta el 08/06: ítem 9 y 10. Enviar link al repositorio con el notebook por mail. Un correo electrónico por grupo. Poner en copia a todos los integrantes del grupo.

Hasta el 21/06: ítems 11 y 12. Enviar link al repositorio con el notebook, app.py y predict.py por mail. Un correo electrónico por grupo. Poner en copia a todos los integrantes del grupo.
Cada entrega puede demorarse hasta dos días después de la fecha pactada, **con disminución de la nota final del trabajo práctico.**

No se aceptan entregas finales con fecha posterior al 23/06/24. En caso de no tener todos los ítems entregados para esta fecha, la condición es automáticamente de libre.

La defensa de los TP se hará desde el lunes 24 de junio, en horarios de clase, separados por turnos que la cátedra asignará según el orden en el que se fue entregando. En caso de detectar errores o una presentación en la que falten conocimientos sobre el trabajo realizado que se consideren lo suficientemente graves, se pactará una fecha para una segunda defensa (donde deberán estar realizadas las correcciones y más acertada la presentación). *En caso de reprobar en esta segunda instancia de defensa, la condición es de libre*.

Los ítems se pueden ir perfeccionando a medida que se va avanzando, aunque no se tiene la misma consideración para la nota si fue editado luego de la fecha de entrega. **Pero sí importa!**