



udp UNIVERSIDAD
DIEGO PORTALES

Facultad de Ingeniería y Ciencias
Escuela de Informática y Telecomunicaciones

Tarea 2 [Avance]

Inteligencia Artificial

CIT-3346

Nicolás Flores Sepúlveda

Profesor: Martín Gutiérrez

Ayudante: Javier Carrión

Mayo, 2019

1. Introducción

En esta tarea se pidió crear y comparar dos algoritmos clasificadores/predicadores. En los cuales se debe evaluar la eficiencia y efectividad de ambos.

En una primera instancia se buscó un dataset acorde para poder clasificar y predecir los datos con los algoritmos creados.

Luego se utilizó un algoritmo de clasificación para agrupar los datos del dataset.

Posteriormente se utilizara un algoritmo de predicción, para saber en qué grupo se asigna un nuevo dato.

2. Desarrollo

2.1. DataSet

Para realizar esta tarea se eligió un dataset que se encuentra en la página web *kaggle.com*.

El dataset seleccionado contiene datos sobre algunas partidas competitivas del videojuego *League of Legends*, en donde se especifican los siguientes datos:

1. Id de la partida
2. Duración de la partida
3. Id de la temporada
4. Ganador (1 = team1, 2 = team2)
5. Primer Barón, dragón, inhibidor, Herald, torre eliminada y primera sangre realizada (1 = team 1, 2 = team2, 0 = ninguno)
6. Campeón y hechizo elegido por cada jugador (Asignadas según la ID de los campeones y hechizos definidos por Riot)
7. Número de torres, inhibidores, Barón, dragón y Heraldos eliminados por cada equipo.
8. Los 5 bloqueos de campeones realizados por cada equipo.

Luego de seleccionar el dataset, se creó un gráfico de 3 dimensiones, en donde los ejes del plano tridimensional fueron los siguientes:

- Cantidad de inhibidores destruidos por el equipo 1.
- Cantidad de barones eliminados por el equipo 1
- Cantidad de torres destruidas por el equipo 1.

Estos datos fueron seleccionados debido a que para ganar una partida se deben destruir algunas torres e inhibidores enemigos, por lo que al destruir una mayor cantidad de estos objetivos la probabilidad de ganar una partida aumenta. Por otro lado el Barón es un monstruo poderoso que otorga un bonus de daño al equipo que lo mata.

Además, se asignó un color a cada partida según el equipo ganador. El color Azul indica que el equipo 1 gano la partida. En cambio el color Rojo indica que el equipo 2 fue el vencedor.

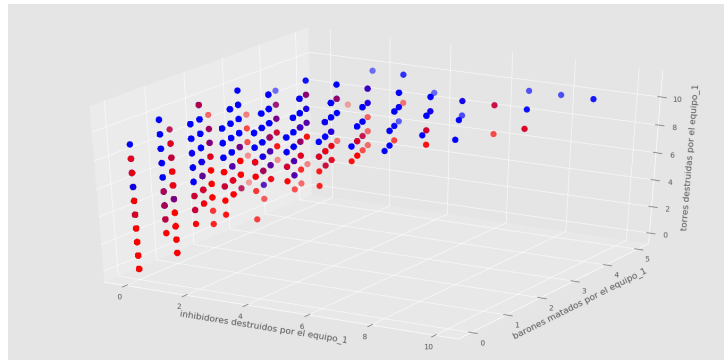


Figura 1: Graficos del dataset con 3 variables

2.2. Clustering

Para realizar el clustering se utilizó K-means sobre los datos, con $K=2$, para que de esta forma se creen dos grupos. En un grupo estarán las partidas que gano el equipo 1, el cual se destacara con el color Azul y en el otro estarán las partidas que gano el equipo 2 (Rojo).

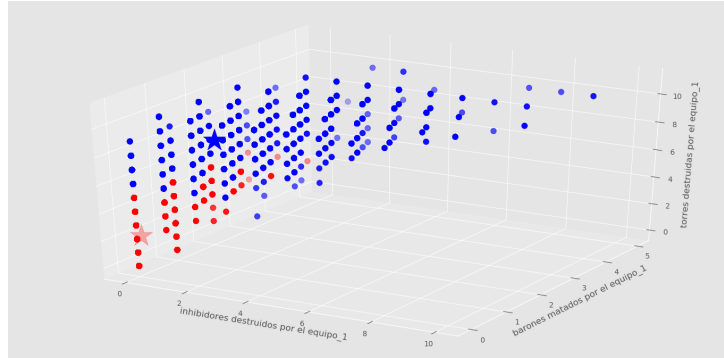


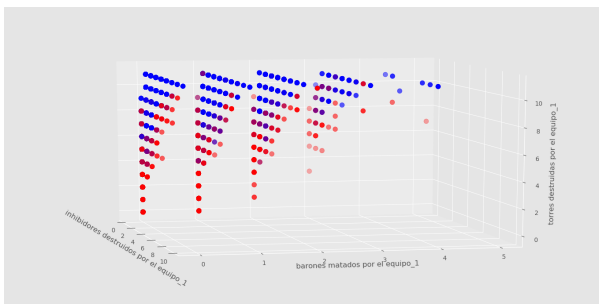
Figura 2: Grafico al utilizar K means

Los centroides de cada grupo se ubican en las posiciones :

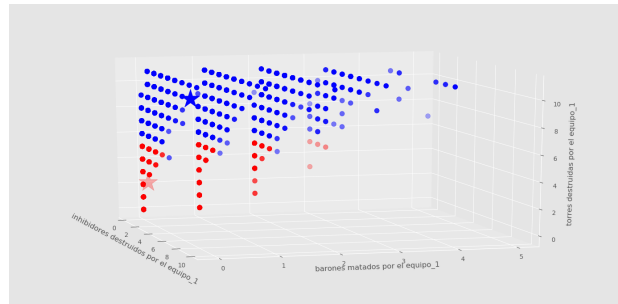
- Azul: (1.92156278, 0.63432001 , 8.97696093)
- Rojo: (0.03441985, 0.08732668, 2.13500365)

De la agrupación realizada por el algoritmo K-Means, se puede comprobar que entre más objetivos destruya o elimine el equipo 1 (Torres, Inhibidores y Barones), es más probable que el equipo 1 gane la partida.

Por esta razón el grupo Azul se encuentra en la parte superior del gráfico. A diferencia del grupo rojo que se encuentra en la parte inferior, el cual indica que el equipo 2 gano la partida.



(a) Gráfico del dataset



(b) Grafico al realizar K means

Figura 3: Comparación entre los graficos

2.3. Predicción

2.4. Clasificación/Predicción

2.5. Descripción del algoritmo

El algoritmo lee un archivo csv en donde se encuentra el dataset que se utilizara.

Luego se crea un gráfico de 3 dimensiones con los ejes: “t1-inhibitorKills”, “t1-baronKills” y “t1-towerKills”.

Cabe destacar que se utiliza el parámetro “winner” como etiqueta de los datos, el cual permite separar los datos en 2 grupos.

Posteriormente se utiliza K-Means con $k=2$, para que el algoritmo pueda agrupar los datos, en dos grupos.

2.6. Observaciones del Algoritmo

El algoritmo esta escrito en python 3.5.4 y para que funcione se debe tener instalado las siguientes librerias: pandas, numpy, matplotlib, KMeans, pairwise-distances-argmin-min y Axes3D

3. Conclusión

4. Referencias

<https://www.kaggle.com/datasnaek/league-of-legends>

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

5. Anexo

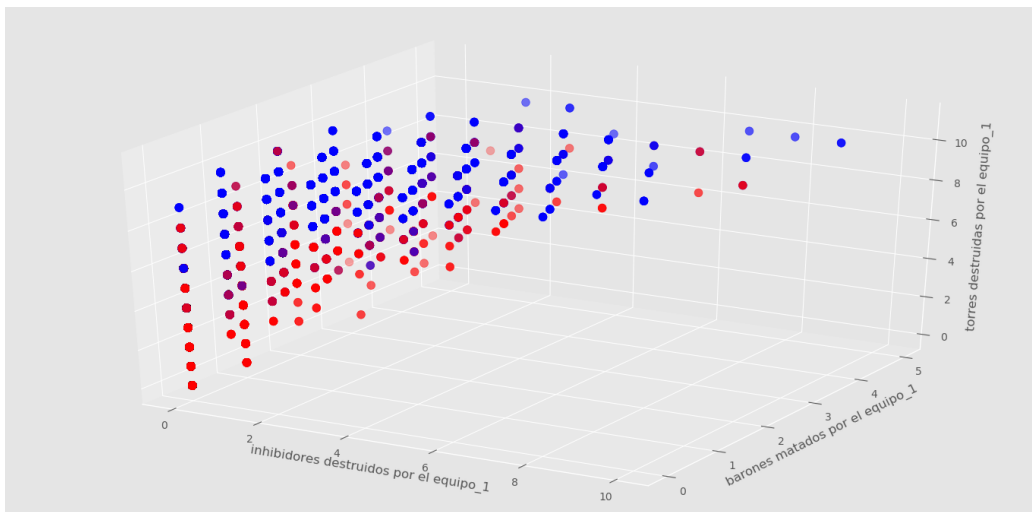


Figura 4: Gráfico del dataset con 3 variables

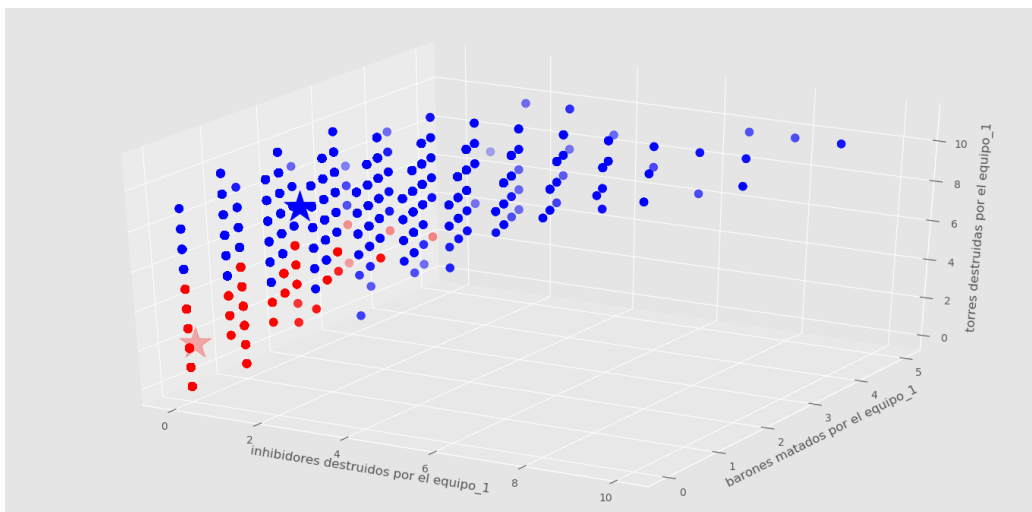


Figura 5: Gráfico al utilizar K means

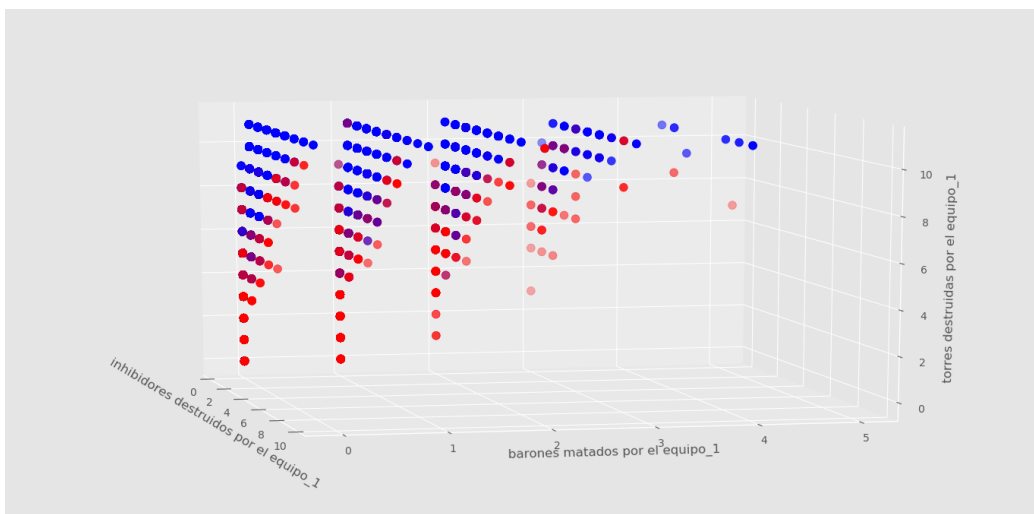


Figura 6: Gráfico del dataset

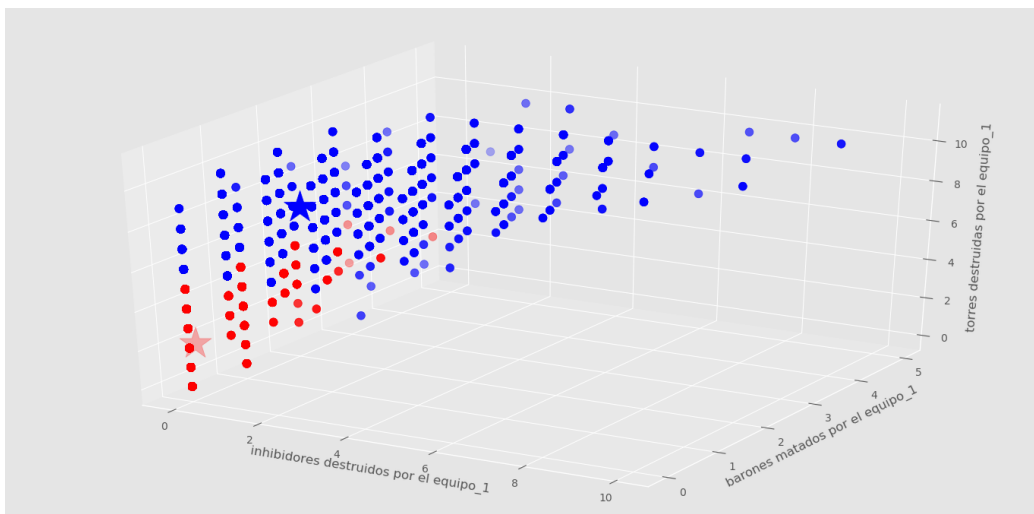


Figura 7: Gráfico K-means