Flight delay predictor

Maneesha Tejani (mtejani@umass.edu), Sahan Reddy (spodduturire@umass.edu)

Abstract

The overall goal of our project is to develop a flight delay predicting system that tells its users the probability of a delay for a particular airline at any given time of the year. For this purpose, we aim to use Spark ML module provided by Spark to train a Machine Learning model that can learn from past trends and extrapolate them to future. We choose Spark ML to do this machine learning task rather than any other off the shelf machine learning libraries such as TensorFlow or PyTorch firstly because we want to exploit the benefits that a distributed system gives us to analyze how increasing the number of nodes in a cluster affect our training times and secondly because Spark provides a very sophisticated way to train machine learning models that is faster and efficient than if we were to train on a single machine. The reason for this is that spark uses RDDs that not only allow nodes to benefit from shared storage but also make use of shared memory. RDDs allow memory to be shared across clusters and since machine learning training is an iterative task where the input of a previous iteration goes into the next iteration, shared memory on top of shared storage greatly enhances training time. After training different machine learning models on our dataset such as Linear Regression, Decision Trees and Random Forests, we aim to evaluate all these models using some performance metric such as accuracy and root mean squared error and choose a final model for our predicting task. We believe the opensource Flights Delay dataset provided by Kaggle which contains airline delays for more than 300 domestic airlines in US shall provide a good start to our project and shall train the model sufficiently well to work well in the real world.

1. INTRODUCTION

Airline delays can cause unnecessary disruptions in people's lives. The problems caused by flight delays include missing important business meetings or failing to catch another connecting flight which can lead to further delays. This could result in additional costs as people are stuck at the airport for an extended period of time and have to make more purchases at the airport as a result. To add to this, the previous year for airlines in the United States was chaotic with more than 22% of all flights suffering from delays or cancelations, with some carriers having even more issues. Moreover, it is not a commonly known fact that airlines are actually obliged to post flight delay data on their website. But as the holiday season looms, airlines are failing to meet this obligation which could easily help passengers avoid flights with a history of delays and cancellations. The primary reason for this is that airlines need enough people to book all the flights they operate to keep them profitable and there is a risk that if passengers see a flight as unreliable they will steer clear of it. It would be much easier for people if they had information as to whether a particular airline had a higher chance of delay compared to a flight from a different airline travelling around the same time.

Our project is designed to help solve this problem. Through our model, we aim to predict whether a flight will be delayed on a particular day of the month based on numerous other possible factors which we feel contribute to flight delays. This way, customers can choose a different airline if they realize that their current flight has a higher delay. Through this model, we also think we can get an estimate of the days of weeks during a particular month of the year where there is a higher chance of

flight delays.

There were some questions that we needed to answer before we could start working on our project. We first had to decide what type of project was best suitable to tackle the flight delay prediction problem in order to narrow down a list of frameworks which were most applicable to our situation. We noticed that there are a significant number of machine learning models that people had come up with in the past in order to tackle prediction problems similar to our use case. These models were not only quick to train but also highly accurate. Thus, we decided on training a machine learning model to predict flight delays.
Secondly, we had to figure out the scope of our project (whether we need a UI?). We decided on going with a backend application which takes in a massive dataset and is able to efficiently processes the data and output highly accurate predictions. Appending any new data in the future to this training data would be very simple. We also felt that given the time constraints, this would be sufficiently challenging to implement even without a UI.

To go about our project, we first needed to decide on how we were going to obtain enough relevant data which we can use in order to predict flight. We have decided to go with an open-source flights dataset which is easily available on Kaggle. This dataset contains around 2 million datapoints. This is good for us as our model will be able to extract more information and find more available patterns from the dataset which will help make accurate predictions. Instead of burdening our local machines with this data, we decided to store this dataset remotely using Amazon's S3 service offered by AWS that provides object storage through a web service interface. We decided on S3 because it is not only scalable but also provides several security and fault tolerance features which we felt would make our data more secure. The web interface is also very easy to interact with and make any important changes on the fly.

Our model obtains data from the S3 bucket and then preprocesses it before training the machine learning model. The preprocessing step is very important in our model as it significantly reduces the amount of data that our model has to filter through during the training process by eliminating a lot of redundant columns from our training set; as well as some columns whose types are incompatible with the model type.

We then decided to utilize Spark in order to train our model. Spark is designed for fast, interactive computation that runs in memory, enabling faster machine learning processes. The Spark ML library has abundant machine learning algorithms that are ready to integrate with available datasets. Spark significantly improves our training speed as it parallelizes the training task amongst multiple worker threads utilizing more of the CPU's resources. We did some experiments to find the optimal number of threads to run our program on. We then evaluated the performance of our model by looking at the Root Mean Squared Error that is outputted the model.

2. BACKGROUND

We used Amazon S3 to store our dataset. Amazon Simple Storage Service (Amazon S3) is an object storage service that offers very high scalability (as it uses the same scalable storage infrastructure that Amazon.com uses to run its e-commerce network) and data availability along with security and fault tolerance measures. Amazon S3 stores data as objects within buckets. An object is a file and any metadata that describes the file. A bucket is a container for these objects. S3 makes it easy to configure access to data within these buckets.
Apache Spark is a powerful, open-source tool for large-scale data processing and modeling. Spark provides an interface for programming clusters with implicit data parallelism and fault tolerance. Spark consists of a master which automatically parallelizes and divides the work among multiple worker threads which significantly amplifies processing speeds. PySpark allows us to write spark applications using Python APIs. It provides a built-in machine learning library (MLlib), which can be used to build scalable machine learning models such as the one we have designed on our application.
Regression is a statistical method that is used to analyze and understand the relationship between two or more variables of interest. Regression helps to understand which factors are important, which factors can be ignored, and how they are influencing

each other. We are training a Regression model here by predicting a number (Flight Delay) based on the features present in our dataset. We then evaluate how close we are to the actual value.

3. RELATED WORKS

There have been models in the past which have worked on the same problem. But there are slight differences in the approach taken in training those models. There have been efforts to build an accurate flight delay prediction model using Deep Learning. An example we looked at was by Maryam Farshchian Yazdi, Seyed Reza Kamel, Seyyed Javad Mahdavi Chabok & Maryam Kheirabadi - Flight delay prediction based on deep learning and Levenberg-Marquart algorithm. There have also been cases where classification algorithms were used in order to predict whether a flight would be delayed or not but that would not be applicable to our case. Yuemin Tang - Airline Flight Delay Prediction Using Machine Learning Models. These papers and others helped build a foundation for our project. Also, by looking at some of the observations and problems faced in other research papers, we were able to avoid similar situations while working on our own project.
We then trained the model using Spark ML.

4. TECHNICAL APPROACH

4.1 AMAZON S3

We start by first downloading the Kaggle Airline Flight Delay Dataset which is available open source. We then decided to store the dataset on an Amazon S3 bucket. Storing objects on an S3 bucket gives us numerous advantages as it allows us to enable and utilize numerous security and fault tolerance features that are inbuilt into S3. We enabled bucket versioning for our S3 bucket which allows us to keep multiple variants of an object on the same bucket. This allows us to recover more easily from both unintended user actions and application failures as it is easy to retrieve and restore every (previous) version of every object stored in your buckets. This comes in handy when we append new data to our dataset as we can backtrack to previous versions. S3 also contains a default encryption which we have enabled. The objects are encrypted using server-side encryption with either Amazon S3 managed keys (SSE-S3) or AWS KMS keys stored in AWS Key Management Service (AWS KMS) (SSE-KMS). This is not too handy for us at the moment because the dataset that we're using is open-source; but we are planning on adding new attributes to the dataset in the future which will help our predictions even more along with some of our source code. We have also enabled object lock on the S3 bucket which utilizes a write-once-read-many (WORM) model to store objects and help prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. This adds an extra layer of fault tolerance to our data. We retrieve the data from S3 using a single access token. We have created an IAM Administrative User for our bucket which has full control over all S3 operations; and is assigned the Access Token. We have created a python script to retrieve the data from S3 using this access token.

4.2 DATA PREPROCESSING

The dataset retrieved is stored in a pandas DataFrame where preprocessing steps are executed. Several attributes are removed from the DataFrame which are redundant for predicting flight delays. Some attributes together form another attribute in the dataset. For example, in our current dataset "DepTime" and "CRSDepTime" are used to calculate "DepDelay". In such cases, only the resulting column was kept, and the other columns were removed. We noticed that some of the features had categorical values. Since we were training a regression model, we had to convert categorical columns to numeric values so that those features could be used in our model. A sample of the code to do this is present below.

```
df["Origin"].replace(df["Origin"].unique(), range(len(df["Origin"].unique())), inplace=True)
```

Finally, any duplicates present in the data after performing all the above steps were also removed. This led to eliminating nearly 500,000 datapoints from our dataset.

4.3 SPARK ML

We then decided to use Spark in order to train a regression model on the preprocessed data. Spark was used as it parallelizes the training task among multiple workers and significantly speeds up the training time of our model. Spark ML's LinearRegression algorithm was chosen to train our

model. We had initially tested out with DecisionTreeRegressor, RandomForestRegressor and GBTRegressor models. We noted that the performance of the linear regression model for our current predictions was significantly better than the other models in terms of accuracy as well as training time.

$$Y_i = f(X_i, \beta) + e_i$$

We set the Spark job to run locally with 4 cores to train our linear regression model. The pandas DataFrame which contains our data is converted to a Spark DataFrame and then passed on to the Spark job for training the model.

We then used Root Mean Squared Error (RMSE) in order to evaluate the performance of our model. In this way, we are able to accurately predict flight delay based on the data present.

5. EXPERIMENTS

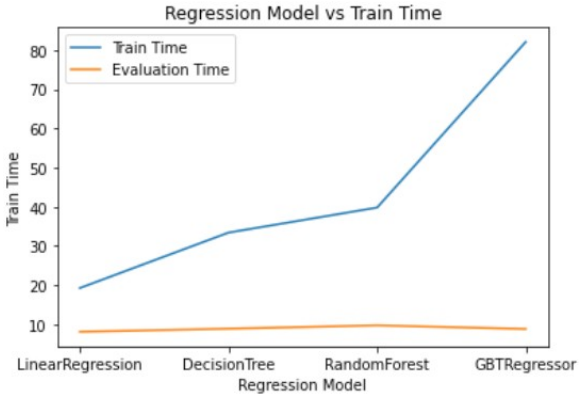
There were a couple of major tests we had to perform in the process of coming up with an accurate and quickly trainable Regression model to predict flight delays.

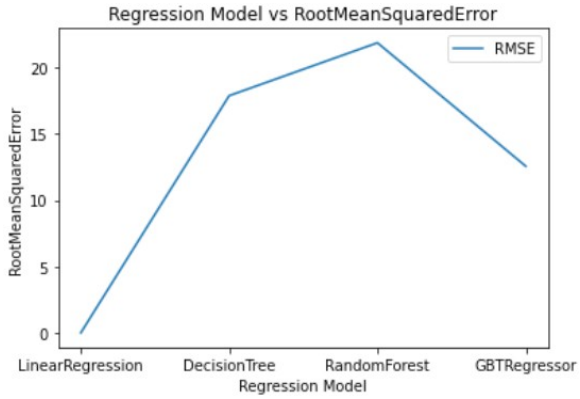
5.1 REGRESSION ALGORITHMS

We tried training our model using many different Regression algorithms before finally choosing to use LinearRegression. The key factors which we used to decide on a go-ahead model was i) the amount of time taken to train the model and ii) the accuracy of the models predictions which we gauged by comparing the RootMeanSquaredError outputted by the different models trained using these algorithms.

Our main goal was to find a tradeoff between these two factors and decide on model based on the results.

LinearRegression (Train Time = 19.28382134437561, RMSE = 1.14974e-13), DecisionTreeRegressor (Train Time = 33.42856192588806, RMSE = 17.8889),





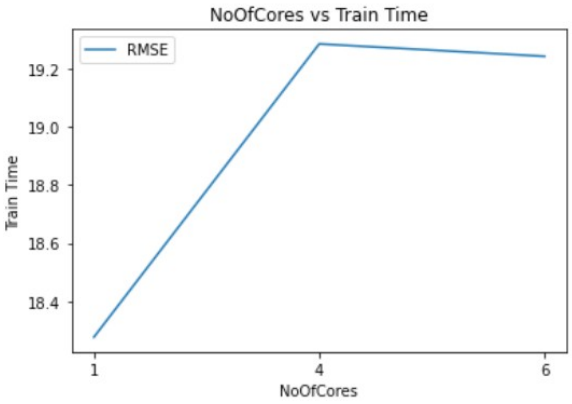
RandomForestRegressor (Train Time = 39.82293224, RMSE = 21.8782), GBTRegressor (Train Time = 82.12162017822266, RMSE = 12.5626)

The linear regression model outputted the lowest training time as well as the lowest RMSE. Thus, we decided to go with the LinearRegression algorithm to train our model.

5.2 SPARK CORES

After we figured out that we were going to be utilizing the LinearRegression model. We then decided to go about running some tests on the optimal number of cores which we want the Spark Job to run on. We tried three different configurations (1,4,6 cores).

(1 core – 18.2373, 4 cores – 19.286432342,
6 cores – 19.21748392)



We found that there is very little difference between the Train Time. Considering that we might do further operations in the future to optimize our model and add additional functionality, we decided to go with 4 cores instead of 1 as it will assist us in the future.

6. CONCLUSION

Overall, we were able to build a successful flight delay prediction model that accurately predicts late aircraft delay times based on the information that is available to us. This is very useful for future customers who are planning on leveraging such results when booking flights in the future. We were honesty surprised that LinearRegression was the most accurate out of all the models in predicting flight delay values. The Spark Job tests which we ran with different number of cores (1,4,6) also gave us some valuable insights. The number of cores did not affect train time too much for training LinearRegression or DecisionTreeRegressor models; however, the power of increased cores was at full display while training the RandomForestRegressor and GBTRegressor models. This information can be useful in the future when we are running Spark ML jobs using the latter two models.

Our model currently only predicts the flight delay values. In the future, we are planning on adding functionality to predict a stretch of days or weeks during the year when there is a high chance of delay.

We don't have a timeline for this update as of yet, but we will be working on it. Another major change which we would like to do in the future is run all our jobs remotely rather than on a local machine. We have looked at Amazon EC2 and AWS Lambda as possible platforms we can explore to go about this process. We are currently in the process of comparing storage costs and various functionalities that are present within the two services before making a final decision as the functionalities might come in handy in the future as we further enhance our code.

7. TEAM CONTRIBUTIONS

There was a ton of efforts put into getting this project to run smoothly. A lot of discussions were held, and ideas were put forth by both of us before we decided to work on the Flight Delay Prediction problem. After choosing to work on this problem, we initially searched the web for different datasets. Some of the problems we faced was that i) either the dataset was too small or ii) there were too many columns that we felt were redundant for our calculations. Therefore, we decided to go with the Kaggle Dataset.

Combined – Business decisions made as to which columns are redundant. Research on Amazon S3 and integrating with EC2 and Lambda for future.

Sahan – Set up the initial Amazon S3 bucket. Enabled Bucket Versioning, Object Locks. Set up an IAM User with Secret Access Key to the S3 bucket. Inserted Dataset into bucket through the S3 User Interface. Created the **AuthCredentials.py** file in order to retrieve data from S3 using the IAM User's secret access key. Created **data-preprocessing.py** file in order to work on retrieved data and eliminate lots of redundant features from the dataset.
Ran tests to find the optimal number of Spark cores for the model.

Maneesha – Created **FlightDelayLinearRegresion.py** file. Setup a local Spark program which takes in the preprocessed dataset and then uses Spark ML's LinearRegression algorithm to train our ML model. Ran tests with different models – DecisionTreeRegresor, RandomForestRegresor, GBTRegressor. Plot graphs based on the results outputted from the different models.

8. REFERENCES

Alison Fox. 2022. The airlines with the most delays this year, according to the Bureau of Transportation

Statistics. (October 2022). Retrieved December 13, 2022 from https://www.travelandleisure.com/most-delayed-airlines-2021-2022-6814429

Matt Clark, Amani Wells-Onyioha, and May Mailman. 2022. Airlines are keeping flight delay data hidden as holidays loom. (November 2022). Retrieved December 13, 2022 from https://www.newsweek.com/warning-airlines-are-keeping-flight-delay-data-hidden-holidays-loom-1760946

Giovanni Gonzalez. 2019. Airlines delay. (November 2019). Retrieved December 13, 2022 from https://www.kaggle.com/datasets/giovamata/airlinedelaycauses

Aveek Das. 2021. Getting started with Amazon S3 and Python. (April 2021). Retrieved December 13, 2022 from https://www.sqlshack.com/getting-started-with-amazon-s3-and-python/

Yuemin Tang. 2021. Airline Flight Delay Prediction Using Machine Learning Models. In 2021 5th International Conference on E-Business and Internet (ICEBI 2021), October 15-17, 2021, Singapore, Singapore. ACM, New York, NY, USA, 7 Pages. https://doi.org/10.1145/3497701.3497725

Yazdi, M.F., Kamel, S.R., Chabok, S.J.M. et al. Flight delay prediction based on deep learning and Levenberg-Marquart algorithm. J Big Data 7, 106 (2020). https://doi.org/10.1186/s40537-020-00380-z