Instituto Tecnológico y de Estudios Superiores de Monterrey

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

Selección, configuración y entrenamiento del modelo

Equipo 5

Jorge Eduardo De León Reyna - A00829759
David Esquer Ramos - A01114940
Francisco Mestizo Hernández - A01731549
Adrián Emmanuel Faz Mercado - A01570770

Agosto 28, 2023

Introducción

Una vez que terminamos de hacer la limpieza de los datos, podemos comenzar con la decisión del modelo que se utilizará. El problema que se nos presenta es predecir si un tripulante Titanic sobrevive o no basado en diferentes características. Las características que permanecieron después de realizar la limpieza de los datos están relacionadas con su persona, el lugar donde embarcó, la clase de su cabina y cuánto pagó por su boleto.

Los algoritmos de clasificación están especializados en aprender patrones en sets de datos que se encuentran previamente categorizados. Una vez que el modelo está entrenado y detecta estos patrones, se le pueden dar nuevos sets de datos que no estás categorizados y el modelo predecirá a qué categoría pertenecen (Wolff, 2022).

Por esto, podemos utilizar un algoritmo de clasificación para predecir si un pasajero del Titanic sobrevivió o no basado en sus características. Algunos de los modelos de clasificación que existen son la regresión logística, Naive Bayes, K-Nearest Neighbors, Support Vector Machines, árboles de decisión y Random Forest (Wolff, 2022).

Definición de los modelos

Para decidir el modelo de clasificación que se utilizará, es importante conocer los datos que utiliza y los resultados que podemos obtener de ellos. Por esto se listan a continuación las características principales de cada modelo (Wolff, 2022).

- **Regresión logística:** Este modelo puede recibir parámetros categóricos o numéricos, pero el resultado que dará siempre será categórico. El resultado será la probabilidad de que el elemento pertenezca a una categoría, en una escala del 0 al 1.
- **Naive Bayes:** Así como la regresión lineal, el modelo Naive Bayes busca dar la probabilidad de que un elemento sea parte de la clase o no. Es un modelo que puede hacer predicciones multiclases.
- **K-Nearest Neighbors:** En el modelo de K-Nearest Neighbors, un elemento se clasifica según las categorías de sus 'k' vecinos más cercanos en el conjunto de datos. La elección de 'k' determina cuántos vecinos considerar. Es simple y efectivo, pero la elección incorrecta de 'k' o una gran cantidad de datos puede afectar su rendimiento.

- **Support Vector Machines:** Este modelo genera hiperplanos que dividen a los datos en categorías. Lo que esté de un lado del plano es de una categoría y lo que esté del otro lado será de la segunda categoría. Es un modelo que se puede adaptar bien a sets de datos grandes ya que es multidimensional.
- **Árboles de decisión:** Este es un modelo que permite hacer una clasificación sencilla, permitiendo tener clases, sin la necesidad de mucha supervisión humana. Este algoritmo funciona tomando decisiones en cada capa, dependiendo del valor que una variable tiene, hasta llegar al final del árbol, donde obtendremos predicción de la clasificación.
- **Random Forest:** Es una expansión de los árboles de decisión. Los Random Forest generan promedios para los datos y así generan las conexiones. Resuelven el problema de los árboles de decisión a forzar que las variables sean categóricas.

Como podemos ver, existen diferentes modelos que son más efectivos que otros en diferentes casos. Además, presentan limitaciones en las entradas que pueden tener o en la cantidad de clasificaciones de salida que soportan. Para determinar el modelo que mejor se ajusta a nuestros datos, se harán pruebas con 6 algoritmos diferentes y se comparan la precisión y el F1 de cada modelo para decidir cuál es el mejor.

Pruebas de modelos

Para entrenar los modelos, utilizaremos los datos proporcionados en el archivo de train.csv. Solo utilizaremos el 80% de los datos para el entrenamiento, porque el 20% restante se reservará para evaluar, validar y probar su rendimiento en datos no vistos previamente. De esta manera, nos aseguramos de tener una forma de medir qué tan bien el modelo puede predecir datos nuevos, sin que se “memorice" solo lo que ya conoce.

Selección de los modelos

Una vez que analizamos nuestros datos y entendimos las características del modelo que necesitamos, realizamos pruebas con 6 tipos de modelos distintos que funcionan para

problemas de clasificación como el que estamos trabajando. Los modelos que probamos fueron los siguientes:

- Regresión Logística
- Random Forest Classifier
- Support Vector Classifier
- Extreme Gradient Boosting
- Extra Trees Classifier
- Gradient Boosting Classifier

Para cada uno de estos modelos, se entrenó el modelo con los datos de entrenamiento, realizamos las pruebas con nuestros datos que reservamos de prueba y finalmente obtuvimos la matriz de confusión y el valor de la exactitud ("accuracy") para analizar el desempeño de cada uno.

Los algoritmos que obtuvieron los mejores resultados de exactitud fueron los siguientes:

1. Logistic Regression (0.865)
2. Gradient Boostinger Classifier (0.8426)
3. Extreme Gradient Boosting Classifier (0.831)

Sin embargo, decidimos que para tener un mejor desempeño, en donde se pueda intentar tener una mayor exactitud y precisión con los resultados, sería una buena idea usar alguna especie de combinación en donde se tomen en cuenta todos los modelos con mejor desempeño, esto nos permitiría tomar en cuenta diferentes algoritmos.

Para lograr esta combinación, decidimos utilizar un Voting Classifier, el cual es un método de ensamblaje que combina las predicciones de varios modelos de aprendizaje automático para tomar una decisión final. Esto nos permite mejorar la precisión y la robustez del modelo.

Después de entrenar al voting classifier podemos ver que tiene más overfitting que el modelo de logistic regression. Esto se observa porque tiene una diferencia más alta entre la precisión de testing y la de training. Sin embargo, el modelo de logistic regression tiene cambios muy altos entre cada iteración si dejamos que sea aleatorio, por lo que sacrificamos un poco de overfitting para tener un modelo más estable.

Referencias

Wolff, Rachel. (2022). *5 Types of Classification Algorithms in Machine Learning*. Estados Unidos: Monkey Learn. https://monkeylearn.com/blog/classification-algorithms/