Proyecto Inteligencia Artificial

Santiago Alejandro Aristizabal Martínez

Pontificia Universidad Javeriana

Link a github: https://github.com/Santy4001/Proyecto-IA-

Introducción:

En el presente informe se mostrará el diseño de un algoritmo de inteligencia artificial, que clasificará una partida del juego league of legends, determinando si el equipo azul sale o no victorioso dados los primeros diez minutos de juego, este resultado se determina mediante datos tomados de mas de diez mil partidas usando como características ciertos datos de los primeros diez minutos de juego y como etiqueta la victoria del equipo azul al final del juego.

Manejo de datos:

Se tiene un dataset con mas de diez mil partidas de rangos altos y alrededor de treinta y nueve características para determinar si el equipo azul gana o pierde, hay cierto número de datos que pueden ser fácilmente omitidos ya que se tiene una gran correlación entre datos, como por ejemplo si el equipo azul realizó 15 muertes, se sabe que el equipo rojo sufrió 15 muertes. Por este tipo de datos se pueden reducir varias características del dataset, por último, se define la característica objetivo como "blueWins" que será 1 si el resultado es una victoria o 0 si se produjo una derrota, dándonos así un problema de clasificación binario, que fue resuelto mediante tres algoritmos de inteligencia artificial distintos, los cuales son: KNN, Logistic Regresion y Random Forest, para probar que tan útiles son para nuestro problema de aprendizaje de máquina.

Random Forest:

Realizomos el modelo de Random Forest con la ayuda de pycaret y posteriormente ajustamos los hiperparámatros en búsqueda de una mejora en el coeficiente de correlación de Matthews (MCC por sus siglas en inglés), luego de haber creado el modelo se realiza una interpretación de este que nos indica cuales son aquellas características con mayor relevancia para la determinación de un resultado, seguido de esto realizamos la evaluación del modelo que nos dará diferentes tipos de información como por ejemplo la matriz de confusión que se tuvo con el modelo o el AUC que presenta el modelo, por último realizamos predicciones con el modelo con el conjunto de datos usados para test y vemos que tenemos resultados buenos como un MCC de 0.5 indicando que la correlación que hay no es la mejor pero es buena, o valor de AUC de 0.8114 el cual sugiere que el modelo tiene un buen rendimiento en términos de discriminación entre clases.

Logistic Regresion:

Se puede considerar que el punto fuerte de la regresión logística son los problemas binarios, y se reafirma en el código cuando vemos que entre la gran mayoría de modelos es el que mejor rendimiento tiene. Igual que con Random Forest se crea el modelo con la misma

función y se hace una mejora al modelo finalmente podemos ver sus puntajes en la predicción destacando un valor de 0.7341 de recall, indicando que el modelo de Regresión Logística captura el 73.41% de las instancias positivas, y un valor de precisión de 0.7308 que indica que el 73.08% de las instancias clasificadas como positivas son realmente positivas.

KNN (K-Nearest Neighbors)

Para realizar KNN se realizó el mismo procedimiento que en los dos modelos pasados con la diferencia ahora en el código que se hizo un pequeño agregado el cual muestra una gráfica sobre cuantos clusters es mejor usar para realizar el modelo, en este caso se obtuvo que 2 clusters es lo mejor para trabajar el problema, el resto se hizo igual a lo anterior y para los puntajes del modelo también se obtuvieron resultados que demuestran que el valor es útil para resolver el problema.

Prueba con un dataframe creado:

Para finalizar se realiza un dataframe con diferentes datos para las características del modelo y se prueba en cada uno de ellos, el resultado de todos los modelos es el mismo, dando como resultado una victoria para el equipo azul, lo que diferencia a los modelos es el puntaje de predicción que dan los cuales son 0,71, 0.78 y 0.75 para los modelos de Random Forest, Logistic Regresion y KNN respectivamente, con esto podemos determinar que Logistic Regresion es ligeramente mejor al resto de modelos pero todos funcionan y se acoplan al proyecto de una buena manera.

Conclusiones:

- Aunque la regresión logística sea buena para problemas binarios si se llega a agregar un valor más de resultado a las etiquetas podría disminuir en gran manera el rendimiento del modelo por ejemplo si existiera el empate tal vez el modelo no serviría tanto como en este momento para nuestro problema, mientras que KNN y Random Forest son más constantes.
- Los algoritmos de Machine Learning aunque muy útiles, no pueden definir por completo una partida ya que siempre puede suceder algo inesperado luego de los diez minutos de juego, siendo así que no se puede confiar plenamente en la respuesta del modelo como un resultado final.
- El preprocesamiento de los datos ayuda mucho para mejorar la velocidad y rendimiento de nuestros modelos por lo cual es muy útil realizarlo para nuestros datos reduciendo la cantidad de características.