Reporte de Análisis de Modelo de Machine Learning

Modelo Seleccionado: Modelo con aplicación de un Framework

Inteligencia Artificial Avanzada para la Ciencia de Datos (Grupo 101)

Módulo 2: Machine Learning

Emilia Victoria Jácome Iñiguez - A00828347

Ingeniería en Transformación Digital de Negocios (ITD)

Profesor Docente

Iván Mauricio Amaya Contreras

Viernes 9 de septiembre de 2022

Tecnológico de Monterrey, Campus Monterrey

Dataset utilizado

Name: Brain stroke prediction dataset full_filled_stroke_data.csv

Source: Kaggle
Original Source: Data files © Original Authors

Length: 4981

Modelo de ML Aplicado

Clasificador de Árbol de Decisión/ Decisión Tree Classifier

Separación y Evaluación del Modelo

[Separación y evaluación del modelo con un conjunto de prueba y un conjunto de validación (Train/Test/Validation)]

Para la separación de las variables entre un conjunto de entrenamiento, un conjunto de prueba y un conjunto de validación, se procedió a utilizar la función train_test_split de la librería de sklearn.model_selection para separar los datos.

La proporción utilizada para separar cada subconjunto de datos, fue de 60% – 30% – 10% para entrenamiento – prueba – validación respectivamente. Para lograr 3 grupos en lugar de uno, se utilizó la función de train_test_split dos veces, la primera en función del dataset completo y la segunda con respecto a los datos de prueba.

Por lo que se tienen 2988 datos para el entrenamiento, 1494 datos para la prueba, y 499 datos para la validación.

Análisis de Sesgo

[Diagnóstico y explicación el grado de bias o sesgo: bajo medio alto]

El grado de sesgo en este modelo es de nivel medio ya que los datos con los que se cuenta contienen mayor cantidad de pacientes que no han tenido un infarto cerebral que pacientes que si han presentado tal y cómo se puede observar en la siguiente imagen.

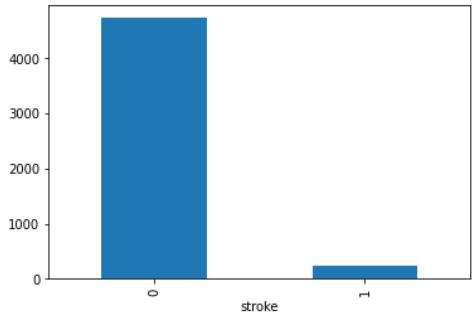


Figura No.1 Conteo de datos por columna de stroke

Este sesgo en los datos afecta directamente al entrenamiento y prueba del modelo ya que tiene más información sobre personas que no se han infartado y muy poca información sobre personas que si, por ende, tiende a detectar con un alto grado de precisión (mayor al 90%) si la persona no se infartó tal y como se muestra en la siguiente imagen.

	precision	recall	f1-score	support
0	0.96	0.95	0.96	1423
1	0.16	0.18	0.17	71
accuracy			0.92	1494
macro avg	0.56	0.57	0.56	1494
weighted avg	0.92	0.92	0.92	1494

Figura No.2 Reporte de Clasificación del Árbol 1

No obstante, el algoritmo tiene un nivel de precisión bajo (menor al 20%) para acertar a una persona que en efecto se infartaron ya que el set de datos de entrenamiento (y el general) contiene muy pocos casos de pacientes infartados. Por lo tanto, se puede decir que el algoritmo contiene un sesgo a etiquetar a un paciente como no infartado cuando en realidad si le ocurrió un infarto y esto repercute por completo la calidad del modelo.

	precision	recall	f1-score	support
0	0.96	0.95	0.96	1423
1	0.16	0.18	0.17	71
accuracy			0.92	1494
macro avg	0.56	0.57	0.56	1494
weighted avg	0.92	0.92	0.92	1494

Figura No.3 Reporte de Clasificación del Árbol 1

De igual manera, en la etapa de prueba del modelo, se puede apreciar el efecto de este sesgo a través de la matriz de confusión:

```
[[1346   77]
 [  56   15]]
```

Figura No.4 Matriz de Confusión del Árbol 1

Se puede ver claramente que los valores de la diagonal principal están muy desbalanceados ya que hay una diferencia desproporcional entre ellos y que la

cantidad de falsos negativos es casi 90 veces más grande que el valor de verdaderos positivos.

De igual manera, se justifica la existencia de este sesgo por medio de la similitud de resultados obtenidos entre los dos modelos de árboles de decisión propuestos. He aquí los resultados:

	precision	recall	f1-score	support
0	0.96	0.95	0.96	1423
1	0.16	0.18	0.17	71
accuracy			0.92	1494
macro avg	0.56	0.57	0.56	1494
weighted avg	0.92	0.92	0.92	1494

	precision	recall	f1-score	support
0	0.96	0.95	0.95	1423
1	0.16	0.21	0.18	71
accuracy			0.91	1494
macro avg	0.56	0.58	0.57	1494
weighted avg	0.92	0.91	0.92	1494

Figura No.5 Comparación de Reportes de Clasificación entre Árbol 1 y Árbol 2 respectivamente

Dado que no varía mucho las métricas de desempeño entre el primer modelo y el segundo, se puede concluir que se debe a la limitada data de pacientes que si tuvieron un infarto a pesar de que cada modelo tiene un criterio diferente para definir el valor/peso de sus hojas.

Análisis de Varianza

[Diagnóstico y explicación del grado de varianza: bajo medio alto]

Al analizar la varianza del modelo, se realizó una prueba para verificar **el cambio en la estimación del modelo** al cambiar la proporción de datos de entrenamiento con la finalidad de determinar el nivel de variabilidad de la función de costo si se cambiara **al producir pequeños cambios en el set de entrenamiento**.

Por lo que se procedió a comparar los valores de las métricas de desempeño para diferentes proporciones del set de entrenamiento:

	Name	Score_a	Score_p	Score_r	Score_f1
0	Tree1	0.895571	0.230952	0.223333	0.191667
1	Tree2	0.897184	0.203333	0.198333	0.166667

Figura No. 6 Métricas de evaluación K-Folds para una proporción de 50% datos de entrenamiento.

	Name	Score_a	Score_p	Score_r	Score_f1
0	Tree1	0.907918	0.195000	0.34	0.233333
1	Tree2	0.913959	0.235952	0.41	0.289062

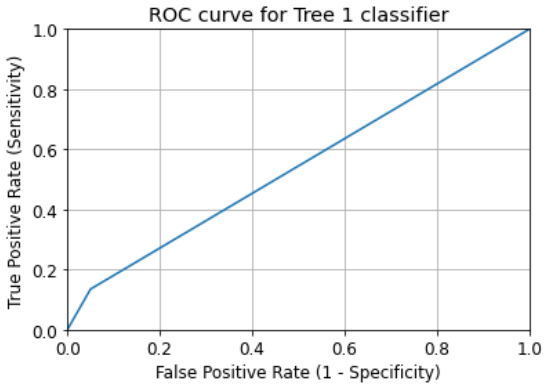
Figura No. 7 Métricas de evaluación K-Folds para una proporción de 60% datos de entrenamiento.

	Name	Score_a	Score_p	Score_r	Score_f1
0	Tree1	0.914296	0.30	0.206667	0.213810
1	Tree2	0.898293	0.19	0.153333	0.130476

Figura No. 8 Métricas de evaluación K-Folds para una proporción de 70% datos de entrenamiento.

Al concluir la prueba, se llegó a la conclusión de que la variación en la calidad de las estimaciones **es mínima** en cuanto a la métrica de precisión con un margen de variación de +-0.01, tomando como referencia el ratio de separación del dataset original (60%-30%-10), lo cual **no es significativo** para marcar una diferencia entre la calidad del modelo o no. Si bien existe una variación más significativa en la especificidad, esta se debe al sesgo de la data anteriormente explicado donde predomina la cantidad de resultados con valor 0 que de 1. Por lo que, aún así la varianza no es significativa para este caso. Por consiguiente, se puede decir que el nivel de **varianza del modelo es bajo**.

Consecuentemente, esto también puede ser corroborado por las **curvas ROC,** las cuales indican el nivel de desempeño general en base a la tasa de verdaderos positivos frente a la tasa de falsos positivos:




```
[ ]  1 # IMPORTANT: first argument is true values, second
     2 print(metrics.roc_auc_score(y_test, y_pred_prob))
```

```
0.5424016774984052
```

Figura No. 9 Curva ROC y valor AUC para para una proporción de 50% datos de entrenamiento

Figura No. 10 Curva ROC y valor AUC para para una proporción de 60% datos de entrenamiento

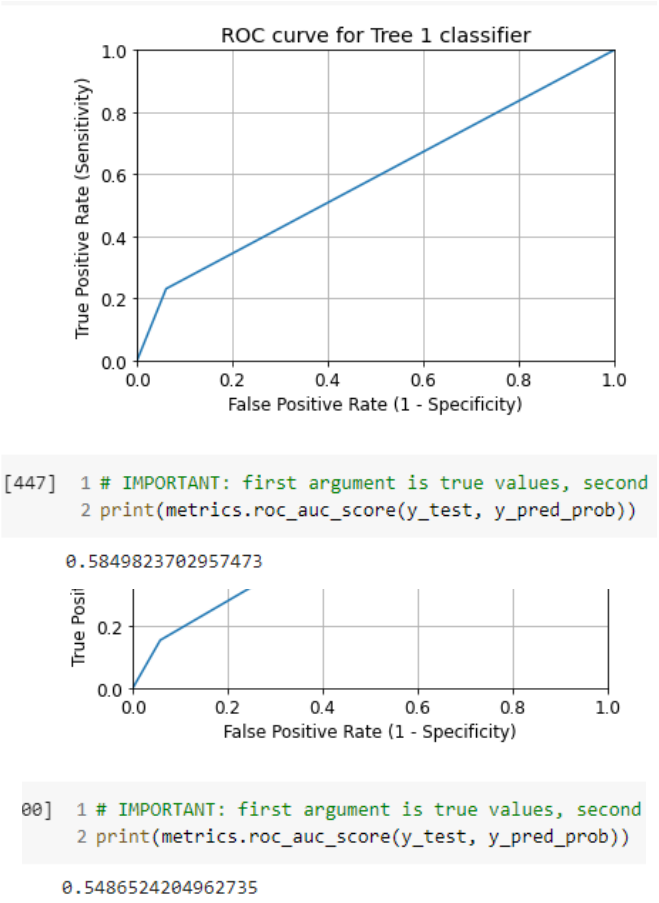


Figura No. 11 Curva ROC y valor AUC para para una proporción de 70% datos de entrenamiento

Dichas curvas junto con el valor de su respectiva área **muestran poca variación** con respecto a la separación del dataset original con un margen de +-0.04. Claramente, el valor de AUC tiene una tendencia a aumentar con respecto a la cantidad de datos para el entrenamiento, no obstante, su incremento no es significativo. En conclusión, el hecho de que el modelo tenga poco nivel de variación implica que es un modelo flexible y cuasi óptimo.

Nivel de Ajuste

[Diagnóstico y explicación del nivel de ajuste del modelo: underfitt fitt overfitt]

En cuanto al nivel de ajuste del modelo, se afirma que existe un caso de underfitting donde el modelo es incapaz de llegar a los valores reales debido a que no tiene la suficiente complejidad en su configuración para ajustarse a todos los datos disponibles y no tiene un buen nivel de precisión con su dataset de entrenamiento. Esto en parte, se debe al sesgo proveniente de la fuente de datos, que a su vez provoca un desbalance en el entrenamiento y afecta directamente la calidad de predicciones que genera el algoritmo propuesto.

Técnicas de Regularización/Ajuste de parámetros

[Basándote en lo encontrado en tu análisis utiliza técnicas de regularización o ajuste de parámetros para mejorar el desempeño de tu modelo y documenta en tu reporte cómo mejoró este.]

Se utilizaron varias técnicas de regularización y/o ajuste para mejorar los resultados del modelo. Entre ellas, la primera que se aplicó, fue la de ampliar el conjunto de entrenamiento, restableciendo el ratio de entrenamiento-prueba-validación a 70% – 22.5% – 7.5%.

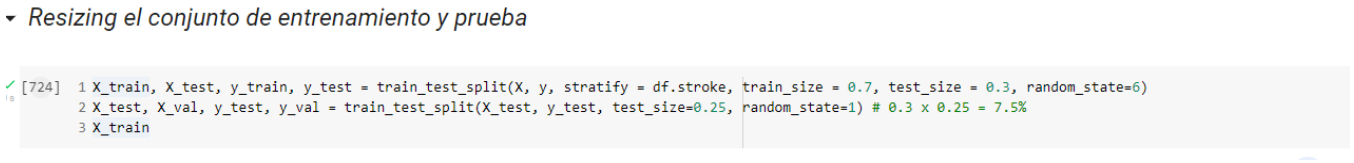


Figura No. 12 Repartición del conjunto de datos de entrenamiento y prueba

En segunda instancia, se procedió a aplicar la técnica de prunning o "podado" del árbol, la cuál consiste en reducir las partes del árbol que realmente no aportan a clasificar las instancias. Los árboles de decisión son los algoritmos de Machine Learning más propensos al sobreajuste (overfitting) por lo que es bien conocido que esta técnica es de mucha utilidad para erradicar este suceso.

[Falta incluir la métrica de cómo cambia la varianza y el sesgo tras la aplicación de estas técnicas]

Finalmente, en tercera instancia, se procedió a aplicar un modelo de bosques aleatorios el cuál es capaz de entrenar árboles con condiciones diferentes, seleccionadas de forma aleatoria. Por ende, el objetivo es reducir la variación de las salidas de los datos ya que corresponde al promedio de las salidas de los árboles de los que se compone.

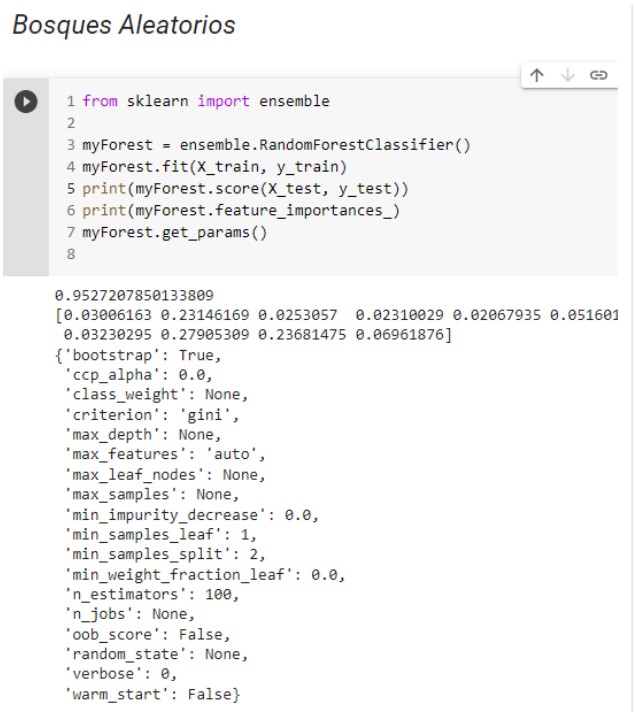


Figura No. 13 Aplicación del Modelo de Bosques Aleatorios

[Falta incluir la métrica de cómo cambia la varianza y el sesgo tras la aplicación de estas técnicas]

Comparación