**Real-Time Flight Delay and Distance Prediction**

A PROJECT REPORT

***Submitted by***

|  |  |
| --- | --- |
| BL.EN.U4AIE20019 | Hema Srivarshini Chilakala |
| BL.EN.U4AIE20037 | N.Preeti |
| BL.EN.U4AIE20061 | Sreekar Praneeth Marri |

**21AIE304 – Big Data and Database Management**

**B.Tech. in Computer Science and Engineering (Artificial Intelligence)**



AMRITA SCHOOL OF ENGINEERING, BANGALORE

 AMRITA VISHWA VIDYAPEETHAM

BANGALORE 560 035

JANUARY – 2023

**ABSTRACT**

Flight delays are steadily getting worse, which causes airline businesses to have greater trouble making money and to lose customers. Supervised machine learning models were used to anticipate flight delays to address this issue.

The application of collaborative decision-making is encouraged by accurate flight departure time forecast, which permits the prudent use of airport support services, aprons, and runway resources.

Predicting flight arrival and departure is a classification issue. Big data Analysis is essential for examining such patterns, trends, and correlations in vast quantities of unprocessed data in order to make data-driven choices. The analytical pipeline ensures that there is no data loss when the data is transferred into all segments and later shown on the virtual dashboard.

Predicting whether an aircraft will arrive or depart on schedule is the key goal. The data from the flight APIs is read for this project using Kafka. The flight data is analyzed using machine learning methods like linear regression before being fed into a NoSQL database using Spark Streaming, which receives the data through Kafka (MongoDB). The user may now visualize the data stored in MongoDB using a dashboard and make queries to obtain information.

**INTRODUCTION**

Because of the financial losses that the aviation industry suffers as a result, flight delays have become a major issue for air travel across the world.

Both the airlines and the customers are inconvenienced by these delays. The effect is a lengthening of the journey, which raises the cost of housing and meals and, eventually, adds stress to the travelers’ lives. The extra costs connected with their staff, aircraft repositioning, fuel waste while attempting to shorten flight durations, and many other factors harm the reputation of the airlines and frequently lead to a decline in consumer demand.

There are essentially two types of flight delay predictions. In Mode 1, the model considers static information on the routes, airports, and aircraft qualities during a given time. In Mode 2, the time-series data for the airport, airline, arrivals, and departures are dynamically captured by the model.

This study assesses the second approach, which takes timer series data into account to forecast flight delays.

the pipeline process goes like, the data is first supplied to Kafka, which is connected to spark streaming, where the implementation of streaming the data and machine learning occurs. Once the needed output is created, it is pushed into MongoDB.

Streamlit is a front-end based programming provided by python, the database pushed into the MongoDB is visualized on the front end using streamlit.

**DATASET**

The Bureau of Transportation Statics provided the real-time dataset.

We are given the opportunity to select the feature column for the analysis, and as a result, we receive the dataset based on the features we selected.

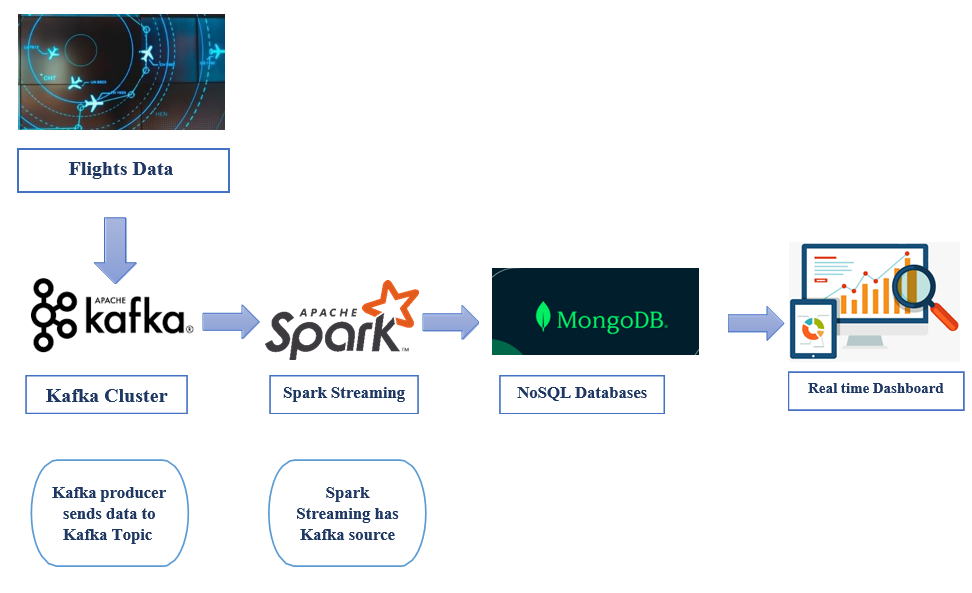
There are nine features, which are:

* id: ID composed of carrier, date, origin, destination, flight number
* dofW: day of week (1=Monday, 7=Sunday)
* carrier:delay caused by aircraft cleaning, aircraft damage, awaiting the arrival of connecting passengers
* dest: Destination Airport
* crsdephour: scheduled departure hour
* crsdeptime: scheduled departure time
* depdelay: departure delay in minutes
* crsarrtime: scheduled arrival time
* arrdelay: arrival delay in minutes
* crselapsedtime: elapsed time
* dist: distance between source and destination

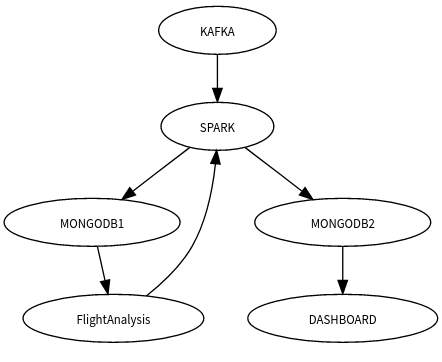
**ARCHITECTURE**

Architectures considered:

1. LOGISTIC REGRESSION

****

1. LINEAR REGRESSION

******

**IMPLEMENTATION**

**DATASET:**

The dataset consists of 9 main features which play an important role in classifying and predicting the outcomes. The information is gathered from Bureau of Transportation and Statistics

**KAFKA:**

Zookeeper, which is used by Kafka brokers to identify which broker is the leader of a given partition and topic and carry out leader elections, should be started after installing Kafka on the system.

Start the Kafka server after that, then create the topic you want to work on by setting the partitions and replication factor. Launch the Kafka producer now, add the data, and keep it running.

Version: Kafka-0-10\_2.12:3.2.1

The above version of Kafka package is imported in Spark

**SPARK STREAMING:**

Spark Version: 3.2.1

Applications are written using Scala.

Scala Version: 2.12

**LOGISTIC REGRESSION**

The next stage is to build the machine learning method and access the producer code in the sark streaming. Initially, launch the spark shell.

Now import all the necessary packages, including sparksession, features, categorization, functions, struct type, trigger, and others.

Construct the schema for our data once the importing is finished, then create a spark session on the local host and give the appName the value "mlmodel."

Once this is complete, we read the dataset and combine it with the earlier-created schema. We now define the bucketizer by separating the parameter and assigning depdelay as the input column and delayed as the output column. The data is then

transformed using the bucketizer function. The transformed data is then mapped in a related manner. We now stratify the data by taking into account a few feature columns and joining them with indexes.

Once this is complete, the encoders will use the index data as input, and the output column will be the column name concatenated with Enc.

The vector assembler, which creates a single vector column from a specified list of columns, accepts an array containing all the feature columns as input. This output and input should be passed to the logistic regression algorithm. Indexed data, encoded data, labelled data, assembled data, and Dtree data should all be combined into a single variable and sent to the pipeline. On a variable that contains the output from the pipeline, we run the Cross Validator and MultiClassificationEvaluator. Predictions are made once the final data is fitted into the main model. The last stage is to execute the write stream query, which pulls data from the producer and incorporates the machine learning component, once the prediction model has been established. Even the mongoDB connection has been formed by these query instructions.

Application written in Scala language in Spark.

**LINEAR REGRESSION**

The second prediction performed is based the distance feature. Two columns i.e., features and label are necessary hence the data needs to be prepared accordingly. When fitting Linear Regression Model without intercept on dataset with constant nonzero column by “l-bfgs” solver, Spark MLlib outputs zero coefficients for constant nonzero column. This approach is done without using pipeline.

A producer file is created to remove unwanted symbols like “” and [] and is run. In spark platform required statements are imported and it is connect to mongoDB database. The data is read using readstream and is inserted into the collection created in MongoDB. This data is now read retrieved using writestream and the streaming is started. Once the data is retrieved the required features are selected for the model , then schema is made , vector assemble is used to keep all the features in a single

vector or an array. Linear Regression function is called on the output and the prediction are made.

Application written in Python language in Spark i.e., Pyspark.

**MONGODB :**

Shift to the database into which the classified and predicted data is added to, then print the collection in which you have save the data the classified values are displayed in MongoDb database. When fitting LinearRegressionModel without intercept on dataset with constant nonzero column by “l-bfgs” solver, Spark MLlib outputs zero coefficients for constant nonzero column.

A connecter to connect Spark and MongoDb is used for Spark Scala application.

Connector Version: mongo-spark-connector\_2.12:10.1.0

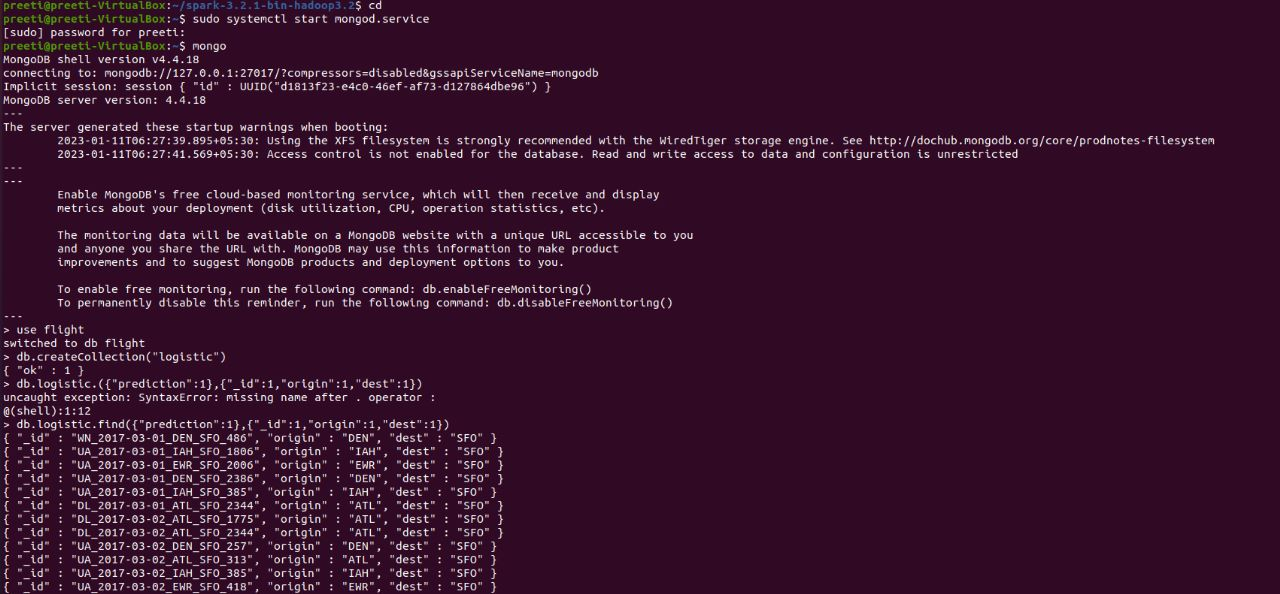
**DASHBOARD:**

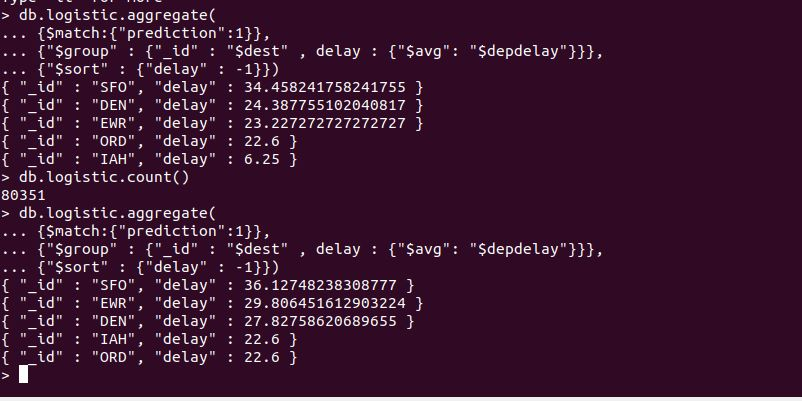
The interface which is used to develop the front end is called streamlit. Required import statements are imported and MongoDB connection is made. The front end is styled according to the developer’s choice, how the developer wants the frontend to be viewed by the users.

The output databases stored in MongoDB are connect by navigating to their respective databases and retrieving them by using collection.find

**RESULTS**

**LOGISTIC REGRESSION :**

**Fig(1): The data is sent to mongodb where spark and mongodb connecter is used to link it. The package is used is --packages org.mongodb.spark:mongo-spark-connector\_2.12:10.1.0**

****

**Fig(2): Queries Performed on the final data**

**LINEAR REGRESSION :**

**A picture containing text

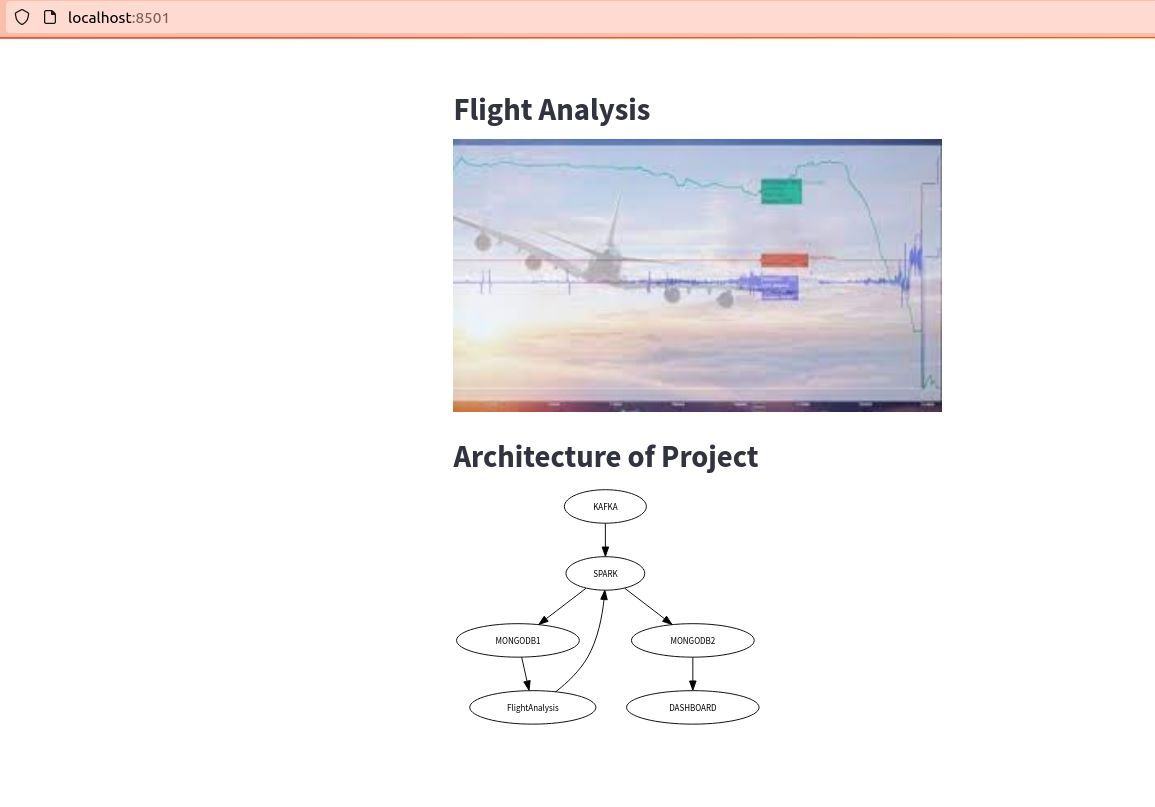
Description automatically generated**

**Fig(3): Printing the Collections in MongoDB before performing ML**

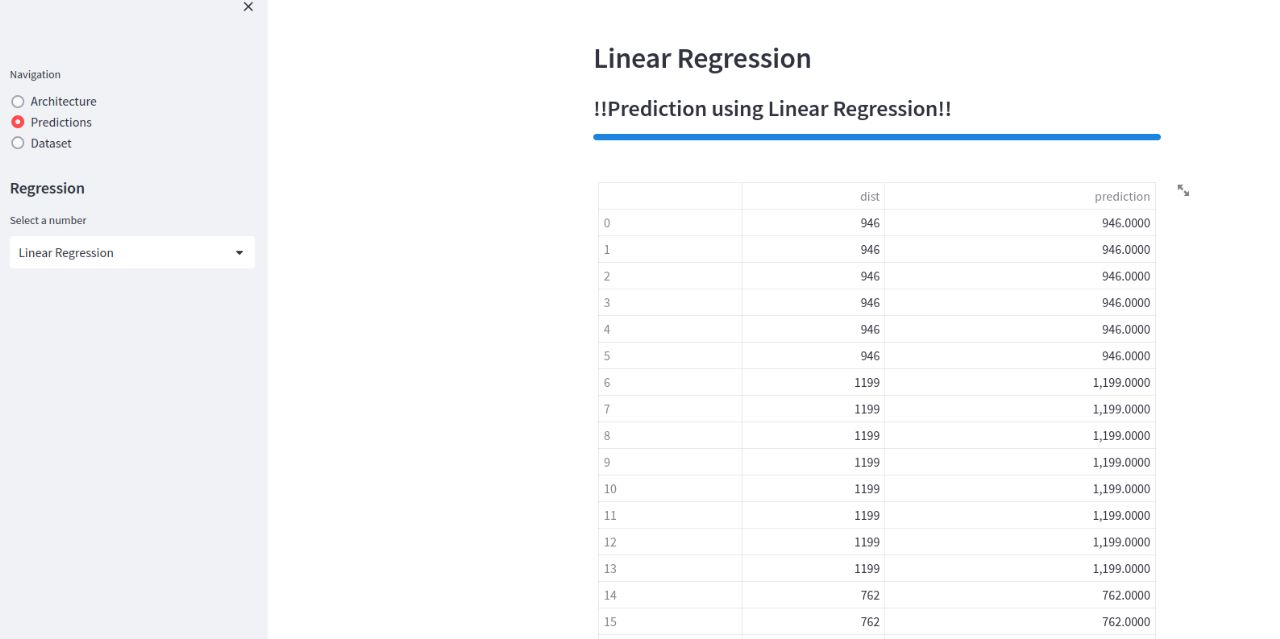
**Graphical user interface, text

Description automatically generated**

**Fig(4): Predicted values displaying in MongoDB**



**Fig(5): Dashboard for Flight Analysis and Architecture of the project is displayed when the respective option is selected in the navigation sidebar**



**Fig(6): Distance Prediction for the flights is displayed when the respective option is selected in the navigation sidebar**

Graphical user interface, application, table

Description automatically generated

**Fig(7): Logistic Regression is performed to classify the delay**

Graphical user interface, application

Description automatically generated

**Fig(8): Dataset used for the project is displayed when the respective option is selected in the navigation sidebar**

**CONCLUSION**

The main aim of the paper is to implement two different supervised machine learning algorithms, Logistic Regression and Linear Regression

The arrival of the flight will be delayed or not is performed by Logistic Regression as the output is of binary form. The distance between source and destination is predicted using Linear Regression. As we know handling such huge data is not an easy task, so to overcome this obstacle Big DataAnalysis pipeline was implemented , the data for the regression models are sent into the kafka producer, and then they are streamed as records via spark streaming platform , respective machine learnign algorithms are performeD. The outputs are stored in MongoDB and are further vizualised using streamlit in the front end