FACULTAD
DE INGENIERIA
Universidad de Buenos Aires

Trabajo Práctico 2

Análisis de Datos FiuMark

Segundo Cuatrimestre 2020

Fecha de Entrega: 16/02/2021

Integrantes:

Nombre y Apellido	Padrón
Anarella Nicoletta	94.551
Juan Cruz Opizzi	99.807

Introducción

El objetivo del presente informe es mostrar los resultados obtenidos al aplicar distintas técnicas de Machine Learning, sobre el set de datos provisto por FiuMark.

Tomando como punto de partida, el análisis exploratorio previo y el baseline definido para el set inicial, se aplicaron diferentes algoritmos, tanto de preprocesamiento como de predicción de datos, a fin de mejorar las predicciones.

Se buscó predecir si la persona que fue al cine a ver la película Frozen 2 regresará al mismo para ver su secuela, Frozen 3. Los datos sobre los espectadores fueron proporcionados por FiuMark, y la evaluación final de los modelos se realizó generando predicciones sobre registros para los que desconocemos si el espectador regresaría o no.

Modelos de Machine Learning

Preprocesamientos utilizados

A continuación detallamos las distintas funciones de preprocesamiento desarrolladas. Las mismas se encuentran en el archivo preprocessing.py y son reutilizadas por distintos modelos.

Nombre	Descripción	Función de Python
Dataset inicial	Genera el dataset inicial, mergeando los 2 datasets. Agrega el campo ‘volvería’ y ‘cant_acompañantes' ('parientes' + 'amigos'). Elimina ‘id_ticket', 'fila' y 'nombre'.	preprod_tp1(df_datos, df_predict)
Dataset inicial para test de holdout	Hace lo mismo que *Dataset inicial,* pero solamente con un dataFrame.	prepod_tp1_un_df(df)
Reemplazo edad (métrica)	Reemplaza los valores nulos de ‘edad’ por la moda, la mediana o media.	replace_nulls_edad(df, metrica)
Encoder	Aplica Binary Encoder a los atributos categóricos 'tipo_de_sala', 'genero' y 'nombre_sede'.	encodear_atributos_categoricos(df)
Normalización	Normaliza todos los atributos numéricos del dataFrame.	normalizar_atributos_numericos(df)

Modelos utilizados

Hemos aplicado y testeado diferentes modelos de Machine Learning, cuyos resultados procederemos a detallar. Antes, sin embargo, nos gustaría hacer algunos comentarios.

En el tunning de los modelos, utilizamos Random Search, para encontrar las mejores combinaciones de hiper-parámetros. Además, al momento de entrenar (con esos hiper-parámetros obtenidos) aplicamos Stratified K-Fold.

En el caso de KNN y el árbol de decisión, usamos Grid Search para encontrar los hiper-parámetros. En el caso de KNN, el limitar las combinaciones posibles al subset que nos interesaba fue lo que nos permitió aplicar el algoritmo. Con respecto al árbol de decisión, como es un modelo simple, rápido y con pocos hiper-parámetros, lo aprovechamos como baseline. Es decir, nos permitió analizar si

los modelos más complejos eran demasiado complejos para el caso, o si directamente estaban prediciendo muy mal.

Las métricas utilizadas para evaluar los modelos fueron:

- AUC-ROC
- Matriz de confusión
- Accuracy
- Precisión
- Recall

Nombre	Preprocesamiento	AUC-ROC	Matriz de confusión	Accuracy	Precisión	Recall	F1 Score
KNN	Dataset inicial Reemplazo edad (media) Encoder Normalización	0.9758	TP:69 FP:0 TN:127 FN:5	0.9751	1.0	1.0	0.9650
Support Vector Classifier (SVC)	Dataset inicial Reemplazo edad (media) Normalización Encoder	0.8901	TP:38 FP:4 TN:123 FN:36	0.8010	0.9048	0.9685	0.6552
Árbol de decisión	Dataset inicial Reemplazo edad (media) Encoder	0.8906	TP:57 FP:12 TN:115 FN:17	0.8557	0.8261	0.9055	0.7972
Random Forest	Dataset inicial Reemplazo edad (media) Encoder	0.8660	TP:45 FP:11 TN:116 FN:29	0.8010	0.8036	0.9134	0.6923
Red neuronal	Dataset inicial Reemplazo edad (media) Normalización Encoder	0.8362	TP:33 FP:11 TN:95 FN:22	0.7950	0.7500	0.8962	0.6667
Bagging	Dataset inicial Reemplazo edad (mediana) Encoder	0.8650	TP:43 FP:11 TN:116 FN:31	0.7910	0.7963	0.9134	0.6719
AdaBoost	Dataset inicial Reemplazo edad (media) Encoder	0.9123	TP:60 FP:11 TN:116 FN:14	0.8756	0.8451	0.9134	0.8276
GradientBoost	Dataset inicial Reemplazo edad (media) Encoder	0.9160	TP:57 FP:9 TN:118 FN:17	0.8706	0.8636	0.9291	0.8143
Voting	Dataset inicial Reemplazo edad (mediana) Encoder	0.8536	TP:57 FP:8 TN:119 FN:17	0.8756	0.8769	0.9370	0.8201

Conclusiones

En base al análisis y comparaciones realizadas, recomendamos a la empresa FiuMark utilizar el modelo de datos KNN, para predecir el regreso de su audiencia.

Este clasificador fue el que obtuvo el mejor score en las métricas evaluadas, pero a la vez sin caer en el caso de overfitting (o memorizado de datos). El hecho de complementar al modelo con acciones de preprocesado que incluyan normalización de los datos, y a la vez potenciar el entrenamiento a través de Stratified K-Fold, fueron los responsables de mejorar considerablemente la performance general de KNN.

En el análisis exploratorio previo, habíamos obtenido un accuracy de 0.82, sin aplicar ninguna inteligencia automática de predicción de datos. Considerando que tanto el accuracy como el AUC ROC de KNN valen 0.97, puede verse claramente una gran mejoría en los resultados obtenidos.

Asimismo, el baseline de la primera parte actuó como nuestro benchmark, para determinar si nuestros modelos estaban realizando una buena predicción o no, ya que si un modelo de Machine Learning no posee un score superior al que ofrecen una serie de if's anidados, entonces estamos frente a un problema. El mismo puede estar alocado en el feature engineering, en el entrenamiento, o mismo en el set de hiperparametros, entre otros.