

B.Tech. Project Report

on

Working with Big Data Tools

Submitted by
Anjali Priya
201751007

under the supervision of
Dr. Naveen Kumar
(On Campus)

(Signature of the Supervisor)



INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
VADODARA
2017- 2021

Declaration

I, Anjali Priya, declare that this written submission represents my ideas in my own words, and where other's ideas or words have been included, I have adequately cited and referred to the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated any idea/data/fact/source in my submission. I fully understand that any violation of the above will cause disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained.

Anjali Priya

Date: 26/04/2021

201751007

Abstract

In today's world, digitization has resulted in rapid data growth. Numerous social apps are being developed which result in increasing data massively every day and talking about social media platforms, millions of users connect on daily basis, information is shared whenever users use a social media platform or any other website, And these huge amounts of data is handled processed and stored with the help of Big Data. Nowadays, Established Enterprises capture all the data that streams into their businesses, apply analytics, and get significant results from it for their growth. This report represents my learning and Work in the domain of big data with its tool and combining it with the Machine-Learning concept to understand better the enterprise-level implementation of a real-life problem i.e. Flight Delay Prediction.

Acknowledgements

I would also like to thank IIITV BTP Committee for providing me, this wonderful opportunity. Also, especially I would like to thank Dr. Naveen Kumar for his constant guidance and support, for all the motivation, encouragement, and for always believing in me.

Contents

1	Introduction	1
2	Literature Survey	3
2.1	Technology Used	4
3	Workflow	5
4	Elaboration	6
4.1	Dataset	6
4.2	Analysis of the data	6
4.3	Loading data	8
4.4	Pipeline	8
4.5	Processing data	9
4.6	Observing Spark UI	9
5	Results and Discussion	13
5.1	Processing Time Comparison	13
5.2	Best Suited Algorithm	14
6	Conclusion and Future Work	16

Chapter 1

Introduction

In recent decades, increasingly large amounts of data are generated from a variety of sources. The size of generated data per day on the Internet has already exceeded two exabytes. These amounts of data are stored and analyzed by different tools which allow distributed storage, real-time processing, and data analysis. Data is considered as a key source for promoting the growth and well-being of society. Every day, more than 2 quintillion bytes of data are being created in this info-centric digitized world from various sources like scientific instruments, sensors, mobile phones, social networks, web authoring, the aviation industry, the telecommunication industry, social media, etc. Aviation industries are also dealing with handling so much data and processing it to make meaningful predictions.

As, in the Aviation industry, Flight delays create problems in scheduling, passenger inconvenience, and economic losses, there is growing interest in predicting flight delays beforehand in order to optimize operations and improve customer satisfaction. With the rapid growth of air traffic, increasing flight delays have become a serious and prominent problem. According to the Bureau of Transportation Statistics (BTS), nearly one in four airline flights arrived at their destination over 15 min late. It is reported that the annual total cost of air transportation delays was over \$30 billion, which poses a significant challenge to the development of Next Generation Air Transportation System[1]. Delay is one of the most remembered performance indicators of any transportation system.

The objective of this project is to perform an analysis of the historical flight data to gain valuable insights. with the help of machine learning models, we can build a predictive model to predict whether a flight will be delayed or not, given a set of flight characteristics. And, also we can make that model efficient by building it on a Big Data Platform, which handles more data to make accurate predictions.

This report will explain the work carried out by me in last four month. Starting from the collection of the data from the US Department of Transportation directory, then its pre-processed analysis, to its processing . Processing methods along with some machine learning algorithms has been mentioned with their recorded observations. The internal working of Spark UI has also been mentioned. Then, based on observations, comparisons have also been made and work has been properly concluded.

Chapter 2

Literature Survey

During this project span, I learned a lot many concepts. Especially, I got an opportunity to dive into the world of Big Data and understand it better. Also, I learned and applied my learning by building this application, in which the combination of Big Data and Machine Learning has been used. In this era, where "Data is the new oil", concepts like Big Data are bliss. Basically, It is a combination of structured and unstructured data collected by different organizations that can be mined for information and used in many machine learning projects, to make predictive models and other advanced analytics applications. Machine learning is a part of artificial intelligence that provides our systems with the ability to automatically learn and improve from experience without being explicitly programmed the system. Machine learning mainly focuses on the development of computer programs that can access data and use that data to learn for themselves. And, one of the most popular Enterprise-level Data processing frameworks is Spark. It provides us a detailed, unified framework to manage big data processing requirements with a variety of datasets that are diverse in nature as well as the source of data like batch and real-time streaming data. Spark enables applications in Hadoop clusters to run much faster in memory. It can perform tasks much faster than MapReduce, during multi-stage jobs. In addition to MapReduce operations, it also supports SQL queries, streaming data, machine learning.

These capabilities can be used by developers, in form of stand-alone or combine them to run in a single data pipeline use case. Spark's DAG(Directed Acyclic Graph) can be distributed more efficiently. It manages data through RDDs(Resilient Distributed Datasets) using partitions that help parallelize distributed data processing with negligible network traffic for sending data between executors. Spark is very developer's friendly.

After taking the inspiration and learning the concepts that are mentioned in the above paragraph I have tried summarize the initiatives used to address the flight delay prediction problem, according to scope, data, and computational methods, I have built an application i.e. flight delay predictor in which I have used the concept of big data and machine learning using scala and spark. Here I have used the Apache Spark framework for processing the large dataset and distribute the task across multiple computers. In this project, I have used two machine learning models to train the application first one is Linear Regression and the second one is Random Forest.

2.1 Technology Used

- **Spark[2]**:- Apache Spark is a data processing framework, can quickly perform processing tasks on very large data sets, and can also distribute data processing tasks across multiple computers, either on its own or in tandem with other distributed computing tools.
- **Scala[3]**:- Scala is a compiler based and a multi-paradigm programming language
- **Python**:- Python is an interpreted, high-level and general-purpose programming language.
- **IntelliJ**:-IntelliJ IDEA is an integrated development environment (IDE)

Chapter 3

Workflow

There is an overview of the process of data mining and data modeling, from collecting the data, through the data preparation and finally the data modeling.

Collection

Through various data sources, relevant data is collected.

Ingestion

Data is ingested into the spark environment so that it stores them in a cluster and does fast computations.

Preparation

Data is cleansed, shaped, transformed and estimates are made to proceed.

Computation

Here machine learning algorithms and techniques run to bring useful results.

Presentation

Result is presented and visualized.

Chapter 4

Elaboration

4.1 Dataset

The data[4] used was published by the US Department of Transportation and can be found here :- <http://stat-computing.org/dataexpo/2009/the-data.html>.

It gives detailed information on every flight, which incorporates their booking and take-off circumstances and real takeoff, origin, destination, date, and carrier number, arrival time, departure time, delays, taxiout time and more. It comprises almost 23 years' worth of data, and analyzing such amount of data requires big data tools such as Spark.

4.2 Analysis of the data

Firstly, data is cleaned by removing any null values, missing value columns, and by filling dummy variables.

Then, Since most variables are numericals so firstly I created a correlation matrix, of the numerical variables, to understand if I can find any correlation between the attributes.

Figure below representing observed Correlation matrix between attributes.

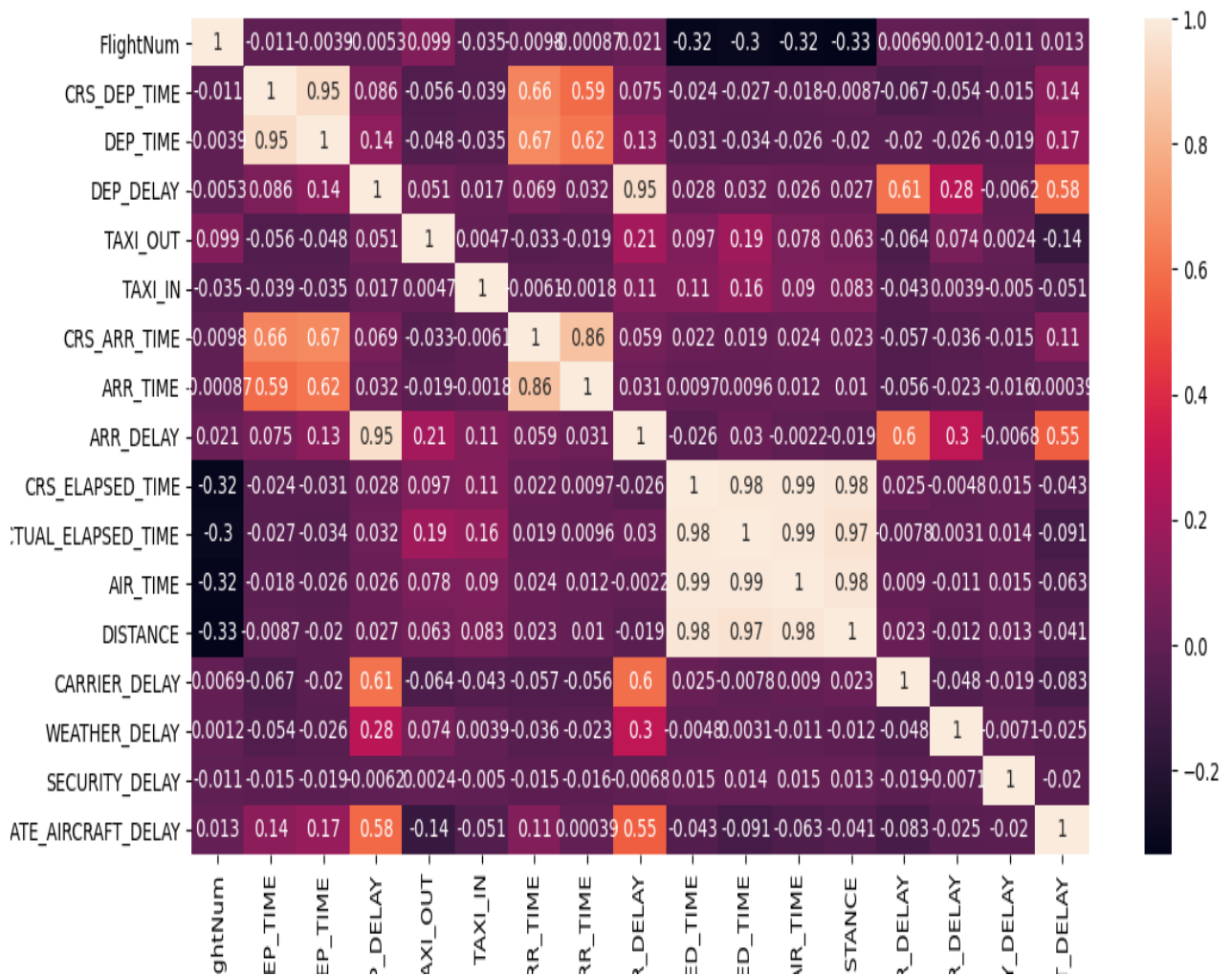


fig 4.2.1- Correlation Matrix

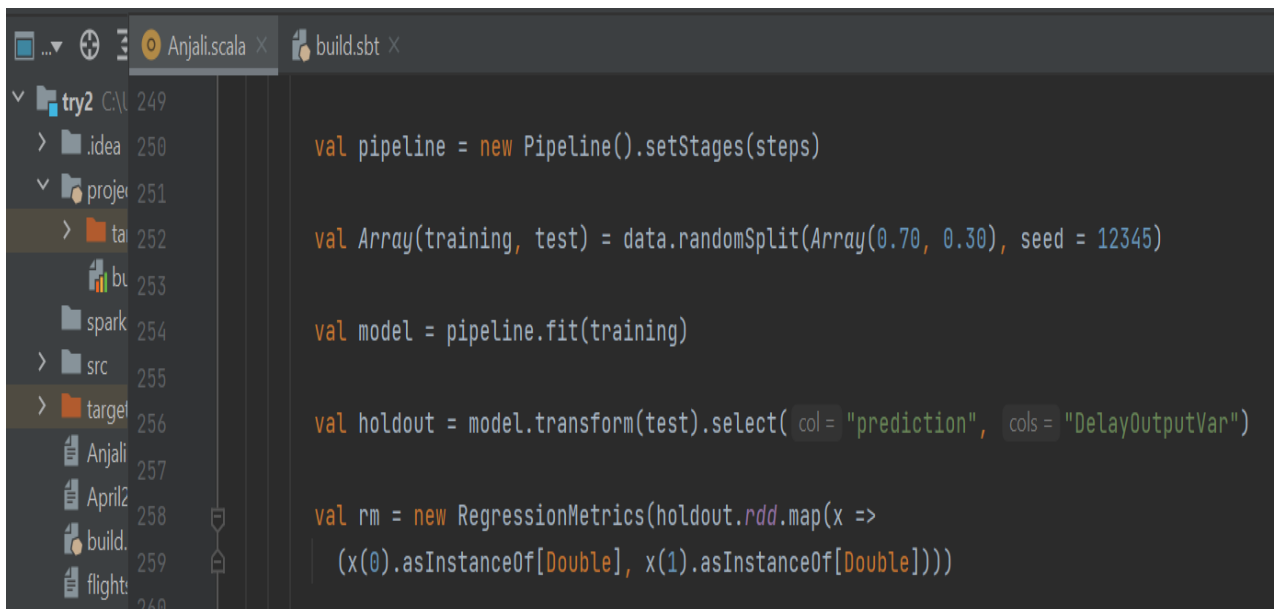
From Figure 4.2.1, I observed that the ARR_DELAY(arrival delay) is highly correlated to DEP_DELAY (departure delay), so the departure delay might be the most indicative attribute of a flight delay. Other weak correlations have also been found. I have considered these correlations to make the prediction.

4.3 Loading data

Now, I read the data from the CSV file using the spark SQLContext. Then loaded the data into a DataFrame which is a distributed collection of data organized into named columns. The relevant columns observed in the Correlation matrix previously So, while running the spark application in a cluster this will distribute the data randomly across the cluster.

4.4 Pipeline

I created a pipeline to process all the transformations on the data, shown in Figure 4.4.1 . It's a sequence of stages. This allows us to define all the steps and run them sequentially. The pipeline comprises the steps- converting the categorical values to numerical ones, then assembling all in a features vector, and running the machine learning algorithm to obtain a model.



```
val pipeline = new Pipeline().setStages(steps)

val Array(training, test) = data.randomSplit(Array(0.70, 0.30), seed = 12345)

val model = pipeline.fit(training)

val holdout = model.transform(test).select(col = "prediction", cols = "DelayOutputVar")

val rm = new RegressionMetrics(holdout.rdd.map(x =>
  (x(0).asInstanceOf[Double], x(1).asInstanceOf[Double])))
```

fig 4.4.1- Pipeline code

4.5 Processing data

The data processing is dependent on the machine learning algorithm implementation. To use the categorical variables for further processing was needed. Variables were converted to numerical values by applying a `StringIndexer` and `OneHotEncoder` methods.

For finding out the best model, I ran different relevant Machine Learning Algorithms(mentioned below) and calculated loss function with each.

Linear Regression:- Since the observation depicts a linear correlation of departure delay attribute with arrival delay attribute, applying linear regression seemed a suitable approach. It basically performs a regression task. Regression models a target prediction based on independent variables.

Random Forest:- Since, this algorithm can add randomness and is diverse, so been used to cover the possibilities. In this method, basically only a random subset of the features is taken into consideration by the algorithm for splitting a node.

4.6 Observing Spark UI

During computation, on the active port(4040), the internal working details has also been noted and displayed below.

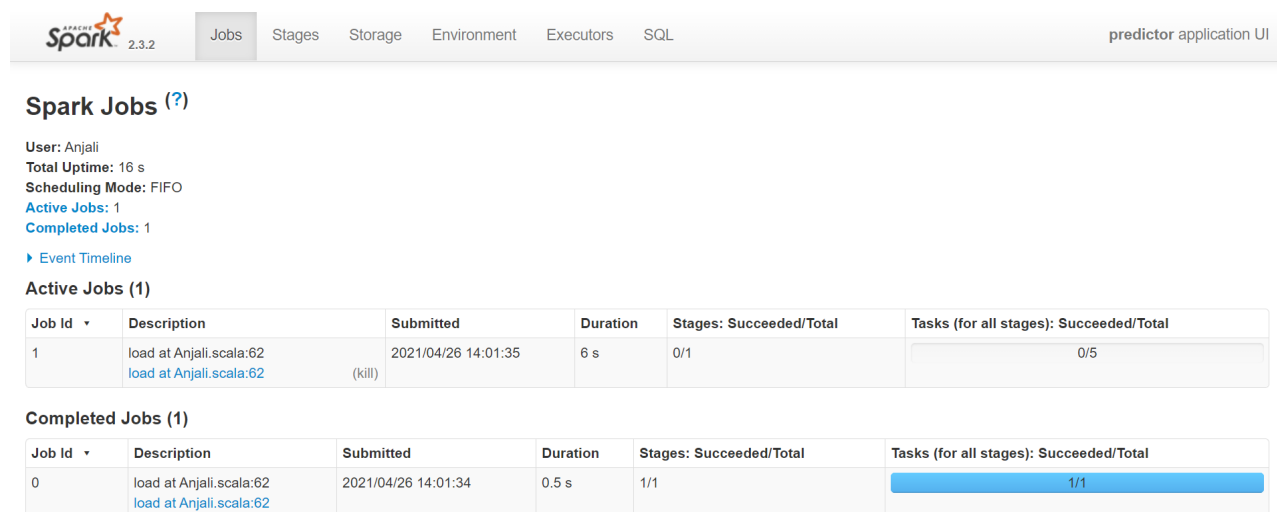


fig 4.6.1- Event Timeline

Figure 4.6.1 describes the active jobs at the moment on port- 4040, when spark application was triggered.

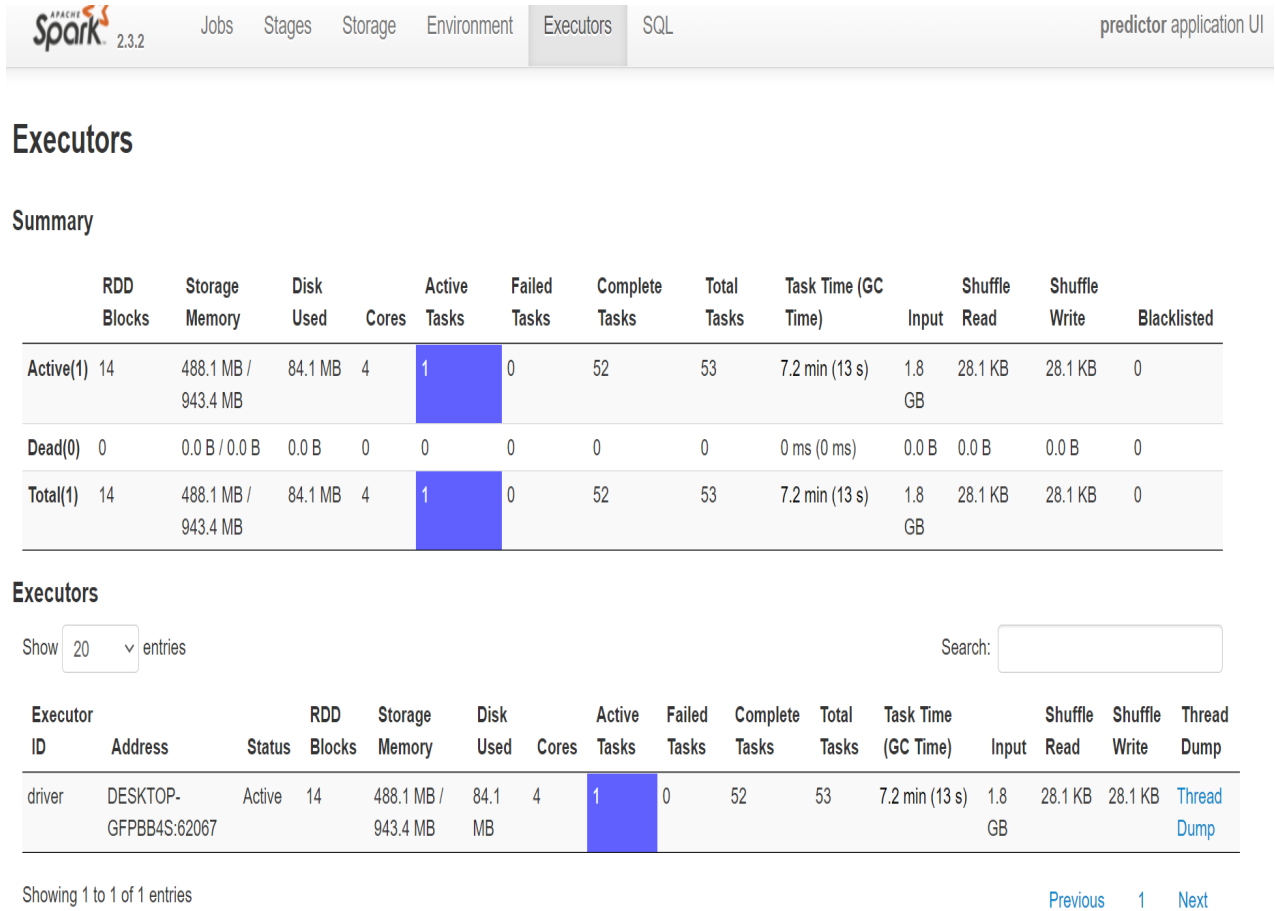


fig 4.6.2- Tasks execution

Figure 4.6.2 describes the storage strain on the disk and cores covered during execution of the tasks.

Stages for All Jobs

Active Stages: 1

Completed Stages: 73

Active Stages (1)

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
73	treeAggregate at WeightedLeastSquares.scala:100 +details (kill)	2021/04/26 14:05:41	Unknown	0/5				

Completed Stages (73)

Stage Id ▾	Description	Submitted	Duration	Tasks: Succeeded/Total	Input	Output	Shuffle Read	Shuffle Write
72	first at LinearRegression.scala:319 +details	2021/04/26 14:05:40	8 ms	1/1	16.0 MB			
71	countByValue at StringIndexer.scala:140 +details	2021/04/26 14:05:40	32 ms	5/5			13.6 KB	
70	countByValue at StringIndexer.scala:140 +details	2021/04/26 14:05:40	0.8 s	5/5	67.2 MB			13.6 KB
69	countByValue at StringIndexer.scala:140 +details	2021/04/26 14:05:39	24 ms	5/5			13.6 KB	
68	countByValue at StringIndexer.scala:140 +details	2021/04/26 14:05:39	0.5 s	5/5	67.2 MB			13.6 KB
67	treeAggregate at RegressionMetrics.scala:57 +details	2021/04/26 14:05:37	2 s	5/5	29.1 MB			
66	count at LinearRegression.scala:921 +details	2021/04/26 14:05:37	8 ms	1/1			295.0 B	
65	count at LinearRegression.scala:921 +details	2021/04/26 14:05:36	0.5 s	5/5	67.2 MB			295.0 B

fig 4.6.3- Stages of each Jobs

Figure 4.6.3 describes the details and sequence wise stages of each job being carried out.

Storage

RDDs

ID	RDD Name	Storage Level	Cached Partitions	Fraction Cached	Size in Memory	Size on Disk
11	*(1) Project [YEAR#10, MONTH#11, DAY_OF_MONTH#12, DAY_OF_WEEK#13, UniqueCarrier#14, TAIL_NUM#15, FlightNum#16, ORIGIN#17, DEST#18, CRS_DEP_TIME#19, DEP_TIME#20, DEP_DELAY#21, TAXI_OUT#22, TAXI_IN#23, CRS_ARR_TIME#24, ARR_TIME#25, ARR_DELAY#26, CANCELLED#27, CANCELLATION_CODE#28, DIVERTED#29, CRS_ELAPSED_TIME#30, ACTUAL_ELAPSED_TIME#31, AIR_TIME#32, FLIGHTS#33, ... 9 more fields] +- *(1) FileScan csv [YEAR#10,MONTH#11,DAY_OF_MONTH#12,DAY_OF_WEEK#13,UniqueCarrier#14,TAIL_NUM#15,FlightNum#16,ORIGIN#17,DEST#18,CRS_DEP_TIME#19,DEP_TIME#20,DEP_DELAY#21,TAXI_OUT#22,TAXI_IN#23,CRS_ARR_TIME#24,ARR_TIME#25,ARR_DELAY#26,CANCELLED#27,CANCELLATION_CODE#28,DIVERTED#29,CRS_ELAPSED_TIME#30,ACTUAL_ELAPSED_TIME#31,AIR_TIME#32,FLIGHTS#33,... 6 more fields] Batched: false, Format: CSV, Location: InMemoryFileIndex[file:/C:/Users/Anjali/Downloads/try2/Anjali.csv, file:/C:/Users/Anjali/Download..., PartitionFilters: [], PushedFilters: [], ReadSchema: struct<YEAR:string,MONTH:string,DAY_OF_MONTH:string,DAY_OF_WEEK:string,UniqueCa...	Disk Serialized 1x Replicated	5	100%	374.1 MB	80.2 MB
27	*(1) Sample 0.0, 0.7, false, -1772833110 +- *(1) Sort [ORIGIN#17 ASC NULLS FIRST, DEST#18 ASC NULLS FIRST, ARR_DELAY#26 ASC NULLS FIRST, CANCELLED#27 ASC NULLS FIRST, FLIGHTS#33 ASC NULLS FIRST, DelayOutputVar#70 ASC NULLS FIRST,	Memory Deserialized 1x Replicated	5	100%	67.2 MB	0.0 B

fig 4.6.4- Memory Statement

Figure 4.6.4 describes the RDD’s details.

Chapter 5

Results and Discussion

After building this model, I have observed some patterns, which are described below.

5.1 Processing Time Comparison

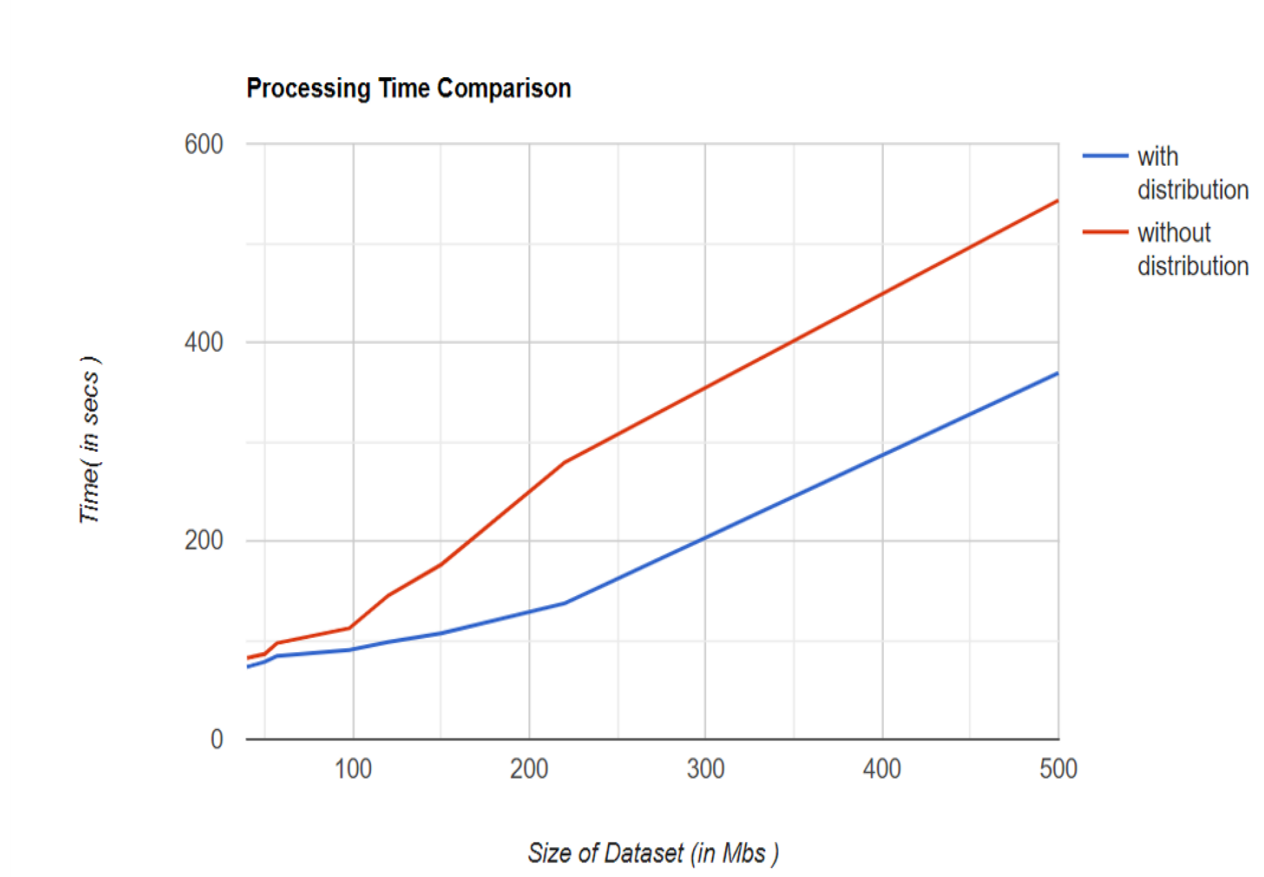


fig 5.1.1:- Processing Time Comparison

Figure 5.1.1 depicts a comparison of time taken by the application when the spark framework was used and time is taken by the application when the spark framework wasn't used. This Graph has been plotted after testing both systems by providing combinations of different sized datasets and the time is taken to produce results has been noted by both systems in each scenario.

As observed, the time taken to compute large datasets in the Spark framework takes a lot less time than one which didn't use the Spark framework. Today, when TBs of data is computed, in this scenario, handling this much data and computing at high speed is the requirement and Spark seems to be a wonderful solution for the same.

5.2 Best Suited Algorithm

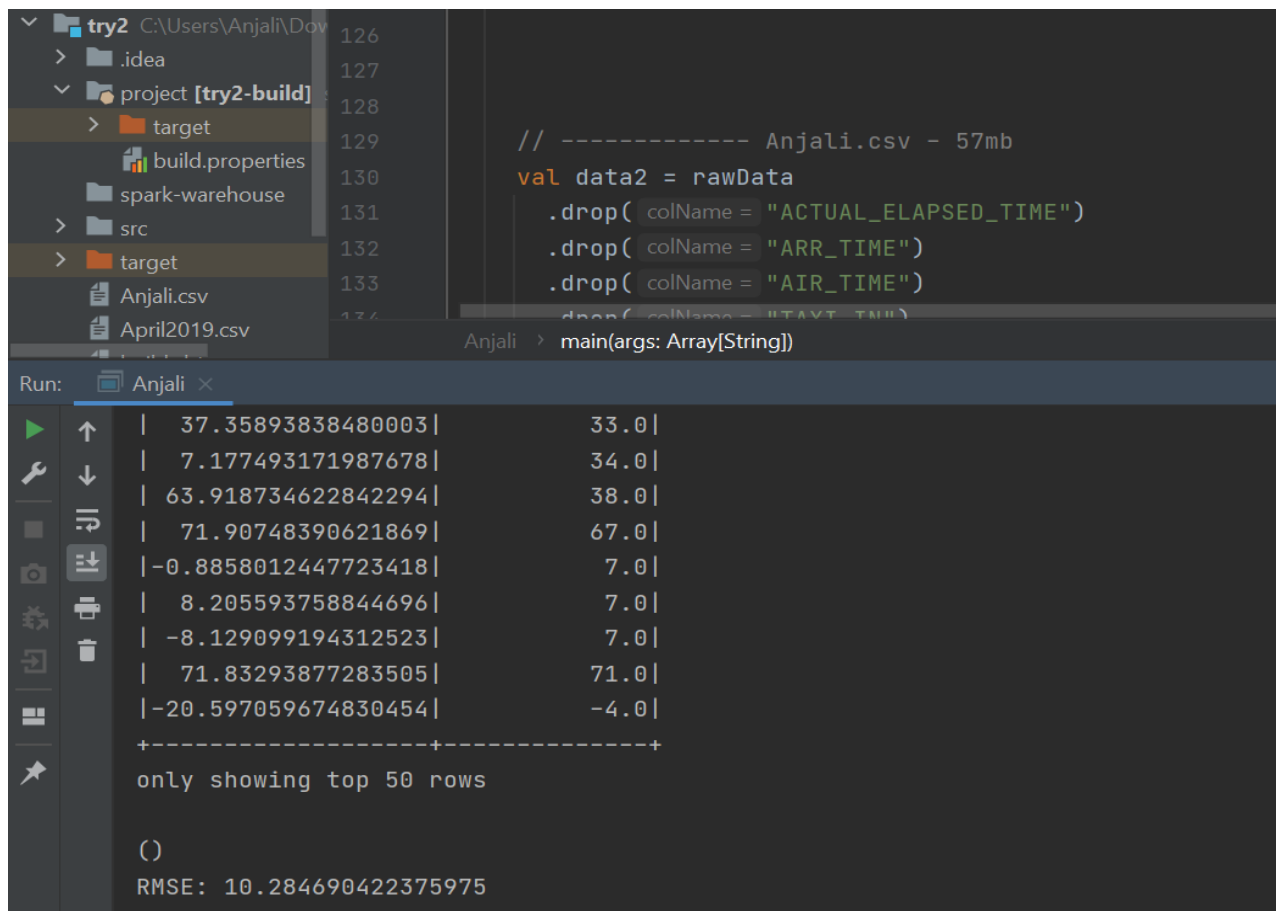


fig 5.2.1:- RMSE with Linear Regression

```
try2 C:\Users\Anjali\I... 252    val Array(training, test) = data.randomSpli
> .idea 253
> project [try2-build 254
> target
build.properties
Anjali > main(args: Array[String])

Run: Anjali x
21/04/26 20:44:23 INFO BlockManager: Using org.apache.spark.storage.Rand
21/04/26 20:44:23 INFO BlockManagerMaster: Registering BlockManager Blo
21/04/26 20:44:23 INFO BlockManagerMasterEndpoint: Registering block man
21/04/26 20:44:23 INFO BlockManagerMaster: Registered BlockManager Blo
21/04/26 20:44:23 INFO BlockManager: Initialized BlockManager: BlockMan
checkpoint-2

size - 4
checkpoint-3 , followed by printing rawData - double casted 3 columns

[YEAR: int, MONTH: int ... 30 more fields]
checkpoint-4, printing data after dropping

[ORIGIN: string, DEST: string ... 5 more fields]
RMSE: 30.9963568449157

Total Time taken by the program (secs) -161
```

fig 5.2.2:- RMSE with Random Forest

RMSE is the square root of the mean square error(MSE). And used as a loss function here in the model. The lesser the numerical value of RMSE, the better the model. From Figure 5.2.1 and Figure 5.2.2, it can be observed that Linear Regression's RMSE is less than Random Forest's RMSE.

As observed, Linear Regression performed better than Random Forest, so it's best suited for this model as most relations between attributes were linear.

Chapter 6

Conclusion and Future Work

In this project, we were able to successfully apply machine learning algorithms to build an efficient model which predicts flight delay within the Spark framework. As flight delay costs a lot to the airlines as well as passengers in financial and environmental terms and as the outcome is directly associated with the passenger and the airlines, it is very crucial to get real-time delay for each one within the air transport system. This project can contribute a lot to scale back monetary loss and to the higher and smooth operation. Also, it was observed by me that the delay of the flights was heavily dependent on the departure airports. This clearly implies that if an airport is busy and is a major airport, the chances of flight delays will be more as compared to the flights by less busy airports. I believe that flights are the fastest way to reach a place, therefore, its efficiency matters a lot. With the help of this model, we can make the overall system more efficient.

In future work, we will continue to explore more in this world of Big Data and will experiment with how different models and tools can be associated to achieve the highest performance in time, correctness, and resource optimization.

References

- [1] Chen J, Li M. Chained predictions of flight delay using machine learning. In: AIAA Scitech 2019 Forum. 2019. p. 1661. <https://www.researchgate.net/publication/330185077>
- [2] Apache Spark Documentation, <https://spark.apache.org/docs/latest/>
- [3] Scala Documentation, <https://docs.scala-lang.org/>
- [4] Dataset, <http://stat-computing.org/dataexpo/2009/the-data.html>