Airplane delay seasonality and prediction

Harin Avvari Sashanka Mouli Mulugu

# Abstract—It can be very tiring and annoying to deal with airport delays. We want to make a model that can show seasonality in delays and hopefully predict what the cause of the delay was. We believe this is a significant issue because addressing these potential causes with the knowledge of seasonality(when these causes happen more often), will allow airports to address these issues before they even happen. Our project could show that it is possible to link seasonality with causes of delays. This again would help airlines address specific issues at specific times of the year.

1. Introduction

It can be very tiring and annoying to deal with airport delays. We want to make a model that can show seasonality in delays and hopefully predict what the cause of the delay was. We believe this is a significant issue because addressing these potential causes with the knowledge of seasonality(when these causes happen more often), will allow airports to address these issues before they even happen. Our project could show that it is possible to link seasonality with causes of delays. This again would help airlines address specific issues at specific times of the year. An extension of this project would be to compare our findings data from multiple years for validation purposes before we can firmly say that this indeed is something that we can correlate.

1. DATA

Data source: [https://www.kaggle.com/datasets/sherrytp/](#_bookmark0) airline-delay-analysis?resource=download

EDA:

1

Analysis: The peak at KDE curve indicates that there are significant flights with no delay and Flights are clustured at point 3.

The first histogram indicated that there are flights with total time delay above 0.5.

From the Second histogram we can say that there are flights with mimal time delay just above 0.

2.

Analysis: We are using Countplot to show how average delays compare among different carriers.

The carrier WN has a peak around 1.4 indicating higher average delay ratio. They are more delayed

The carrier HA has a lowest peak around 0.1 indicating lower average delay ratio when compared to all others.

3.

Analysis:

We could observe from the graph that there is a significant peak in avg total delay in the month of June (6). This indicate their are longer delays in the month of June.

We could also observe that the lowest points are a 9 and 11. Indicating that the avg total delay us very less in these months. They are on time.

4.

Analysis: From this graph we could could observe that Wednesday has the highest average total delay. The second heighest is at sunday and the least is the friday.

5.

Analysis: With this graph we could get which flights has the most delays. Which flight number. This is just a visualization of the spread of count of delays. Higher bars indicates they are more frequently delayed and lower bar indicates they are less delayed.

6.

Analysis: We could observe that the carriers F9 and HA have the highest average speeds, while OH has the lowest.There is a Zigzag pattern indicating the variability of average speeds between different carriers.

(1)

* 7.
* Analysis: We could observe that there is a peak at 2 AM and sudden drop at 5 AM which is the lowest and slightly increment afterwords.
* There might be some operational issues leads to more delays.
* 8.
* Analysis: This box plot shows most of the carriers have similar level of carrier delay around 1500 min .
* AA has the heighest delay among other flights. We could observe that there are few outliers far from the cluster. One prominent one is with OO which to more than 2500 min carrier delay being a big outlier. We could also observe that f9 has the lowest delays and they are most efficient from the graph.
* 9.
* Analysis: This bar plot represents the weather delays based on destination. We could see there are some high bars indicating there is bad weather in that particular area because of this there is highest weather delay. The lower bars indicates there is a lowest average wether delays indicating average weather condition. The delay might be caused by some external factor When reaching the destination.
* 10.
* Analysis: This graph is similar to above graph it represents how weather delays but this time it is by origin of flight. Same like the above graph the long bar indicates higest weather delay and short bar indicated less weather delay at the starting point of flight. We could also see how they are distributed across the different origins.

Algorithms used

1. Linear Regression

Use Case: Predicting the actual total delay time based on continuous features is very useful when we are doing with linear regression

It is easy to interpret the impact of each feature.

With Linear Regression it is easy to see how linear the relationship is between features and delay time.

It is computationally efficient, and it is a good fit for real-time predictions in smaller-scale operations.

But it may not be as appropriate because it assumes a linear relationship, true complexity of flight delays may be ignored.

1. Decision Tree Regressor

Use Case: Predicting non-linear relationships between various factors can be done easily with Decision Tree Regressor.

It is very useful when delays are caused by complex interactions between factors like traffic, etc at airports.

It is easy to understand how decisions are made at each point iif weather is bad they cause more delay.

It can handle missing data well without imputation.

Due to overfitting of noisy data the tree becomes too deep.

1. Random Forest Regressor

When interactions between features are complex predicting delays using Random Forest Regressor is very useful

It is more robust than a single decision tree

It reduces overfitting and improves generalization to unseen data.

It provides valuable insights into the most important factors which are causing the delays

It can handle large datasets effectively and also non- linear relationships between features and delay time It takes more resources when compared to linear regression.

1. K-Nearest Neighbors Regressor (KNN)

It works well when we are predicting delay times based on features like historical data from similar flights

It is a non-parametric model, it won’t consider the relationship between the features and the delay time. Like which are linear or nonlinear works for both.

It works well when we are predicting total delays based on flights that are close in time or flights which are from the same airport.

KNN can be computationally expensive on large datasets,then others because we need to find the nearest neighbor for all predictions.

It is also not very strong with higher dimension features.

1. Support Vector Regressor (SVR)

It works well when predicting total delay when we want to handle outliers

It works well on non-linear relationships between features and total delay.

It can be tuned efficiently to ignore some level of error so that it is efficient to outliers in flight data like very rare delays on few features.

With the help of kernels we can fit it for non-linear relationships between features and delay time.

But it can be slow to fit and train data and it is very sensitive to hyperparameters. It needs to tuned efficiently.

1. Gradient Boosting Regressor (GBR)

When delays are influenced by some specific patterns in the flight delay data this model will give accurate prediction by combining less correlated features.

The Gradient Boosting gives high accuracy. It will iteratively improve based on errors.

It works well with non-linear and non-additive relationships between features and total delay time. With GBR we can know which features are most important in predicting delays.

It is computationally expensive. It takes so much time in training data when there is a large dataset.

SVR turned out to have the best performance.

* 1. SPARK(Data Cleaning/Processing)
  2. Distributed Data Cleaning/Processing

To ensure the dataset was prepared for distributed analysis and machine learning modeling, the following cleaning steps were performed using PySpark:

# Omit Unnecessary Columns:

Column \_c20 was eliminated because it was not relevant to the task at hand.

This decreased dataset intricacy and enhanced processing effectiveness.

# Dealing with Missing Data:

Entries containing NULL values in key columns like OP\_CARRIER\_FL\_NUM and features related to delays were eliminated.

Guaranteed dataset integrity and dependability for machine learning.

# Change the date format:

The FL\_DATE column was converted to a timestamp structure.

Time-based operations have been activated for analyzing seasonality and trends.

# Type Casting for Uniformity:

Important variables such as DEP\_DELAY, ARR\_DELAY, and TOTAL\_DELAY were converted to double data type.

Avoided numerical computation errors caused by mismatched data types.

Scaling features means transforming variables to a specific range.

Utilized VectorAssembler and StandardScaler to normalize numerical attributes such as DEP\_DELAY and ARR\_DELAY.

Normalized scales ensured equal contribution of features to the model.

# Identification and elimination of outliers:

Exceptional values in columns related to delay were detected and eliminated.

Avoided anomalies from distorting model training and statistical measurements.

# Estimation of Absent Data:

NULL values in certain columns were substituted with the average or middle value.

Kept rows that had little missing data in order to preserve the size of the dataset.

Duplicate rows were detected and deleted using key attributes.

# Removing redundant rows:

Invalid or unrealistic values in rows (such as negative delays) were removed through filtering.

Concentrated the examination on pertinent, significant documents.

Changing the format or title of columns:

Column titles were made uniform to ensure clear and consistent naming.

Enhanced data set usability and compatibility with subsequent tools.

1. SPARK(Algorithms/Visualizations)
2. Algorithms/Visualizations
   1. Linear Regression Why Useful:

Linear regression is excellent for modeling relationships between continuous features and a target variable when the relationship is linear.

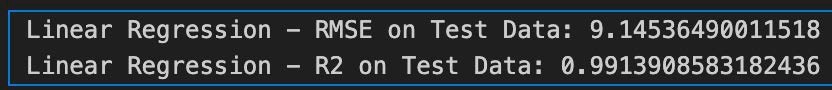
For flight delays, it can predict TOTAL\_DELAY based on numerical features like DEP\_DELAY, ARR\_DELAY, or TAXI\_OUT.

Strengths:

Computationally efficienta nd works wellon large datasets. Easy to interpret.

Limitations:

Assumes a linear relationship, which may not capture complex interactions in flight delays.



* 1. Decision Tree Regressor Why Useful:

Decision trees handle non-linear relationships and can model complex interactions between features.

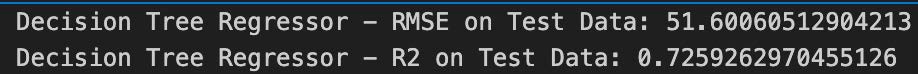
They can uncover how factors like weather, airline, or departure times contribute to delays.

Strengths:

Easy to understand and visualize.

Handles missing data and categorical variables effectively. Limitations:

Prone to overfitting if not regularized. Performance may degrade with very noisy data.



* 1. Random Forest Regressor Why Useful:

Random forests reduce overfitting by averaging multiple decision trees.

Good for identifying important features that contribute to flight delays, such as carrier type or weather conditions.

Strengths:

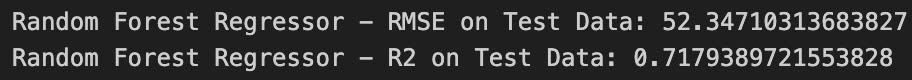
Handles both linear and non-linear relationships.

Resistant to overfitting compared to individual decision trees. Provides feature importance scores.

Limitations:

Computationally intensive on large datasets.

Predictions are harder to comprehend to single decision trees.



* 1. Support Vector Regressor (SVR) Why Useful:

SVR is effective with non-linear relationships and outliers, which are common in flight delay data. however we removed them

The use of kernels allows it to model complex patterns in the data.

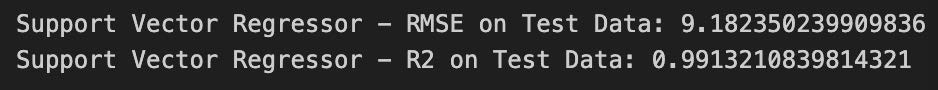
Strengths:

Handles non-linear relationships well.

Can ignore minor noise while focusing on significant patterns by setting error margins.

Limitations:

Computationally expensive for large datasets. Requires careful tuning of hyperparameters.

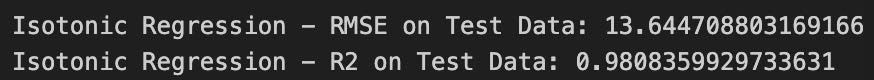


* 1. Isotonic Regressor

Why Useful:

Isotonic Regression is a non-parametric model used for fitting a monotonic relationship between features and a target variable.

In the context of flight delays, it is particularly useful when there is a clear monotonic relationship between features like DEP\_DELAY, ARR\_DELAY, or TAXI\_OUT and the target variable TOTAL\_DELAY. For example, as departure delay increases, the total delay is expected to increase consistently.



* 1. Gradient Boosting Regressor (GBR) Why Useful:

GBR builds models iteratively, correcting errors from previous models, making it highly accurate.

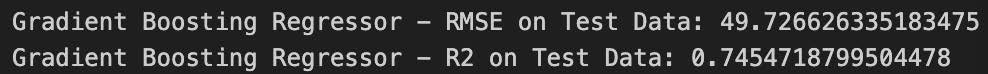
Suitable for capturing subtle patterns in flight delay data. Strengths:

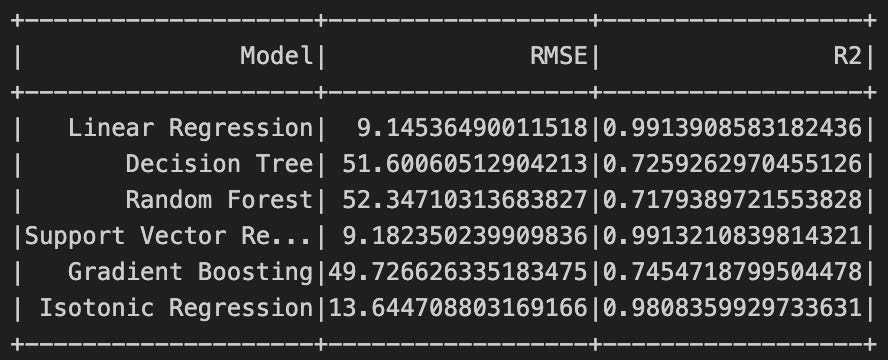
High predictive accuracy for both linear and non-linear relationships.

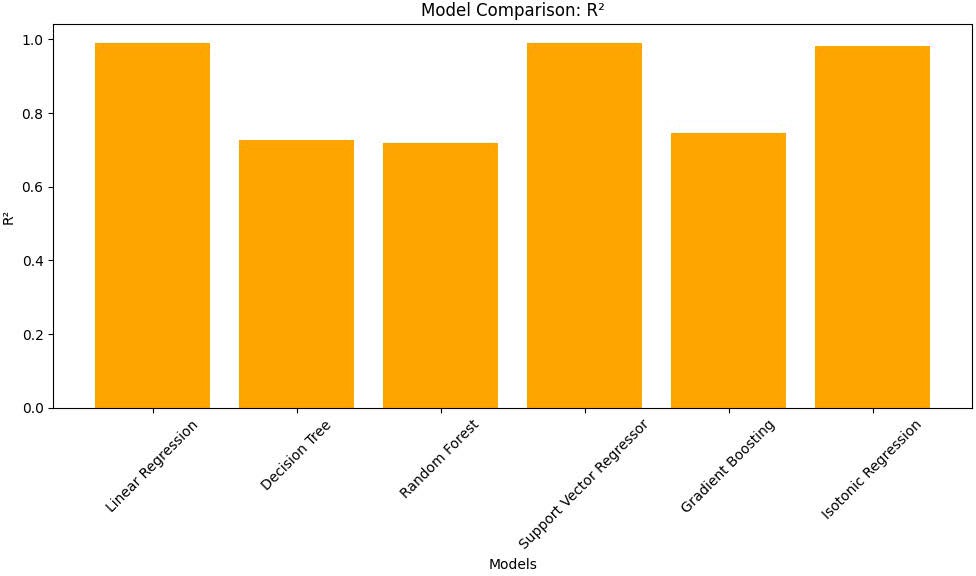
Robust to overfitting with. Provides feature importance metrics.

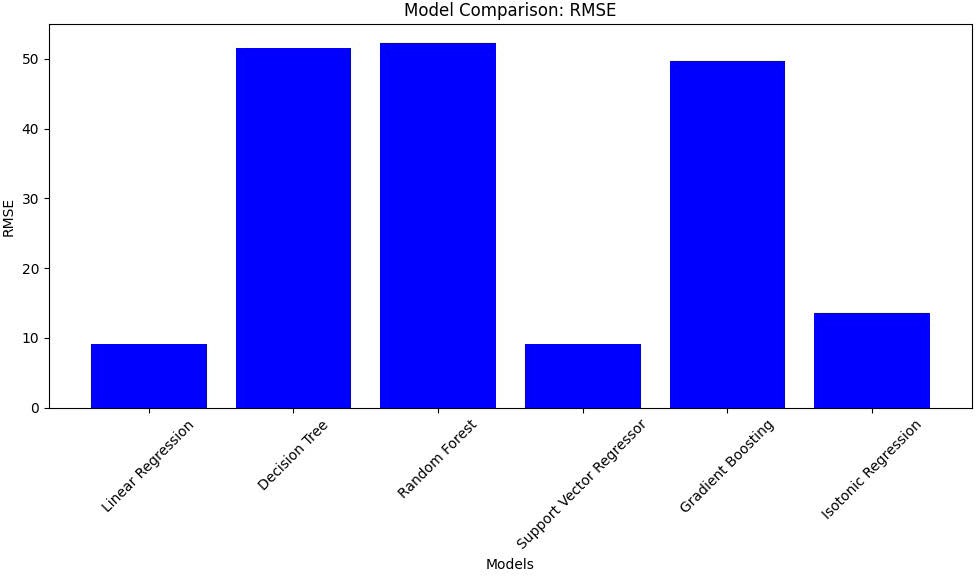
Limitations:

Computationally expensive, especially during training. Sensitive to hyperparameter settings.







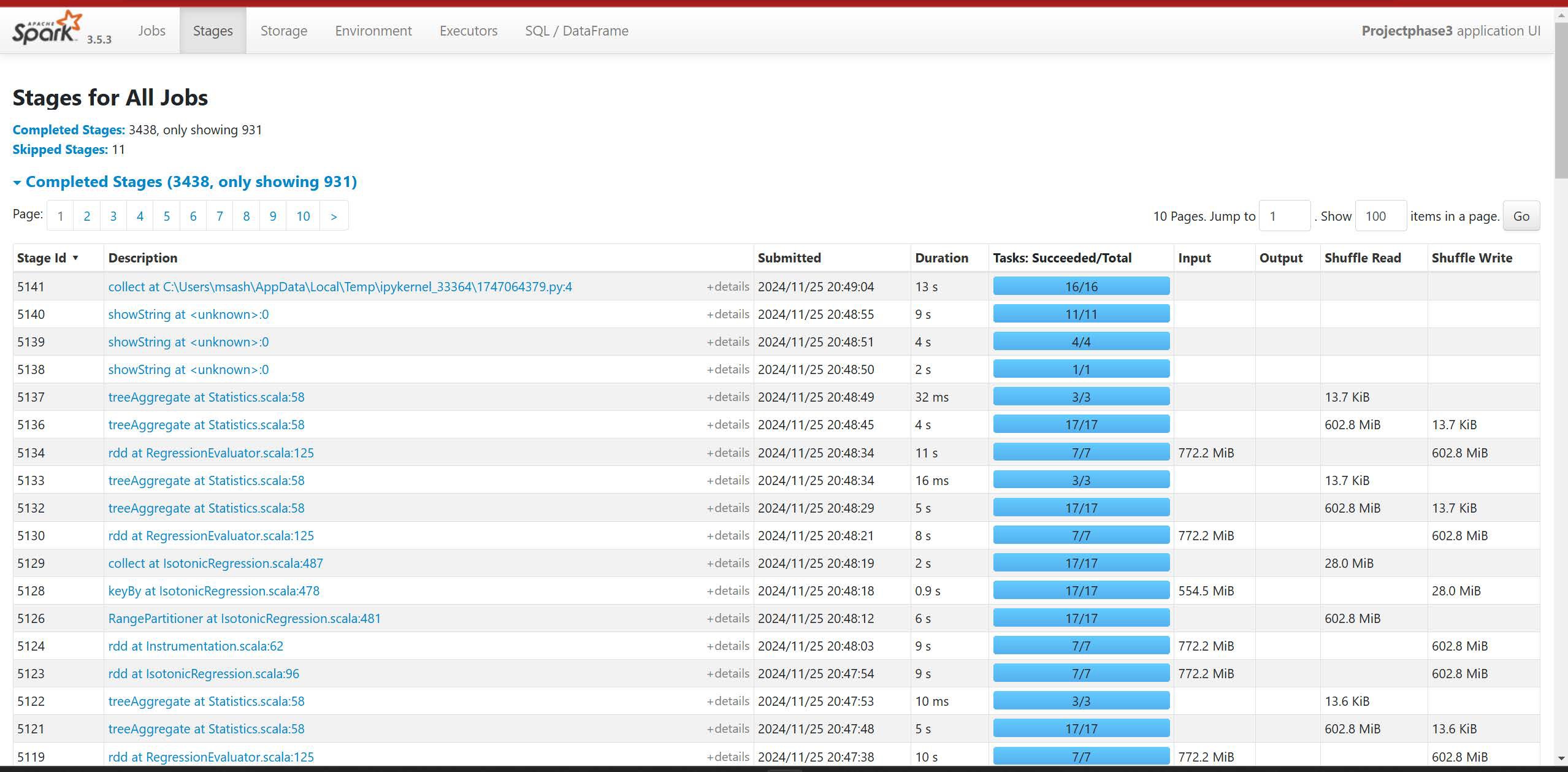


Finding

Advantages of Using Pyspark MLib:

Scalable, and works well with the ecosystem of its own dataframes and apis. Also has DAG visualizations.

A lot of stages were skipped because it is using the previous completed opperation.



**StagHforAIIJobs**

...... ,.,,,,,.,,1>,,..o. -,,,

-\_--

**Comparing Pyspark to Pandas**:

Pyspark worked smoother with larger datasets and

*•-,ow,,n, ......,* " -----

-

-

allowed the flexibilty of utilizing the dag visualizations

\_\_ \_,,.., ,

,.

,,.,

* *-,ow,,n,,.,..,; ,,* -
* .....,,ownn, .....,, ,,\_,,,., ...

..,.,. ..\_\_\_\_ ..\_

-

-

•-101<111n1-·1t 2,\_........

.......,\_,nn,,... o, .,

to understand the steps in the proccess. It also had a built in ecosystem allowing for a much better work enviornment.

..,\_,.,,,\_,

Bonus:

\_m,...,

•-"""''�""'""' " \_m,.,,

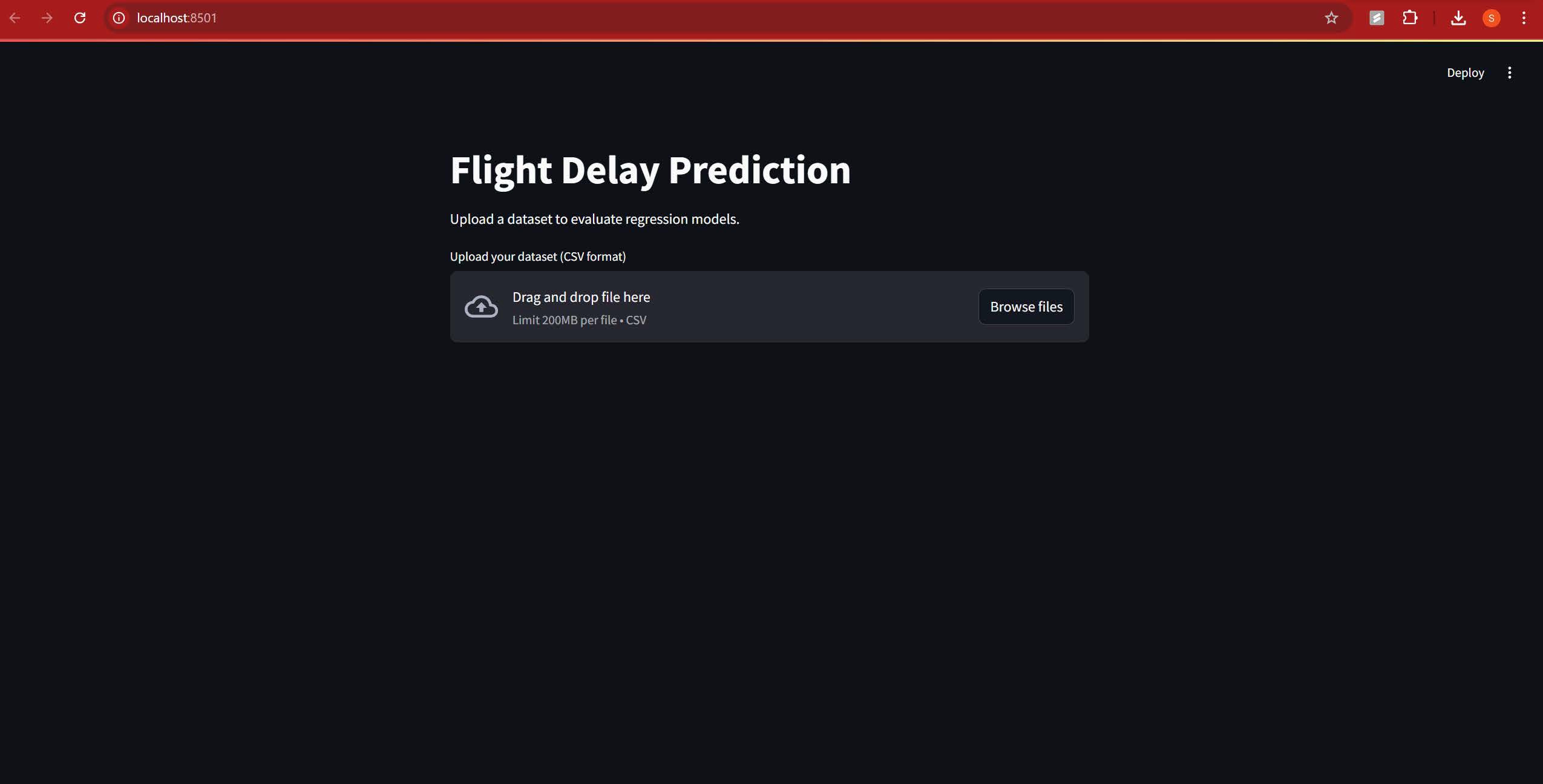
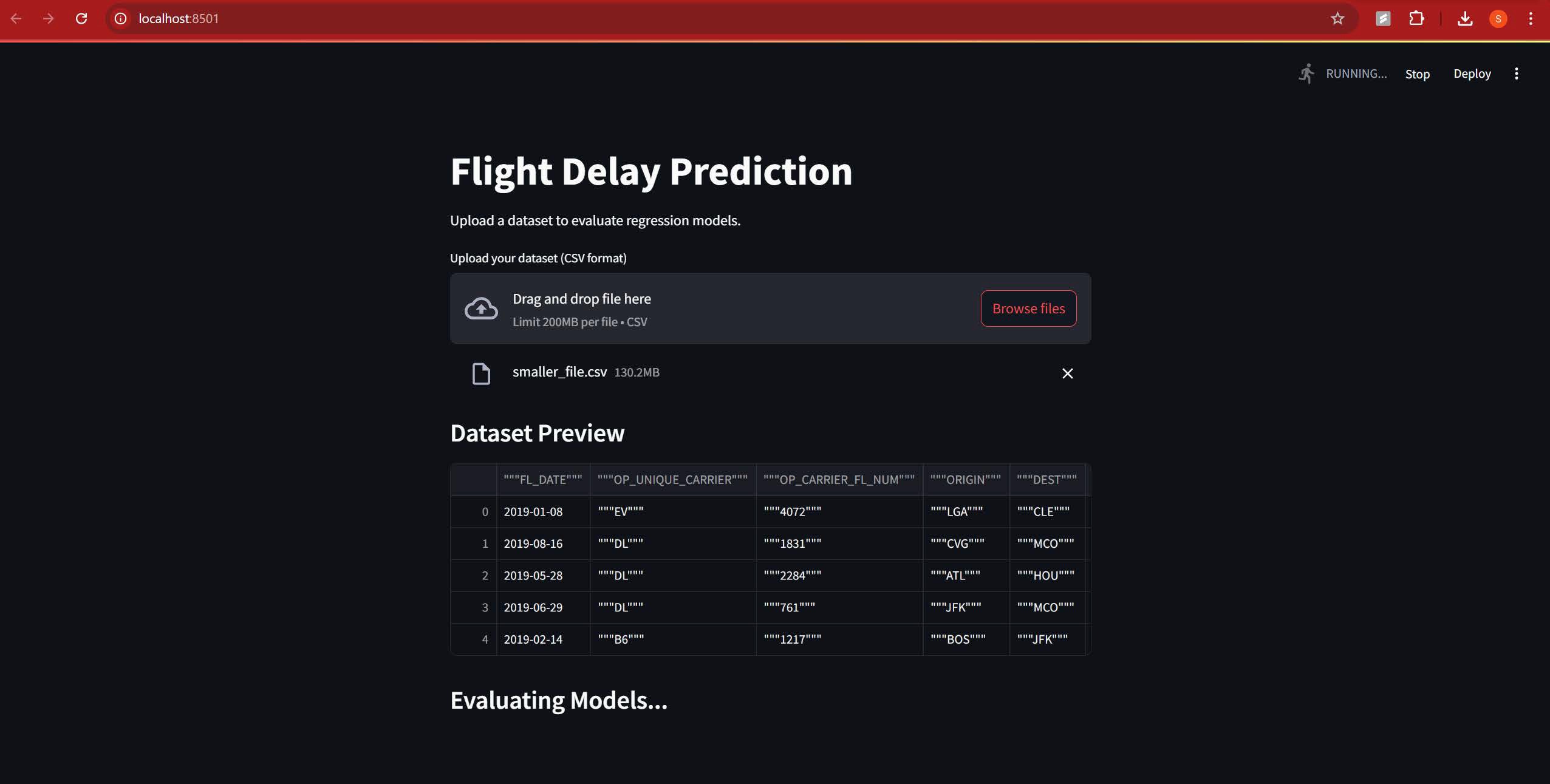
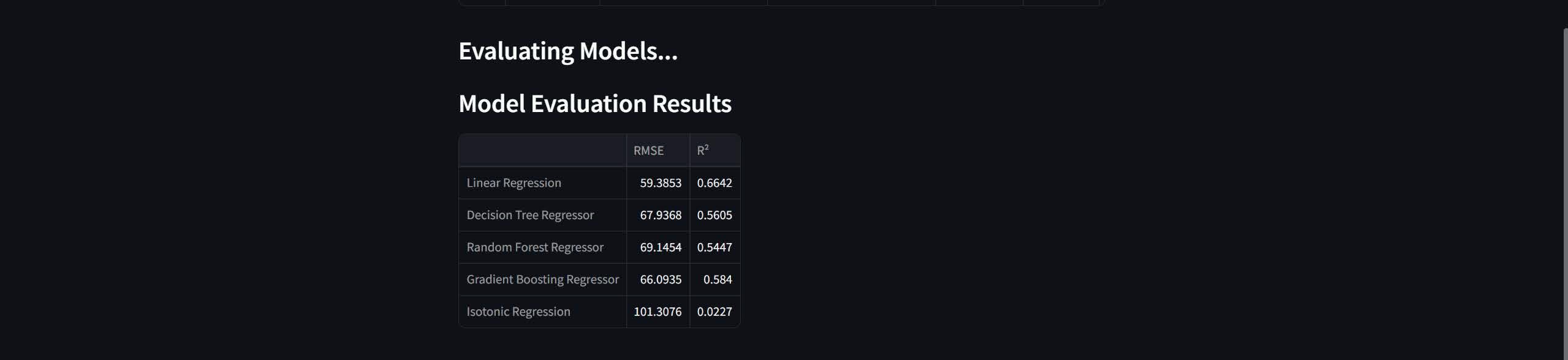
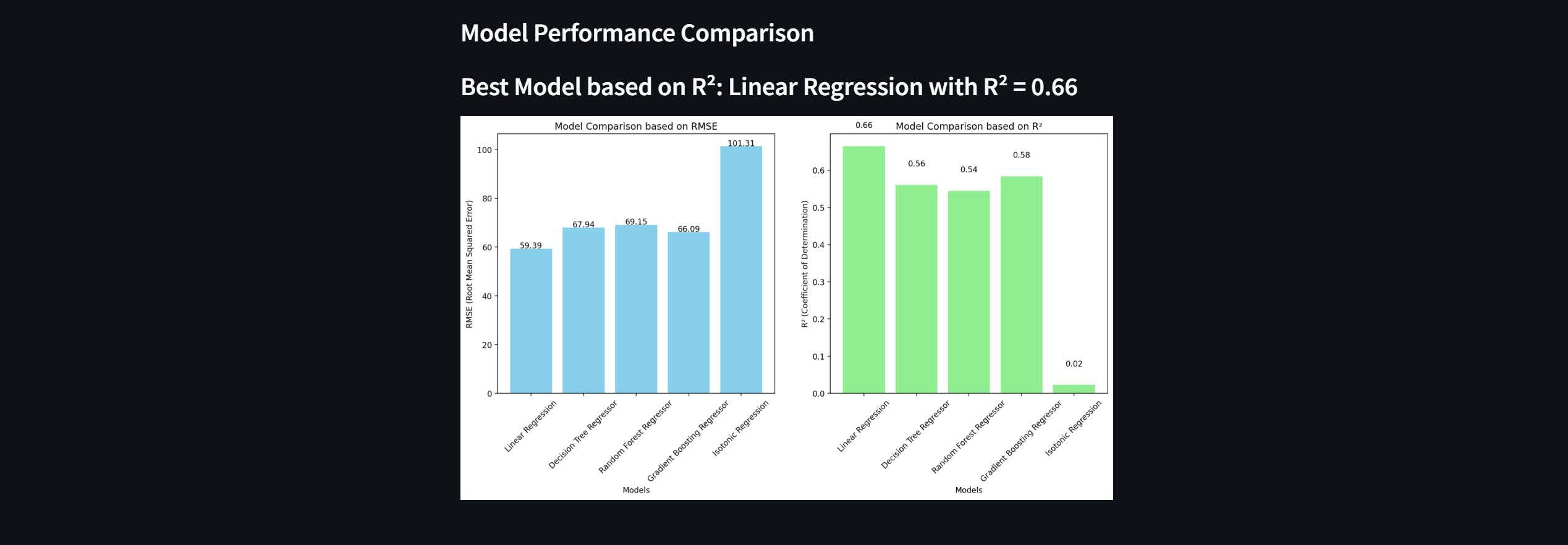
* .,.... ,<1w11m,,,.., .. ,. -

Pandas data frame runs on a single machine. We in general basically use this to run small to moderate data. Whereas PySpark dataframe runs on Apache spark it is designed to run large amounts of data. It is specifically used for data which cannot fit in one single machine.

We load data in the pandas dataframe as a single unit. When there is a large amount of data we often get memory issues. Where in the case of PySpark the data is split into partitions and they are processed parallel through worker nodes. If one worker node fails the other worker node takes care. They are processed parallel across multiple nodes.

In pandas all the operations are executed sequentially. In PySpark the query is optimized byu skipping the actions or operations which are done or completed already. This increases the optimization time.

But pandas dataframe and machine learning model are easy to use, they are easy to set up. Whereas the PySpark setup is difficult, it is more challenging than pandas dataframe.



Link: http://10.84.56.244:8501