

Miércoles

* > Anaconda > Jupyter Lab

- Se corrió el Queri del Join entre Stats, Goles - casa y Goles - visitante
- Se eliminaron las columnas duplicadas

~~Presentación~~

~~Presentación~~

~~Presentación~~

- Hacer presentación y la ~~histórica~~ linea del tiempo de la presentación.

- Tablas de Tableau

Antes se probaron 3
modelos
núméricos
núm y cat con num_norm
núm y cat con num_stand

- Tenemos un dataframe grande (el completo)

- Entrenamos el modelo
lineal (en 2006 - 2017)

- Con este mismo modelo
hacemos el test para
ver si no se sobre
ajuste y si sirve para
hacer predicciones

se eligió el modelo con
la data standarizada por

TRAIN

TEST

- ~~Presentación~~

la ² más grande .75

CROSS - VALIDATION

mientras norm .73

- Datos de temporadas
(df3) 2017 - 2018 que no conoc

para hacer la predicción
de wins y compararlos con las
verdaderas wins.

- ~~Presentación~~

~~Presentación~~

* Se hizo el modelo > lo entreno
> lo testeé (x_norm, x_stand, categorico)
> cross-validation con última temporada

Norma

1 MANTES Pasos

2, como poner
los datos correctos
en el GITHUB

- Obtener la información (Buscar base de datos, Kaggle)
- Descargar ~~zip~~ (2 bases de datos)
- Comenzar a familiarizarse con las columnas y su significado.
- Fueron 2 bases de datos:
 - * Results:
 - Resultados de cada partidos de las últimas 12 temporadas (38 jornadas y 20 equipos, es decir 10 partidos por jornada, $10 \times 38 = 380$ y por las 12 temporadas $380 \times 12 = 4560$ partidos)
 - * Stats:
 - Las estadísticas por equipo de las 12 temporadas (20 equipos \times 12 temporadas = 240 registros).
 - ~~Results~~ MySQL (datos en CSV)
 - * ~~Create~~ la Base de datos Premier League
 - * Crear tablas:
 - * resultados
 - * stats
 - * Cargar información desde los CSV
 - * De la tabla de resultados extraer el número de goles en casa y de visitante por equipo y por temporada.
 - * JOIN las tablas de goles en casa y visitante de cada equipo por temporada con la tabla de stats usando (using) team y season
 - * Llevarme el resultado de este query a python y empezar a conocer y trabajar con él.
 - * Borrar las columnas irrelevantes de team y season a partir del index

Atorando el
gol de los
goles - una
columna de
la tabla

②

Jueves

- Python

> Anaconda > Jupyter

> Importar bibliotecas

> Conexión con SQL

> Query donde se hace JOIN STATS con Goles_casa y Goles_vis

> Eliminar columnas duplicadas (team y season)

> Ver la correlación de las variables con el target "wins"

> Nuevo DF con variables ~~correlacionadas~~ con mayor correlación (goles_casa, goles_visita, total_pcts, corner_takes, touches, season).

> ~~ver minimo, máximos, outliers~~

> Limpieza datos, mínimos, máximos, nulos, outliers

> Revisar la distribución de los datos

> Train y Test del Modelo Lineal

> DF 2 donde este de la primera a la penúltima temporada

> Dropped target "wins"

> Dividir categóricos y numéricos

> Normalizar numéricos

> Estandarizar numéricos

> Tratar categóricos "season" con OneHot Encoding

> Concatenar con X-norm o X-standardized con Encoding

> ~~split~~ Split train test

> Crear Modelos

> Entrenar modelo

> Predecir data

> R^2

~~normalizadas (X-norm)~~

$$R^2 = .75$$

estandarizadas (X-standard.)

$$R^2 = .73$$

> Elección modelo \rightarrow X-norm

> Hacer validación del modelo con los datos de la

~~última temporal.~~ (Cross-validation).

Comparar resultados de predicción ~~de~~ "wins"

con ~~las~~ las verdaderas "wins"