PROYECTO "Patient Survival Prediction"

Miguel Angel Urueña Riobo miguel.uruena@udea.edu.co
C.C 1.006.121.797
Gaia Ramirez Hincapíe gaia.ramirez@udea.edu.co
C.C 1.005.273.358

Tutor:
Raul Ramos Pollan raul.ramos@udea.edu.co

INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL
CÓDIGO: 2508401 - INGENIERÍA DE SISTEMAS V4

Nombre del Proyecto: Patient Survival Prediction

Links Github:

- https://github.com/uruenariobo/ai4eng_health_issues
- https://github.com/gaiamilenium99/ai4eng_health_issues1

1. **Descripción del problema predictivo a resolver:**

   Dadas las características de un paciente y su historia clínica vamos a predecir la probabilidad de supervivencia de dicho paciente.

2. **Dataset a utilizar:**

   Vamos a usar el dataset de kaggle "Patient Survival Prediction" (enlace), que tiene 91.713 número de muestras (pacientes) y 85 columnas, de las cuales las columnas más representativas son:

   - Edad.
   - Índice de masa corporal.
   - Género.
   - Peso.
   - Presión sanguínea
   - Ritmo cardíaco máximo.
   - La ubicación del paciente antes de su ingreso en la unidad.
   - El APACHE IV (es una predicción probabilística de la mortalidad hospitalaria del paciente).

UNIVERSIDAD DE ANTIOQUIA
1803

3. **Métricas de desempeño requeridas (de machine learning y de negocio):**

Nuestro modelo de predicción de la supervivencia en pacientes debería de tener un porcentaje de acierto >80%, pero también un false negative rate <10%, ya que es un dictamen grave y delicado, por lo cual es preferible no fallar una predicción de un paciente que verdaderamente morirá, aunque eso implique que aumente el número de falsos positivos.

4. **Criterio sobre cuál sería el desempeño deseable en producción:**

Si la ocupación hospitalaria de las unidades de cuidados intensivos no aumentan más de un 10% no merece la pena poner el modelo en operatividad ya que el coste de desarrollo y mantenimiento no cubriría las ganancias adicionales de ese aumento.

5. **Iteraciones de desarrollo:**

Preprocesado de datos

El dataset que seleccionamos para el proyecto contenía 91713 registros y 85 campos, considerando que esperamos obtener un modelo de predicción de la supervivencia de distintos pacientes ingresados a centros hospitalarios en diferentes situaciones médicas, es importante la precisión en la información por lo que no pudimos utilizar métodos para completar datos faltantes cómo se aprendió en clase.
Para realizar el filtro de esta información se eliminaron campos que no tenían importancia en lo que se desea predecir, cómo la identidad del paciente o del hospital, o incluso de la icu. También se eliminaron datos que podrían presentar relación o dependencia con otras variables ya contenidas en el modelo, estás corresponden a algunas de los campos relacionados con Apache (luego de revisar qué información representaban, y observar una posible relación).
Luego de tener este filtro, el dataset pasó a tener 71 campos, donde realizamos una eliminación de los registros nulos en el dataset quedando con 62498 registros, como se observa en el colab.

Modelo supervisado

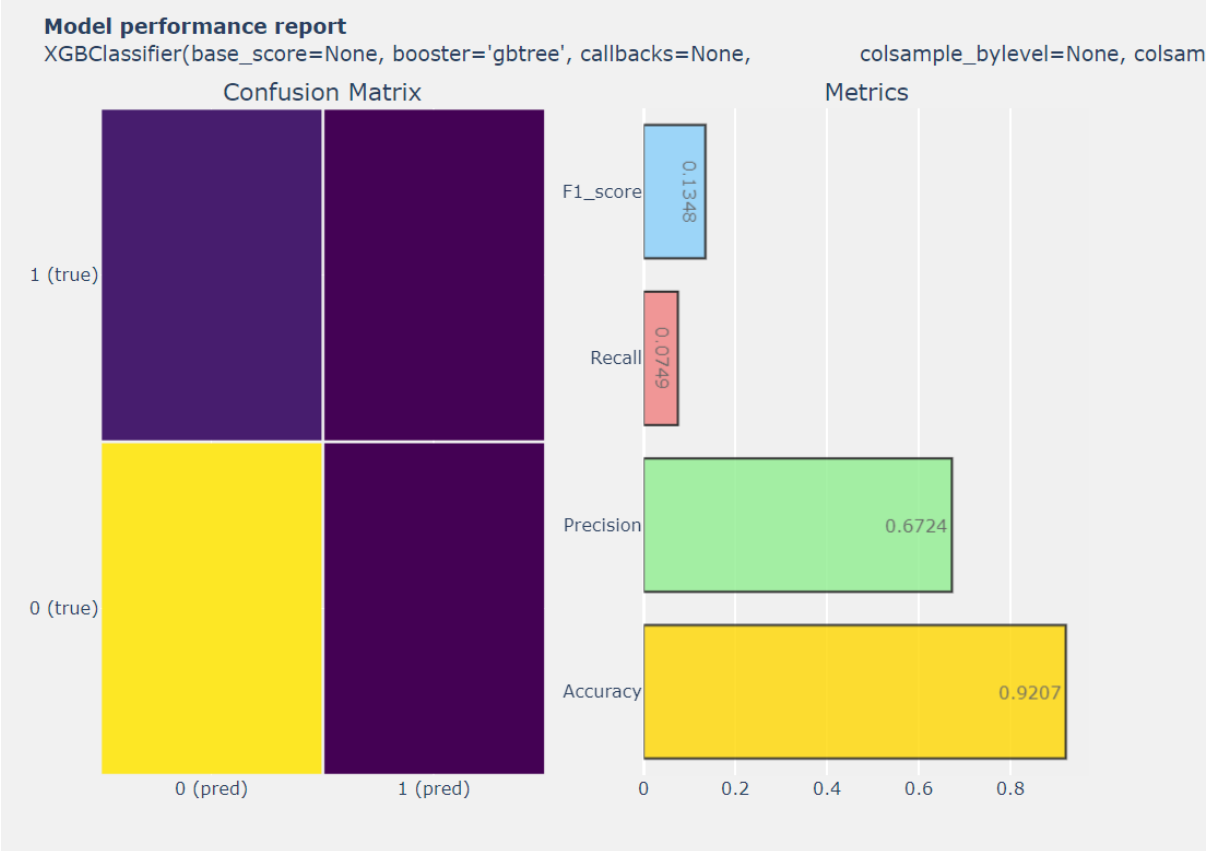
Se comienza con la preparación de los datos para el modelado. Se define una lista de variables numéricas y categóricas que se van a incluir en el modelo. A continuación, se usa la función pd.get_dummies() para convertir

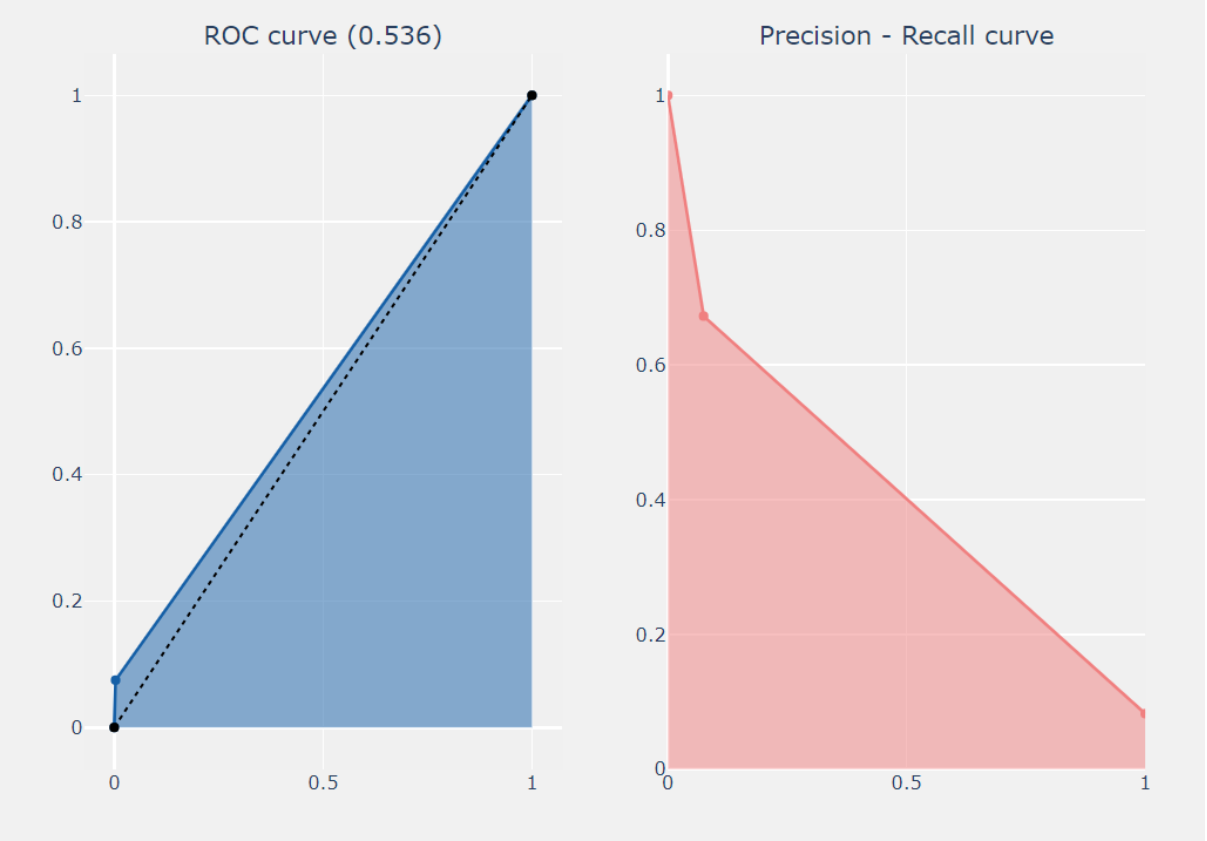
las variables categóricas en variables ficticias, lo que permite que los algoritmos de aprendizaje automático procesen los datos. Luego se seleccionan solo algunas variables numéricas para el modelado y se dividen los datos en conjuntos de entrenamiento y prueba usando train_test_split(). Se muestra la cantidad de valores de cada clase en los conjuntos de entrenamiento y prueba.

Resultados, métricas y curvas de aprendizaje.

A continuación, se define una función plot_model_performance() que toma un modelo entrenado y los datos de prueba y devuelve una representación gráfica de la precisión del modelo. Esta función calcula una matriz de confusión y muestra varias métricas de rendimiento del modelo, incluida la precisión, la recuperación, el puntaje F1 y el área bajo la curva ROC. También dibuja la curva ROC y la curva de precisión-recuperación del modelo. Finalmente, se usa la función make_subplots() de plotly.subplots para mostrar todas las gráficas en una sola figura.

Los resultados conseguidos fueron:





En resumen, hasta el avance obtenido se logró la preparación de los datos para el modelado, entrena un modelo y luego muestra la precisión del modelo mediante una serie de gráficos.

6. **Retos y consideraciones de despliegue:**

Retos de despliegue:

1. Asegurarse de que el modelo sea escalable y pueda manejar grandes cantidades de datos en tiempo real.
2. Garantizar que el modelo sea preciso y no tenga sesgos, ya que una mala predicción podría tener consecuencias graves para el paciente.
3. Implementar un sistema de monitoreo y alerta temprana para detectar cualquier problema o anomalía en el modelo.

Consideraciones de despliegue:

1. Se debe utilizar una infraestructura de alta disponibilidad para garantizar que el sistema esté siempre disponible y accesible.
2. Se debe realizar una validación rigurosa del modelo antes de desplegarlo en producción, y se deben realizar pruebas periódicas para asegurarse de que el modelo sigue siendo preciso y no tiene sesgos.
3. Se debe implementar un sistema de monitoreo y alerta temprana que notifique a los responsables del sistema en caso de cualquier problema o anomalía.
4. Se deben proporcionar herramientas de visualización y análisis para que los profesionales médicos puedan interpretar los resultados del modelo y tomar decisiones informadas en consecuencia.