AAGE - Práctica 1 (MLlib):

Andrés Lires Saborido[1] and Ángel Vilariño García[2]

[1] Universidade da Coruña, andres.lires@udc.es
[2] Universidade da Coruña, angel.vilarino.garcia@udc.es

Abstract. En este proyecto se presenta un problema de clasificación y predicción de retrasos de vuelos durante el año 2019, haciendo uso de un conjunto de datos extraído de Kaggle, *2019 Airline Delays w/Weather and Airport Detail* [1]. El objetivo es desarrollar un modelo de clasificación binaria capaz de estimar si un vuelo se retrasará a partir de una serie de variables operativas y meteorológicas conocidas antes de la salida. Dado el gran volumen y la heterogeneidad de los datos, se emplea Apache Spark MLlib [2] para realizar todas las fases del trabajo: preprocesamiento de datos y el entrenamiento de diferentes modelos de aprendizaje automático a gran escala. Se comparan distintos algoritmos, así como diferentes modelos gracias a combinaciones de características o hiperparámetros para tratar de obtener el mejor modelo predictivo.

Keywords: Big Data · Machine Learning · Apache Spark · MLlib · Binary Classification · Prediction · Flight Delays · Airports · Weather · Data Analysis · Data Science

1 Introducción

1.1 Dataset

El conjunto de datos elegido es *2019 Airline Delays w/Weather and Airport Detail* [1], disponible en Kaggle, plataforma gratuita. Contiene más de 6 millones de vuelos realizados en Estados Unidos durante el año 2019, con información sobre aerolíneas, el vuelo, condiciones del aeropuerto, de la aeronave y meteorológicas.

El conjunto presenta 26 variables, explicadas con mayor detalle en la Tabla 1. Las diferentes características, tanto categóricas como numéricas, pueden ser utilizadas de manera conjunta para analizar las condiciones operativas o meteorológicas que llevan al retraso de vuelos.

El número de filas del dataset justifica el uso de Apache Spark, motor de procesamiento distribuido, para el entrenamiento de modelos de aprendizaje automático a gran escala.

1.2 Definición del problema

El objetivo del proyecto es predecir si un vuelo será retrasado o no antes de su salida, utilizando la información disponible en el dataset.

Table 1. Resumen de las variables del dataset

Variable	Tipo	Descripción
MONTH	Numérica	Mes del vuelo
DAY_OF_WEEK	Numérica	Día de la semana (1 = Lunes)
DEP_DEL15	Binaria	Salida retrasada 15 min o más (1 = Retrasado)
DEP_TIME_BLK	Categórica	Bloque horario de salida
DISTANCE_GROUP	Numérica	Grupo de distancia del vuelo
SEGMENT_NUMBER	Numérica	Vuelos previos del avión hoy
CONCURRENT_FLIGHTS	Numérica	Vuelos simultáneos en el mismo bloque
NUMBER_OF_SEATS	Numérica	Número de asientos
CARRIER_NAME	Categórica	Aerolínea
AIRPORT_FLIGHTS_MONTH	Numérica	Promedio de vuelos aeropuerto/mes
AIRLINE_FLIGHTS_MONTH	Numérica	Promedio de vuelos aerolínea/mes
AIRLINE_AIRPORT_FLIGHTS_MONTH	Numérica	Promedio de vuelos aerolínea+ aeropuerto/mes
AVG_MONTHLY_PASS_AIRPORT	Numérica	Promedio pasajeros aeropuerto/mes
AVG_MONTHLY_PASS_AIRLINE	Numérica	Promedio pasajeros aerolínea/mes
FLT_ATTENDANTS_PER_PASS	Numérica	Tripulantes por pasajero
GROUND_SERV_PER_PASS	Numérica	Personal tierra por pasajero
PLANE_AGE	Numérica	Edad del avión
DEPARTING_AIRPORT	Categórica	Aeropuerto de salida
LATITUDE	Numérica	Latitud aeropuerto
LONGITUDE	Numérica	Longitud aeropuerto
PREVIOUS_AIRPORT	Categórica	Aeropuerto previo
PRCP	Numérica	Precipitación (pulgadas)
SNOW	Numérica	Nieve caída (pulgadas)
SNWD	Numérica	Profundidad de nieve (pulgadas)
TMAX	Numérica	Temp. máxima (°F)
AWND	Numérica	Velocidad máxima viento (m/s)

Se trata de un problema de clasificación binaria supervisada, donde la variable objetivo es DEP_DEL15 (valor 1 si el vuelo salió con más de 15 minutos de retraso, 0 en otro caso). Entre las variables que nos ayudarán a lograr la predicción se encuentran características del vuelo (aerolínea, origen, destino, mes, día de la semana) y condiciones meteorológicas en el aeropuerto de origen.

Para resolver el problema se emplearán distintos modelos de Spark MLlib, haciendo uso de diferentes algoritmos, características o hiperparámetros de manera justificada. Estos modelos se evaluarán de acuerdo a métricas adecuadas al problema.

2 Análisis exploratorio de datos (EDA)

En esta sección se llevó a cabo un análisis exploratorio de los datos para comprender mejor las características del dataset y la relación entre las variables. Se utilizaron técnicas de visualización y estadísticas descriptivas para identificar patrones, tendencias y posibles problemas en los datos.

El análisis completo se puede encontrar en el cuaderno de Jupyter adjunto *p1_ AndresLires_ AngelVilarino.ipynb*, aquí se recogen las conclusiones más relevantes obtenidas tras el EDA.

- El dataset contiene más de 6 millones de registros, lo que justifica el uso de Apache Spark para su procesamiento.
- La variable objetivo (DEP_DEL15) está desbalanceada, con aproximadamente un 20% de vuelos retrasados.
- Ninguna variable presenta valores nulos.
- Distribución de la variable objetivo por mes y aerolínea en Figuras 1 y 2.
- La variables meteorológicas (PRCP, SNOW, SNWD, TMAX, AWND) parecen tener una influencia significativa en los retrasos a la vista de la Tabla 2.

Table 2. Promedios de variables meteorológicas según si el vuelo tuvo retraso

Retraso	avg(PRCP)	avg(SNOW)	avg(SNWD)	avg(TMAX)	avg(AWND)
1	0.1608	0.0645	0.1309	71.1289	8.7219
0	0.0904	0.0239	0.0823	71.5477	8.2526

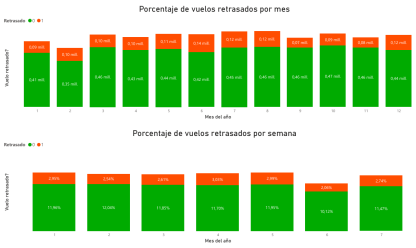

Fig. 1. Distribución de vuelos retrasados por mes y semana (Gráfica hecha en PowerBI)

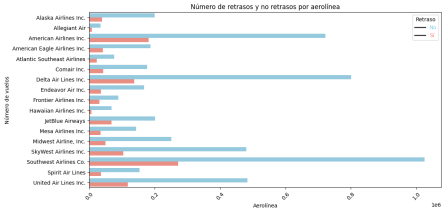

Fig. 2. Distribución de vuelos retrasados por aerolínea

3 Preprocesado de datos

En un primer lugar, se ha realizado un under-sampling del dataset para balancear la variable objetivo, reduciendo el número de registros con valor 0 en DEP_DEL15 para igualarlo al número de registros con valor 1. De esta forma conseguimos equilibrar las clases y al mismo tiempo reducir el tamaño del dataset, lo que acelera el entrenamiento de los modelos.

Se ha realizado un preprocesado general inicial (StringIndexer, OneHotEncoder, VectorAssembler) pero durante la experimentación se han realizado ajustes específicos para cada modelo, que se explicarán más adelante.

4 Experimentación

La fase de experimentación fue muy extensa, probando diferentes combinaciones de características, modelos, métodos de muestreo y métricas de evaluación. A continuación se describen los aspectos más relevantes.

En todos los casos se ha realizado un *train-test split* del 80%-20% para evaluar los modelos.

4.1 Muestreo

Durante la experimentación se probaron diferentes técnicas de reducción de filas o subconjuntos de datos:

- *Under-sampling*: Como se ha mencionado anteriormente, se realizó un undersampling para balancear la variable objetivo.
- *Stratified Sampling*: Se probó a realizar un muestreo estratificado para mantener la proporción original de clases en el conjunto de datos.

4.2 Selección de características

A lo largo de la experimentación se entrenaron modelos con diferentes subconjuntos de características. Las selecciones consideradas son las siguientes:

1. Todas las características.
2. Solo las características numéricas: De esta forma se elimina el coste computacional asociado a las variables categóricas, que se transforman en múltiples columnas tras aplicar el *OneHotEncoder*.
3. Características meteorológicas y categóricas: Estas columnas parecen, según el análisis exploratorio, estar relacionadas con la variable objetivo.
4. Características definidas por un algoritmo de selección (*UnivariateFeatureSelector*).

4.3 Métricas de evaluación

Dado que se trata de un problema de clasificación binaria, se han utilizado las siguientes métricas para evaluar los modelos:

- *Accuracy*: Proporción de predicciones correctas sobre el total de predicciones.
- *Precision*: Proporción de verdaderos positivos sobre el total de positivos predichos.
- *Recall*: Proporción de verdaderos positivos sobre el total de positivos reales.
- *F1-Score*: Media armónica entre *Precision* y *Recall*, útil cuando las clases están desbalanceadas.

Según el conjunto de datos que se utilice en el entrenamiento concreto (balanceado o desbalanceado) se dará más importancia a unas métricas u otras. En el caso del dataset desbalancado, no podemos fiarnos del *Accuracy* (ya que un modelo que siempre prediga la clase mayoritaria obtendría un buen resultado), por lo que se dará más importancia al *F1-Score*.

4.4 Modelos probados

- Regresión logística (Logistic Regression - LR)
- Máquinas de vectores de soporte (Support Vector Machines - SVM)
- Árboles de decisión (Decision Trees - DT)
- Bosques aleatorios (Random Forest - RF)
- Gradient Boosted Trees (GBT)
- Redes neuronales (Multilayer Perceptron - MLP)
- Naive Bayes (NB)

5 Desarrollo de la experimentación

El desarrollo completo de la experimentación se encuentra en el cuaderno de Jupyter adjunto *p1_ AndresLires_ AngelVilarino.ipynb*. Aquí se resumen los aspectos más relevantes y las conclusiones obtenidas en cada fase.

Primero se prueba con el dataset balanceado, entrenando los modelos LR, SVM, DT y RF (modelos clásicos para clasificación binaria) con todas las características y sus hiperparámetros por defecto. Posteriormente, como los resultados no son satisfactorios, se incluye el modelo de Gradient Boosted Trees(GBT) [3], técnica más novedosa, que suele ofrecer buenos resultados en problemas de clasificación binaria. Estos 5 modelos se entrenan con parámetros por defecto sobre los 4 subconjuntos de características definidos anteriormente.

El GBT, sobre todas las características o sobre meteorológicas + categóricas, resulta ser el mejor en términos de accuracy (64%), métrica fiable al tratarse de un dataset balanceado.

A continuación, ya que LR y SVM son sensibles a la escala se prueban a entrenar de nuevo aplicando RobustScaler, resistente a outliers o distribuciones sesgadas (presentes en nuestro dataset) [4].

Los resultados siguen sin ser buenos por lo que se decide probar con un nuevo submuestreo, que mantiene la distribución inicial del dataset desbalanceado[3]. En este caso, como se citó anteriormente, la métrica más relevante es el F1-Score. Se entrenan de nuevo los modelos GBT, LR, SVM, DT y RF, esta vez solamente sobre todas las características y sigue sin haber mejoras significativas. El GBT es en este caso también el mejor modelo, con un F1-Score del 68%.

Es en este momento cuando se decide probar nuevos modelos: MLP y Naive Bayes. El MLP ofrece un F1-Score del 73.1%, superando al GBT, mientras que el Naive Bayes no consigue mejorar los resultados anteriores.

[3] Por motivos de rendimiento este subconjunto es de menor tamaño ($\approx$ 1 millón de filas en total)

6 Modelo final

Tras toda la experimentación realizada, se ha decidido elegir como modelo final el Multilayer Perceptron (MLP) entrenado sobre el dataset desbalanceado con todas las variables, ya que fue el modelo que ofreció mejores resultados. Por último, se ha realizado una búsqueda de hiperparámetros (*grid search*) para optimizar el modelo, probando diferentes combinaciones de:

- Número de capas ocultas: `[input_size, 10, 2]` y `[input_size, 15, 5, 2]`
- Tasa de aprendizaje: `[0.03, 0.1]`
- Número máximo de iteraciones: `[50, 100]`

El mejor modelo obtenido tras la búsqueda de hiperparámetros tiene la siguiente configuración:

- Capas: `[input_size, 15, 5, 2]`
- Tasa de aprendizaje: 0.03
- Número máximo de iteraciones: 100

Con este modelo final se obtienen los siguientes resultados en el conjunto de test:

Table 3. Resultados del modelo final en el conjunto de test

Accuracy	**Precision**	**Recall**	**F1-Score**
0.8115	0.7652	0.8115	0.7298

7 Conclusiones

En este proyecto se ha desarrollado un modelo de clasificación para predecir retrasos en vuelos utilizando Apache Spark MLlib y un conjunto de datos de tamaño considerable. Tras comparar diversos algoritmos y explorar diferentes técnicas de preprocesado o incluso subconjuntos de datos para tratar de obtener un buen modelo predictivo, el Multilayer Perceptron (MLP) obtuvo el mejor rendimiento, con una accuracy del 81% y un F1-Score de 0.73, demostrando una buena capacidad predictiva.

El uso de Spark permitió manejar eficientemente grandes volúmenes de datos y aplicar diferentes estrategias de muestreo y selección de características. Aún así, la naturaleza aleatoria de los retrasos y el desbalanceo de clases suponen limitaciones relevantes.

Como trabajo futuro, sería interesante probar a combinar diferentes conjuntos de datos. La exploración de modelos y técnicas ha sido bastante exhaustiva, pero podrían incorporarse nuevos datos que puedan complementar la información disponible, para tratar de modelizar mejor la complejidad del problema.

References

1. Página de Kaggle del dataset, https://www.kaggle.com/datasets/threnjen/2019-airline-delays-and-cancellations/data, descargado el 08/10/2025.
2. Página sobre MLlib en la web de Spark, https://spark.apache.org/mllib/
3. *Comparison of 14 different families of classification algorithms on 115 binary datasets*. arXiv preprint arXiv:1606.00930 (2016), https://arxiv.org/abs/1606.00930
4. Comparación de diferentes métodos de escalado, https://machinelearningmastery.com/minmax-vs-standard-vs-robust-scaler-which-one-wins-for-skewed-data/