PROYECTO DATA SCIENTIST: Aprendizaje Supervisado

Planteamiento

El hospital F5 lleva un tiempo recogiendo datos cruzados que relacionan diversos indicadores de la vida y estado de salud de algunos pacientes frente a la variable de si han sufrido un ictus o no lo han hecho.

Ahora han puesto esos datos en manos del departamento de análisis de datos para elaborar un prototipo de programa con inteligencia artificial que de manera desatendida y como criba previa a una consulta con doctor pida por línea de comandos los datos necesarios y devuelva si el paciente está en riesgo de sufrir un ictus.

Para validar el proyecto, será necesario también un informe de su rendimiento.

Plazos

La entrega se realizará el día 28 de Octubre de 2024.

Condiciones de entrega

Para el día de la reunión, será necesario entregar:

- El repositorio en GitHub, con el trabajo ordenado adecuadamente en ramas y mensajes de commit limpios y claros.
- Un informe de la clasificación explicado que de cuenta de la capacidad de la IA
- Overfitting menor al 5%
- Trello y herramientas organizativas usadas

Tecnologías a usar

- Scikit-learn
- Pandas
- Git
- GitHub

Datos

Stroke Dataset

Niveles de Entrega

Nivel Esencial:

- ☐ Un modelo de ML funcional que prediga si un paciente está en riesgo de sufrir un ictus.
- ☐ Análisis exploratorio de los datos (EDA) con gráficos y estadísticas descriptivas.
- ☐ Controlar el overfitting, que la diferencia entre las métricas de training y las de test sea inferior a 5 puntos porcentuales.
- ☐ Aplicación en línea de comandos que permita ingresar datos del paciente y devuelva la predicción.
- ☐ Una solución que productivice el modelo (Una aplicación de Streamlit, GradIO, una API, un Dash o algo similar)
- ☐ Informe del rendimiento del modelo con métricas como la precisión, recall, F1-score y AUC-ROC, además de un análisis de las características más importantes que influyen en el riesgo de ictus.

Nivel Medio:

- ☐ Un modelo de ML con técnicas de ensemble
- ☐ Uso de técnicas de Validación Cruzada.
- ☐ Utilizar métodos para mitigar el efecto de los datos desbalanceados en el modelo.
- ☐ Optimización del modelo escogido con técnicas de ajuste de hiperparámetros (optuna, auto sklearn, pycaret, etc)
- ☐ Un sistema que monitorice la performance del modelo en producción.
- ☐ Incluir test unitarios.

Nivel Avanzado:

- ☐ Una versión dockerizada del programa.
- ☐ Guardado en bases de datos de los datos recogidos por la aplicación
- ☐ Despliegue en Cloud de las soluciones aportadas.
- ☐ Implementar un sistema de tracking para los experimentos de ML, registrando parámetros, métricas, código fuente y artefactos de cada experimento (usando MLFlow o similar)

Nivel Experto:

- ☐ Crear un modelo con redes neuronales, y comparar su rendimiento con los modelos de ML clásicos.
- ☐ Sistemas de entrenamiento y despliegue automático de nuevas versiones del modelo (A/B testing, Data Drifting, MLOps).
- ☐ En el futuro también se quieren utilizar imágenes, crear un prototipo de clasificador con redes neuronales convolucionales utilizando este otro dataset (keras, pytorch …)