ENTREGA FINAL DE PROYECTO

Por:

David Londoño Escalante

Jesús Andrés Rodríguez Villa

Asignatura:

Introducción a la Inteligencia Artificial

Docente:

Raúl Ramos Pollán

Universidad de Antioquia

Facultad de ingeniería

Medellín 2022

1. Introducción:

Descripción del problema predictivo a resolver.

Los hospitales normalmente manejan un gran flujo de pacientes, bajo esta circunstancia la permanencia de dichos pacientes por un tiempo prolongado es un parámetro crítico y es la causa principal de la congestión de los servicios de urgencia, esto genera problemas tanto para el personal médico como los pacientes por los largos tiempos de espera, saturación de la capacidad del hospital, mayor mortalidad de los pacientes y una mayor pérdida de recursos debido a pérdidas financieras.

Lo anterior invita a cada vez más tratar de gestionar mejor la estadía de los pacientes en las instalaciones de los hospitales de manera que se pueda mejorar la eficiencia de la atención en salud, facilitando el flujo de usuarios e identificando la gravedad de los pacientes. Es por esto que se propone plantear un modelo predictivo que permita estimar el tiempo de permanencia de los pacientes a partir de un conjunto de registros de un hospital.

2. Exploración descriptiva del Dataset:

Los datos contenidos en el dataset incluyen números de registro de cada paciente, código de los hospitales, número de camas disponibles, entre otros parámetros importantes para el análisis.

Primer archivo: **train.csv**

- **Case_ID:** Número de registro del paciente.
- **Hospital_code:** Código que identifica el hospital.
- **Hospital_type_code:** Código que identifica el tipo de hospital.
- **City_Code_Hospital:** Código que identifica la ciudad del hospital.
- **Hospital_region_code:** Código que identifica la región del hospital.
- **Avaliable extra room:** Cantidad de camas extra disponibles.
- **Departament:** Departamento del hospital en el cual está siendo atendido el paciente.
- **Ward_type:** Tipo de pabellón en el que se encuentra el paciente, en términos de la complejidad.
- **Ward_facility_code:** Código del pabellón.
- **Bed_grade:** Grado de cama
- **PatientID:** Número de identificación del paciente.

- **Codepatient:** Código del paciente.
- **Type of admssion:** Tipo de admisión del paciente.
- **Severity of illnes:** Grado de severidad de la enfermedad.
- **Visitors with patient:** Número de visitantes habilitados con el paciente.
- **Edge:** Limite
- **Admission_Deposit:** Depósito de admisión.
- **Stay:** Estado

Segundo archivo: **test.csv**

- Los datos de prueba tienen las mismas características que los datos de entrenamiento. Pronostican el tiempo de permanencia de los pacientes en el hospital.

Métricas.

El modelo se plantea como un problema de clasificación, basado en el tiempo de permanencia de un paciente en el hospital, el cual está dividido en 11 clases diferentes dependiendo del número de días de estancia en un rango de 0 a 100 días.

Por este motivo, basado en el tipo de problema, las métricas para evaluar el desempeño del modelo son: sensibilidad, exactitud y precisión.

La precisión y la sensibilidad, entre las clases 3 a 7, tienden a ser mayor porque para estas se poseen mayor cantidad de datos en el modelo.

3. Iteraciones de desarrollo:

Desempeño.

El desarrollo final de este modelo debería estar en capacidad de tener una previsión precisa del tiempo de instancia de un determinado grupo de pacientes en hospitales de bajo o alto flujo de personas.

De igual manera, se podría realizar de forma más genérica para adaptarse a distintas situaciones que requieran del manejo de datos en determinados intervalos de tiempo.

Exploración del dataset y preprocesado.

En primer lugar se realizó el preprocesado y exploración de los datos con el fin de identificar las variables que están en el encabezado del archivo, la cantidad de datos en los data frame y los datos faltantes que puedan haber en los mismos.

- En primera instancia se carga la base de datos y se exploran los encabezados de las columnas para el dataset de prueba y el de entrenamiento. También se explora el tamaño de ambos dataset.
- Posteriormente se exploran los datos nulos presentes en ambos dataset y se procede a eliminarlos.
- También se realizó la identificación del tipo de variables en los dataset.
- Posteriormente se realizó el análisis de las variables categóricas presentes en el conjunto de datos, estas variables categóricas son las posibilidades de estados que pueden tomar cada una de las columnas del conjunto de datos. Para el caso los conjuntos de variables categóricas para cada una de dichas columnas son los siguientes.

```
#Variables categóricas
ccols = [i for i in dtr.columns if not i in dtr._get_numeric_data()]
for c in ccols:
    print ("%10s"%c, np.unique(dtr[c].dropna()))
```

```
Hospital_type_code ['a' 'b' 'c' 'd' 'e' 'f' 'g']
Hospital_region_code ['X' 'Y' 'Z']
Department ['TB & Chest disease' 'anesthesia' 'gynecology' 'radiotherapy' 'surgery']
 Ward_Type ['P' 'Q' 'R' 'S' 'T' 'U']
Ward_Facility_Code ['A' 'B' 'C' 'D' 'E' 'F']
Type of Admission ['Emergency' 'Trauma' 'Urgent']
Severity of Illness ['Extreme' 'Minor' 'Moderate']
       Age ['0-10' '11-20' '21-30' '31-40' '41-50' '51-60' '61-70' '71-80' '81-90'
 '91-100']
      Stay ['0-10' '11-20' '21-30' '31-40' '41-50' '51-60' '61-70' '71-80' '81-90'
 '91-100' 'More than 100 Days']
```

- En la imagen anterior se puede observar que variables como el código de hospital toman un conjunto de variables entre las letras a y g, así como el código de región, el tipo de sala y el código de facilidad de sala.
- También se puede observar variables como el departamento del hospital en el que se encuentra cada paciente, el tipo de admisión y la severidad de la enfermedad cuyas variables categóricas responden a nombres particulares.

- Y por último se tiene un grupo de variables en rangos numéricos como la edad de los pacientes que va de 0 a 100 años, y la variable objetivo que es el tiempo de estadía en el hospital que va de 0 a más de 100 días.

Con el análisis anterior se puede tomar la decisión de eliminar algunas columnas que no representan gran cantidad de información o cuya información no es tan relevante para análisis posteriores del modelo, estas columnas eliminadas fueron, el código de región del hospital, el tipo de código, y el código de facilidad de las salas.

Conociendo la variable objetivo, que para el caso como se definió en el entregable uno es el tiempo de estadía de los pacientes en el hospital, se realizó la distribución de dicha variable.

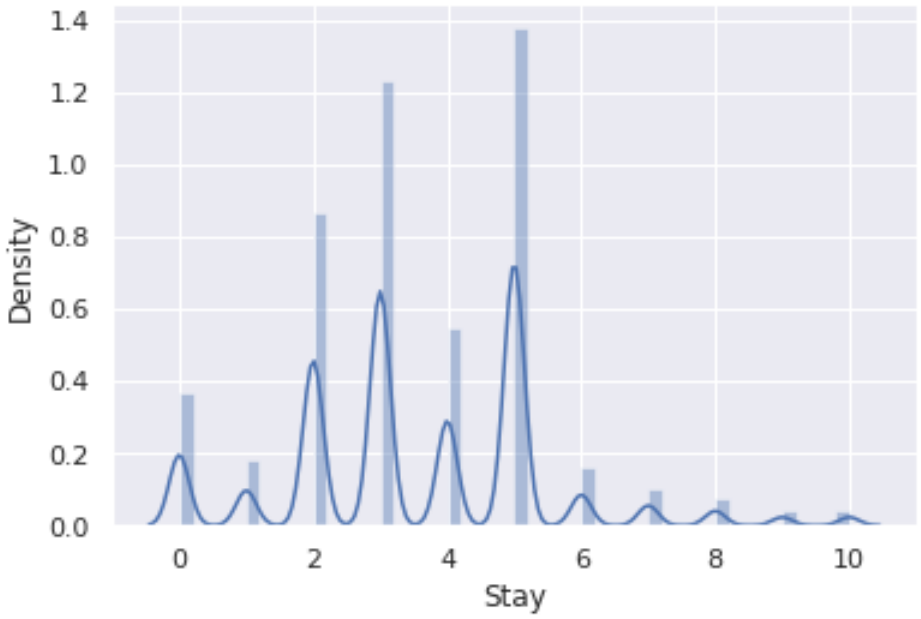


Figura 1. Distribución de la variable objetivo.

Por último se realizó una matriz de correlación que muestra cómo están relacionadas entre sí una a una las variables del conjunto de datos. De esta matriz se pueden ver resultados importantes, como la alta interrelación que tiene variables como la severidad de las enfermedades y el grado de cama, la complejidad del hospital y la cantidad de camas extra disponibles, y también la relación entre el tiempo de estadía en el hospital y la cantidad de visitantes con cada paciente.

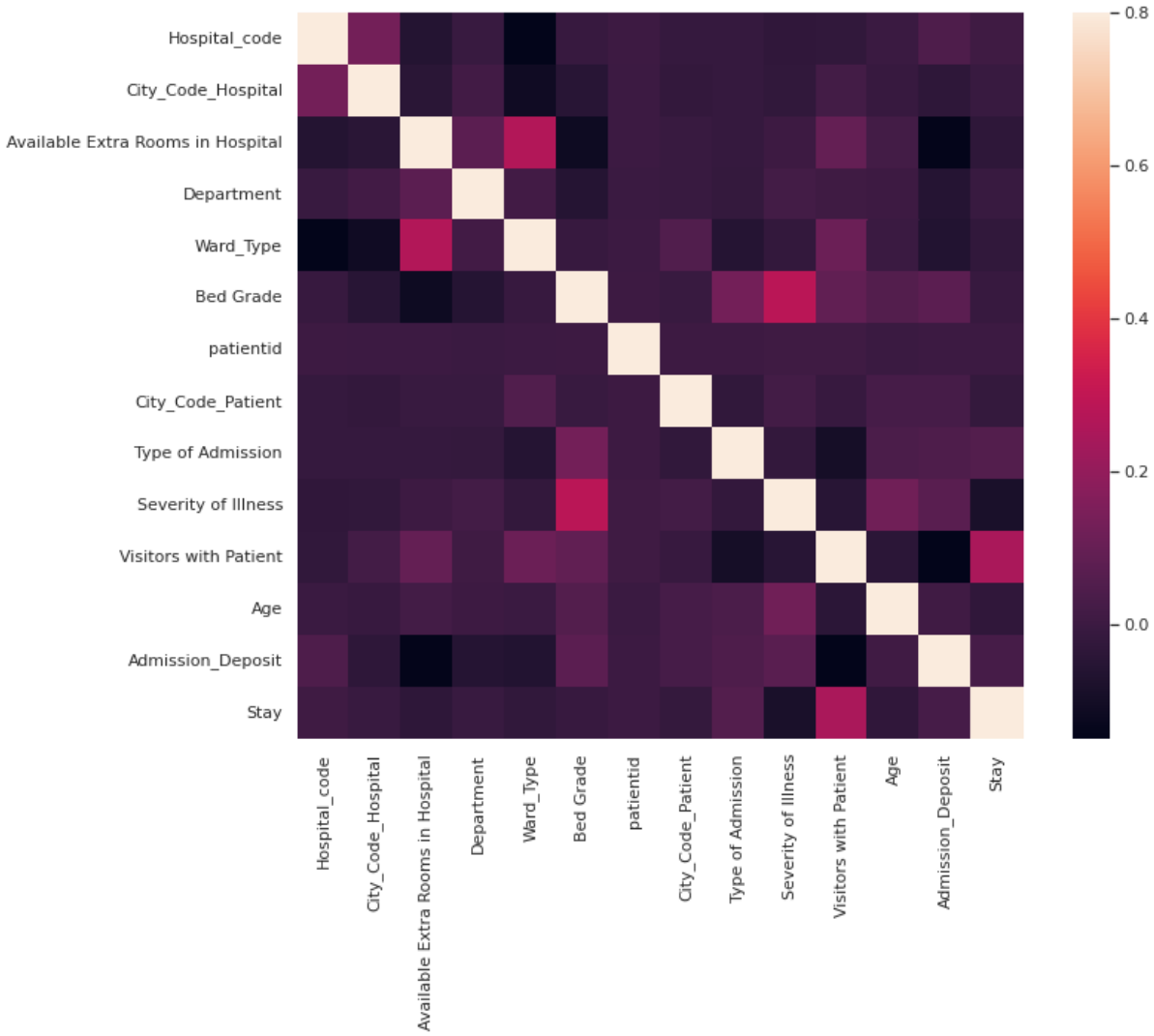


Figura 2. Matriz de correlación.

Generación del modelo.

En primera instancia se utiliza un árbol de decisión con el cual se probaron tres valores de hiperparametros. Este tipo de árbol de decisión nos permite clasificar los datos del data frame calibrando el modelo según se desee para el uso específico.

```
#Utilizando un valor en hiperpárametro bajo
tree_Estimator = DecisionTreeClassifier(max_depth=5)
tree_Estimator.fit(Xtr,ytr)
print ("train accuracy %.2f"%tree_Estimator.score(Xtr,ytr))
print ("test accuracy  %.2f"%tree_Estimator.score(Xts,yts))
```

```
train accuracy 0.39
test accuracy  0.39
```

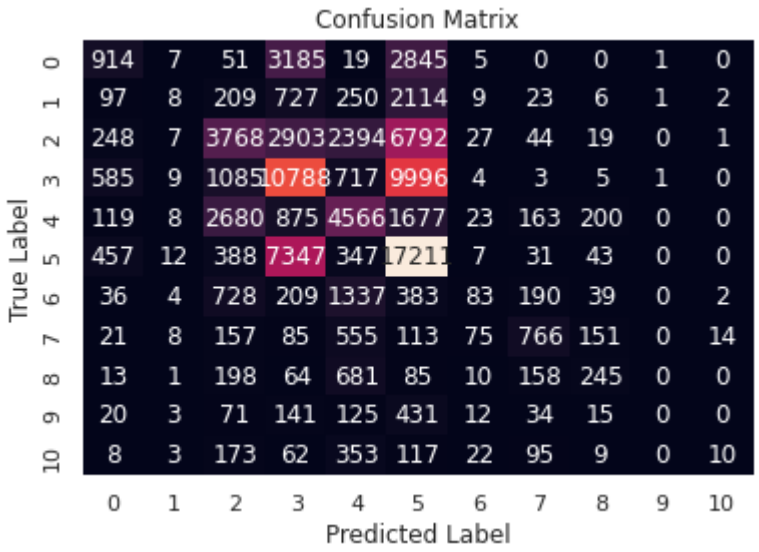
```
#Incrementando el hiperparámetro
tree_Estimator = DecisionTreeClassifier(max_depth=20)
tree_Estimator.fit(Xtr,ytr)
print ("train accuracy %.2f"%tree_Estimator.score(Xtr,ytr))
print ("test accuracy  %.2f"%tree_Estimator.score(Xts,yts))
```

```
train accuracy 0.71
test accuracy  0.35
```

```
#Valor intermedio en hiperparámetro
tree_Estimator = DecisionTreeClassifier(max_depth=10)
tree_Estimator.fit(Xtr,ytr)
print ("train accuracy %.2f"%tree_Estimator.score(Xtr,ytr))
print ("test accuracy  %.2f"%tree_Estimator.score(Xts,yts))
```

```
train accuracy 0.43
test accuracy  0.41
```

Posteriormente se tiene una matriz de confusión en la cual podemos ver las combinaciones diferentes de valores predichos y reales del clasificador



Posteriormente tendremos la curva de aprendizaje con la cual podemos saber que le está sucediendo a nuestro estimador, y saber cómo se comportan las métricas mientras se aumenta el número de datos procesados.

En nuestro caso la curva de aprendizaje muestra cercanía entre el entrenamiento y test pero el desempeño no es el esperado.

4. Retos y consideraciones de despliegue:

Durante estos últimos 4 meses de trabajo en el proyecto se han encontrado distintos retos, los cuales se han logrado solventar, los más destacados son:

- La carga de archivos en la plataforma de evolución, partiendo de la base que la información se suministra en archivos cuyo formato es en extensión .xlsx.
- Encontrar el número correcto de iteraciones basado en el modelo predictivo y que sea acorde a resultados óptimos y concluyentes.
- El discernimiento en la elección de las columnas de datos a eliminar y que este acto no haga perder coherencia en la distribución de la variable objetivo.
- Se pudo ver durante la implementación de los algoritmos predictivos, que para un gran volumen de información es importante planear de forma escalonada el procesamiento y análisis de los datos, esto con el fin de tener tiempos de ejecución más cortos y que requieran menos recursos de la máquina que se esté utilizando, en nuestro caso al final del proceso iterativo se nos presentaron problemas de ejecución ya que para graficar curvas de aprendizaje para el numero de variables y los datos asociados a estas que se precisaba de gran cantidad de tiempo de ejecución lo que en ocasiones impedía el correcto accionar del entorno y la maquina utilizada.
- Se evidencio que las variables tratadas en el modelo corresponden a situaciones bastante diversas en el campo de la medicina, por lo tanto los valores de exactitud que se alcanzaron con el modelo, en general no fueron muy satisfactorios, es claro que para un tema tan importante como

lo es la salud y el manejo de los servicios médicos y hospitalarios se puede escalar a modelos más robustos en los cuales las variables de interés sean tratadas de manera más profunda y con resultados que muestren relaciones más claras entre los tiempos de respuesta y sus resultados reales.

5. Conclusiones:

- El modelo trabajado en el proyecto, en caso de seguir desarrollándose puede ser aplicable en muchas áreas del diario vivir como medios de transporte, disponibilidad de vuelos en aeropuertos y todas secciones afines a la ocupación de turnos.
- El modelo puede ser expandible en zonas de coberturas complejas para el sistema de salud para facilitar actividades de bajo rendimiento, no necesariamente en el área de urgencias.
- Como estudiantes de ingeniería eléctrica (e ingeniería en general) se ha encontrado una gran utilidad de la programación de todo tipo de eventos predictivos, los cuales se pueden adecuar para un modelo de demanda y de generación de energía eléctrica por zonas o en determinado territorio y todo basado en la experiencia del proyecto desarrollado y no tan "convencional" (área de la salud) en el sector energético.