Evaluación Parcial Tópicos IA 2022-02

Saul Andersson D. Rojas Coila
Departamento de Ciencia de la Computación, UCSP
saul.rojas@ucsp.edu.pe

Abstract

El problema de la clasificación en *machine learning* ha sido explorada durante ya varios años; algoritmos capaces de clasificar imágenes, texto, datos variados, etc. son mejorados continuamente para lograr un mejor rendimiento y ser referenciados como el estado del arte en este problema en particular. En el presente proyecto se expondrá un problema de clasificación binaria basada en aprendizaje supervisado: el problema básico es determinar si un alumno desertará o no de la universidad basándonos en otras variables. Se aplicará una metodología paso a paso para resolver problemas de *machine learning*, además que se probarán diferentes técnicas de clasificación, al final se dará un resultado basándono en ciertas métricas utilizadas para los algoritmos de clasificación.

1 Introducción

El problema de la clasificación es bien conocido en el campo del *machine learning*, en el área del aprendizaje supervisado existen varios modelos propuestos, algunos se dan en contexto multiclase y otros solo para clasificación binaria.
En este proyecto se busca evaluar diferentes modelos clásicos de clasificación al problema de determinar si un estudiante desertará de la universidad o no, un problema de clasificación binaria. Se seguirá una metodología paso a paso desde la extracción de los datos hasta la predicción final del modelo.

2 Metodología

La metodología que se siguió (Geron, 2020) comprende una serie de pasos que garantiza una correcta estimación de resultados basándonos en varios modelos utilizados para las tareas comunes en el *machine learning*, más abajo se muestra la lista de pasos a seguir:

1. Encuadre del problema y panorama general: En este paso definiremos el problema que queremos resolver con *machine learning*.

2. Obtención de datos: En este paso obtenemos el conjunto de datos para nuestros algoritmos.

3. Exploración de los datos para las perspectivas: Podemos hacer un estudio preliminar de las variables que participan en la tarea, así como una posible selección de las más relevantes, también estimar el criterio de separación de los datos en un conjunto de entrenamiento y de prueba.

4. Preparación de los datos para la exposición de patrones: Los algoritmos de *machine learning* evalúan datos numéricos, por lo que una transformación de nuestros datos es necesario, basándonos también en ciertos criterios.

5. Exploración de modelos y *ranking* de los mejores: Para nuestra tarea no solo debemos considerar un modelo que resuelva el mismo problema sino varios para escoger el más idóneo.

La implementación del proyecto se encuentra en: **https://github.com/politeperson/Topicos-IA-2022-02/blob/main/ExamenParcial/Proyecto%20Primer%20Parcial.ipynb**

6. Ajuste del modelo: Una vez seleccionados los algoritmos mejor *rankeados* debemos modificar sus hiperparámetros con tal de obtener el mejor resultado.

7. Presentación de la solución: Probar nuestra solución óptima con el conjunto de prueba y verificar algunas hipótesis.

8. Lanzamiento, monitoreo y mantenimiento del sistema: Analizar si en producción nuestros modelos rinden correctamente (*en estas circunstancias como es un trabajo académico no se considerará este paso*).

3 Implementación del proyecto

En esta sección se explicará a grandes rasgos lo que se realizó en cada una de las etapas de implementación del proyecto de *machine learning* siguiendo la metodología anteriormente explicada.

3.1 Encuadre del problema y panorama general

El problema propuesto es determinar si un estudiante desertará o no de la universidad considerando ciertas variables como su semestre, la cantidad de cursos que ha llevado, etc. El problema de clasificación es binario por lo que se evaluarán varios modelos que lidien con este problema. Como dato adicional utilizaremos `python` como lenguaje de programación y librerías como `numpy`, `pandas`, `scikit-learn`, `matplotlib` y `seaborn`.

3.2 Obtención de datos

Los datos se obtienen del siguiente conjunto de datos: **`https://github.com/politeperson/Topicos-IA-2022-02/blob/main/ExamenParcial/datasets/datos_desercion.csv`**, El conjunto de datos pesa aproximadamente 211.1 KB, por lo que este tiene un tamaño manejable que se puede administrar en un jupyter notebook, en total son 4500 registros en el conjunto de datos.

3.3 Exploración de los datos para las perspectivas

Al importar los datos las variables encontradas fueron las siguientes: `cant_semestres`, `sexo`, `prom_notas_semestre`, `prom_inasistencias`, `cursos_reprobados`, `desercion`; entre las variables categóricas se encuentran la de `sexo` y `desercion`, pero la primera están basadas en texto, por lo que será necesaria una transformación en la fase de preparación de los datos. La variable `desercion` es nuestra variable objetivo. La primera observación que se tuvo fué que no es un conjunto de datos balanceado ya que el porcentaje de alumnos que no desertaron contra los que desertaron era de un 96% contra un 4% respectivamente, por ello se separó el conjunto de datos estratificadamente (30% para el conjunto de prueba y 60% para el entrenamiento) por la variable `desercion`, luego también se pudo observar que la variable `cant_semestres` tiene una alta correlación con la de `desercion`, y esto tiene sentido ya que por lo general los alumnos que más desertan son aquellos que están en semestres intermedios o superiores, en la imagen 1, se ve la correlación de todas las variables.

3.4 Preparación de los datos para la exposición de patrones

En cuanto a la preparación de los datos no se encontraron datos vacíos, la variable categórica `sexo` fué codificada con vectores *one-hot*. Para los datos numéricos se usó una normalización estándar, todos los pasos fueron implementados en un *pipeline* tanto numérico como categórico.

3.5 Exploración de modelos y *ranking* de los mejores

Los modelos seleccionados fueron los siguientes: *Naive Bayes*, Árbol de decisión, *Random Forest*, *Support Vector Machine* y Regresión Logística (implementación propia) (Gong, 2022). Para la evaluación de los modelos se utilizó la validación cruzada con 5 *k-folds*, Las métricas utilizadas para la evaluación fueron **F1**, ***Precision*** y ***Recall***, en la sección de resultados se verán los resultados obtenidos.

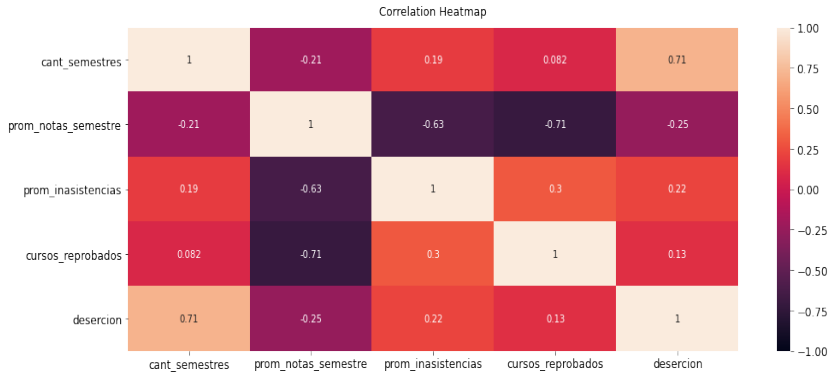

Figure 1: Mapa de calor de la matriz de correlación.

3.6 Ajuste del modelo

La métrica escogida para seleccionar el mejor modelo fué la **F1** ya que respresenta una relación entre la métrica ***Precision*** y el ***Recall***. El modelo que mejor desempeño tuvo fué *Support Vector Machine*, así que se hizo un reajuste del modelo con dos *kernels*, uno lineal y otro *rbf (radial basis function)*, con diferentes parámetros para cada uno para una búsqueda en *grid* usando la librería *scikit-learn*, el modelo más óptimo fué un ***SVM*** *(Support Vector Machine)* con *kernel* lineal y parámetro *C* de 1.

3.7 Presentación de la solución

En cuanto al resultado final el mejor modelo escogido, en este caso el *Support Vector Machine* con *kernel* lineal y parámetro C de 1 obtuvo un puntaje ***F1*** aproximado, en el conjunto estratificado de prueba, de 0.839, lo cual es un resultado aceptable.

4 Resultados

En cuanto a las métricas tomadas por cada modelo, se realizó la validación cruzada con 5 *k-folds*, los resultados mostrados en la tabla 1 son la media de las 5 ejecuciones realizadas en la validación cruzada para cada modelo.

Modelo	**F1**	***Precision***	***Recall***
Naive Bayes	0.797	0.947	0.690
Árbol de decisión	0.674	0.657	0.708
Random Forest	0.794	0.956	0.682
Support Vector Machine	0.798	1.000	0.666
Regresión logística	0.769	0.999	0.634

Table 1: Tabla de resultados.

Los resultados de la regresión logística corresponden a la implementación propia, no a la de *scikit-learn*.

5 Conclusiones

Se realizó un enfoque estructurado para resolver un problema de *machine learning*, los pasos seguidos contribuyeron al entendimiento de cómo se logra dar solución a un problema de este tipo, en cada sección se aplican diferentes técnicas tanto estadísticas, como de visualización y de *machine learning*, lo cual ayudó a encuadrar de mejor manera soluciones y diferentes técnicas que se utilizan en cada parte de la implementación del proyecto. También fué clave investigar sobre los diferentes modelos de clasificación que son aplicados en problemas del mundo real y así tener un mejor criterio a la hora de resolver el problema dado.

References

A. Géron, Hands-on machine learning with scikit-learn, Keras, and tensorflow: Concepts, tools, and techniques to build Intelligent Systems. O'Reilly, 2020.

D. Gong, "Top 6 Machine Learning Algorithms for Classification", Medium, 2022. [Online]. Available: https://towardsdatascience.com/top-machine-learning-algorithms-for-classification-2197870ff501. [Accessed: 14- Oct- 2022].