Predict Crop Yield based on Environmental and Agricultural Factors

1st Arpit Dilip Sharma
Computer Science and Engineering
SUNY Buffalo
Buffalo, NY, USA
arpitdil@buffalo.edu

2nd Darshan Sunil Sonawane
Computer Science and Engineering
SUNY BUffalo
Buffalo, NY, USA
dsonawan@buffalo.edu

***Abstract*—Agriculture plays significant part in lot of countries GDP. This project aims to predict crop yield, measured in tons per hectare based on environmental and agricultural data which includes variables like rainfall, temperature, etc. To get results extensive data cleaning steps are performed to have best data quality by performing operations on uncleaned data. This will help in optimizing agricultural productivity and decision-making for farmers.**

I. Introduction

Crop yield is important for many reasons including Food security, finding solutions to reduce food waste, environmental impact, market prices. Therefore, calculating the crops yielded in tons per hectare plays a significant role. In our project we aim to calculate this with the help of some key features including region, rainfall, temperature and many more. The initiative intends to help farmers and agribusinesses make educated decisions that improve productivity and sustainability by developing a strong predictive model.

II. Problem Statement

The objective of this project is to estimate the crop yield in tons per hectare based on different environmental inputs such as rainfall, temperature, soil type and agricultural inputs like irrigation, fertilizer usage which are some agricultural inputs. Key question is: How can crop yield be predicted based on environmental and agricultural factors?

III. Potential and Contribution

The purpose of this project is to construct an accurate crop yield prediction model in order to make a contribution to the field of agricultural data analysis. This input is essential for a number of reasons:

1) Data-Driven Decision Making: It presents a method for yield prediction based on data, which lowers uncertainty in farming decisions and encourages more strategic, educated choices.
2) Resource Optimization: The model's ability to predict yields can be used to optimize vital agricultural inputs like fertilizer and irrigation, resulting in more sustainable farming methods and more effective use of resources.
3) Generalizability: Should the model's development be successful, it might be applied to a wide range of crops and geographical areas, which would increase its influence on global agricultural productivity.
4) Data Integrity: The project highlights the significance of preparing data to overcome problems like missing values and inconsistent datasets, ensuring the accuracy and reliability of the predictive model.

By creating a robust and adaptable yield prediction model, this project endeavors to provide a valuable tool for enhancing agricultural productivity and fostering sustainable farming practices.

IV. Distributed Data Cleaning/Processing

A. Initial Data Loading and Deduplication

Operation: Data from uncleaned_crop_yield.csv is loaded into DataFrame with schema inference disabled. Post loading the data duplicate rows are searched and removed using the dropDuplicates() method.

Purpose: Performing the operation ensures the data is free from duplicate entries.

Impact: Critical for maintaining the integrity of dataset, also ensures that data that will be processed is unique.

B. Schema Validation and Data Type Conversion

Operation: Cast method is used to change data types for the specified columns like Rainfall_mm, Temperature_Celsius, Yield_tons_per_hectare from string to double.

Purpose: It preprocesses data types for numerical analysis.

Impact: This will make sure that all operations such as aggregations, mathematical computations, and input to machine learning models go well.

C. Handling Missing Values

Operation: The null columns have to be filled with the mean value of that column, and the categorical columns get filled with 'Unknown'. This is done first by finding the means on numeric columns, then performing the fillna method on these columns.

Purpose: Deals with missing data which results in biased models and imprecise predictions. Filling of missing values contributes to the maintenance of dataset consistency and quality.

Impact: Complete datasets for training improve model reliability, avoiding potential errors that may be caused during model fitting.

D. Trimming Whitespace in Categorical Columns

Operation: The trim function is applied to each categorical column, removing whitespace from string columns. This step is critical in the processing of categorical data, as it ensures that categories are not misunderstood due to leading or trailing spaces.

Purpose: This prevents different interpretations of the same category due to differences in whitespace that needlessly fragment the data.

Impact: Leads to cleaner and more uniform categorical data, which is important to be able to further act on proper grouping, indexing, and encoding.

E. Categorical Data Encoding

Operation: The action consists of indexing categorical columns using StringIndexer and encoding them using OneHotEncoder. This transformation is carried out as part of a pipeline to orchestrate the processing easily.

Purpose: It converts categorical variables into a more meaningful format for ML algorithms

Impact: Prepares the dataset for effective machine learning modeling, by allowing algorithms to correctly interpret categorical data.

F. Outlier Detection and Removal

Operation: Numeric columns are assessed for outliers using the IQR rule. Records with outliers more than 1.5 times the IQR away from the first and third quartiles are filtered out.

Purpose: It reduces the effects of extreme variance in data set values that can result in biased results from statistical analysis and predictive modeling.

Impact: This will enhance the model's accuracy and robustness by concentrating on more representative data and excluding abnormalities that could distort the predictive modeling process.

V. Machine Learning Algorithms

a) All the Algorithms is utilized to predict Yield_tons_per_hectare. All models is trained on a dataset split into 80% training and 20% testing partitions.:

A. Linear Regression

Performance Evaluation:

- **Mean Squared Error (MSE):** 0.4393387154159704
- **R-squared (R^2):** 0.838375178094667

Insights: While the model is strong, its R^2 is very high, which indicates a large part of the variance in yield would be predictable from the feature set.

B. Generalized Linear Regression

Performance Evaluation:

- **Mean Squared Error (MSE):** 0.4393387154159705
- **R-squared (R^2):** 0.8383751780946669

Insights: Coefficients and intercept were provided to reflect the relative influence of each feature in the respective prediction yield.

C. Decision Tree Regressor

Performance Evaluation:

- **Mean Squared Error (MSE):** 0.48116872488850904
- **R-squared (R^2):** 0.8229866689693187

Insights: It returns the importance of each feature, helpful in interpreting what variables most affect the predictions of crop yield.

D. Random Forest Regressor

Performance Evaluation:

- **Mean Squared Error (MSE):** 0.5029987982017232
- **R-squared (R^2):** 0.8149557771138446

Insights: Random Forest provides a more stable estimate than any single decision tree and are less prone to overfitting, but the metrics of performance are slightly lower in comparison with linear models.

E. Gradient-Boosted Tree Regressor

Performance Evaluation:

- **Mean Squared Error (MSE):** 0.4412708072858781
- **R-squared (R^2):** 0.8376643962003764

Insights: The performance will range to the linear regression model, which is also very effective for the crop yield prediction.

F. ElasticNet Regression

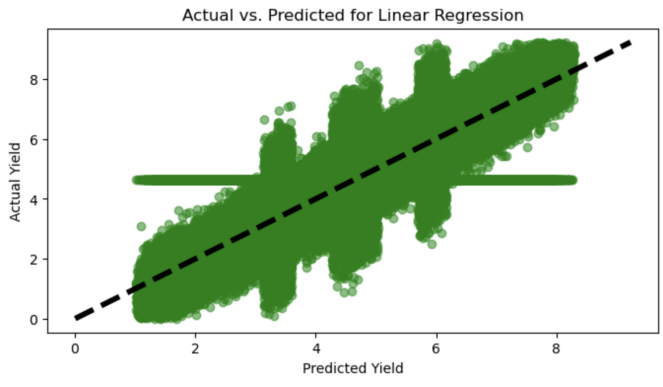
Performance Evaluation:

- **Mean Squared Error:** 0.4393387154159705
- **R-squared:** 0.763195

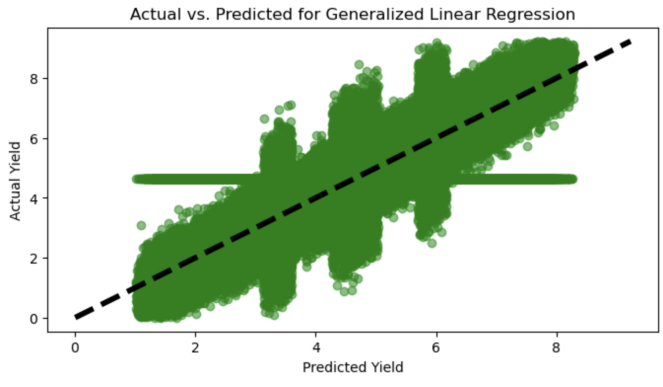
Insights: ElasticNet effectively balances feature reduction and regularization, enhancing model simplicity and interpretability. The strong preference for L1 regularization is especially beneficial in datasets prone to multicollinearity.

VI. Machine Learning Models Visualization

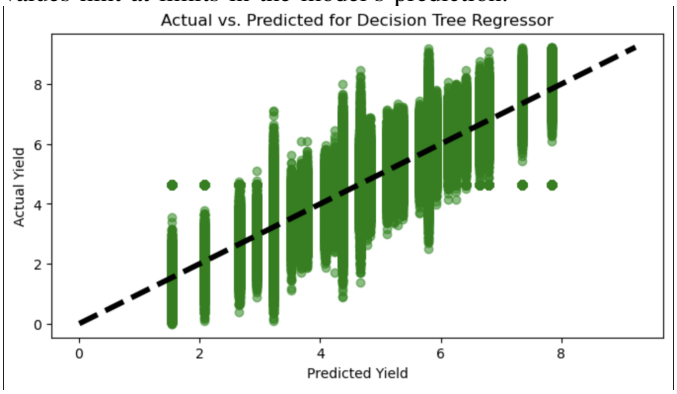
A. Actual vs Prediction



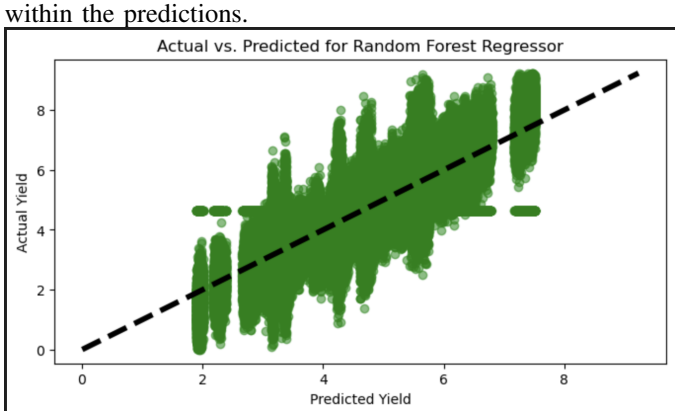
Linear Regression: The plot shows a close alignment of actual yields vs predicted yields, dense points around the line of prediction shows close alignment. Proximity to diagonal dashed line across the output suggests that the Linear Regression model captures the underlying relationship, but deviations at higher yield values hints at limits in the model's prediction.



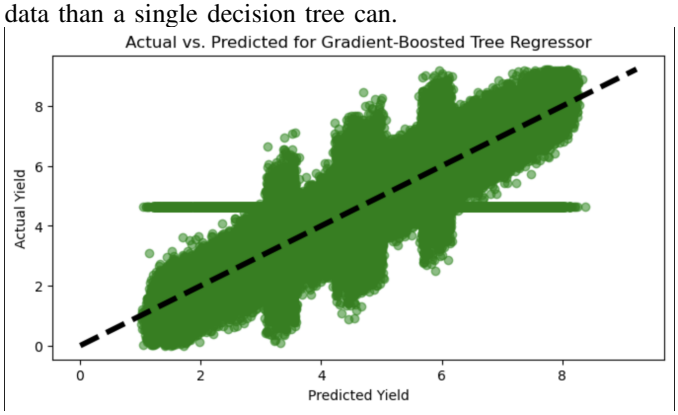
Generalized Linear Regression: The graph above shows the actual yields versus the predicted yields. The closeness of points around the line of prediction shows a high degree of alignment. Nearness to the diagonal dashed line through the output indicates that the model of Linear Regression captures the underlying relationship, but deviations at higher yield values hint at limits in the model's prediction.



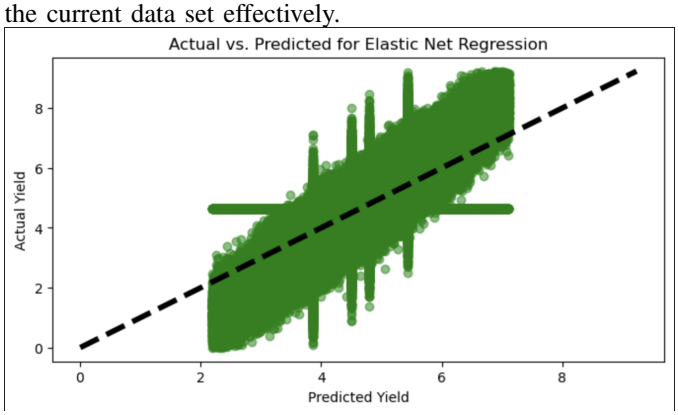
Decision Tree Regressor: Typical of decision tree outputs, the plot contains prominent vertical bands where predictions clump onto specific yield values, reflecting the model's piecewise-constant nature. The trend of the data is captured, but there is not as much resolution or fine-grained detail within the predictions.



Random Forest Regressor: The scatter plot has been showing a most of points hanging dense around the diagonal identity line but with some noticeable spread to showing overall general accuracy of the model, with some variability in prediction. As Random Forest model utilizes several trees, it can better handle outliers and other complexities within data than a single decision tree can.



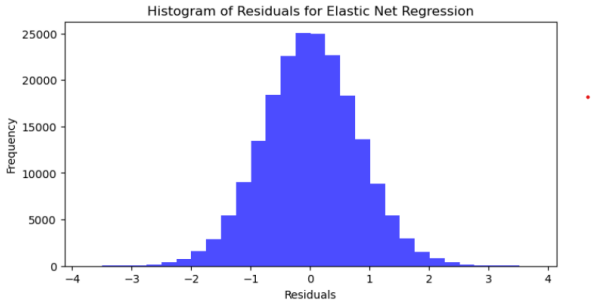
Gradient-Boosted Tree Regressor: Similar to Random Forest, the points in the Gradient Boosted Trees model cluster around the identity line, although with some dispersion. This model outperforms decision trees because it sequentially corrects errors from previous trees, making the clustering tighter, thus handling nonlinearities and interaction effects of the current data set effectively.



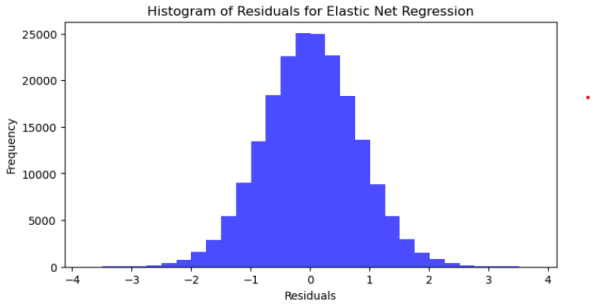
Elastic Net Regression: The actual vs. predicted plot of Elastic Net Regression illustrates a dense packing of data

points along the identity line, where the predictions are in close alignment, indicating very good prediction capability at diverse ranges of yield values. Although there is overall strong performance, some vertical dispersion and a couple of noticeable outliers point to the model sensitivity to those data points or possible overfitting, hence making further tuning of regularization parameters necessary for an optimal balance between bias and variance.

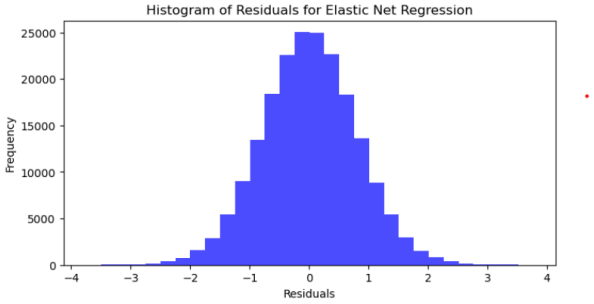
B. Residual Graph:



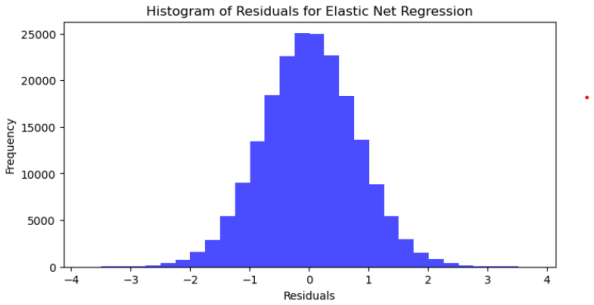
Linear Regression: The model seems to be doing a good job of balancing its predictions. Most residuals are centered around zero, which means the model isn't systematically overestimating or underestimating.The errors are spread out in a way that looks like a normal bell curve, which is generally what we want to see in a well-behaved linear regression model. It suggests the predictions are consistent and reliable.Overall, the graph suggests that the linear regression model is a good fit for the data.



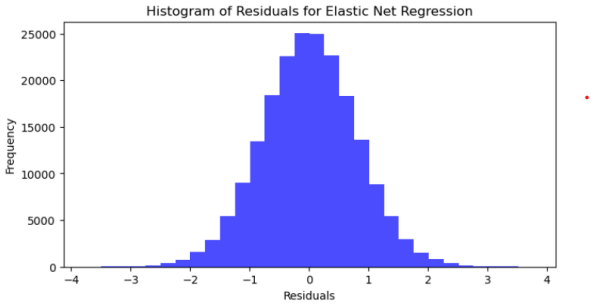
Generalized Linear Regression: The majority of the residuals are close to zero, meaning that for most cases, the model's predictions are accurate. This indicates that the model performs well on the majority of the data. There are relatively few residuals far from zero, meaning the model has few large prediction errors. This indicates robustness in handling diverse data points.



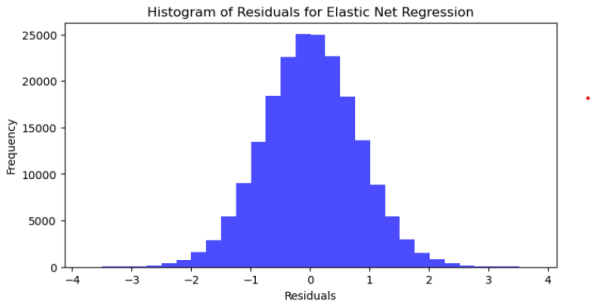
Decision Tree Regressor: A large proportion of residuals are close to zero, which shows that the model's predictions are accurate for most of the data points. However, the peak is slightly less pronounced than in models like linear regression. The model might struggle with overfitting due to its wide spread residuals.



Random Forest Regressor: A significant number of residuals are clustered near zero. This highlights the model's strong performance and accuracy in predicting the majority of the data points. The spread of residuals is slightly narrower than that of the Decision Tree Regressor but not as narrow as simpler linear regression models.



Gradient-Boosted Tree Regressor: The residuals exhibit a relatively narrow spread, signifying that the Gradient-Boosted Tree Regressor minimizes variance while maintaining high accuracy. A significant number of residuals are concentrated near zero, showcasing the model's strong predictive accuracy for the majority of data points.



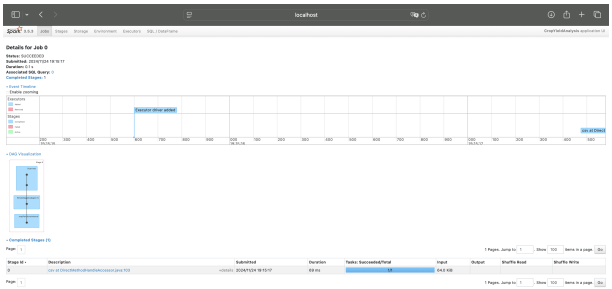
Elastic Net Regressor: Most residuals are concentrated around zero, showcasing good predictive accuracy for the majority of the data points. The residuals display a relatively wider spread compared to models like Gradient-Boosted Trees, Linear Regression.

C. Quantile-Quantile Graph

VII. DAG Visualization Graph:

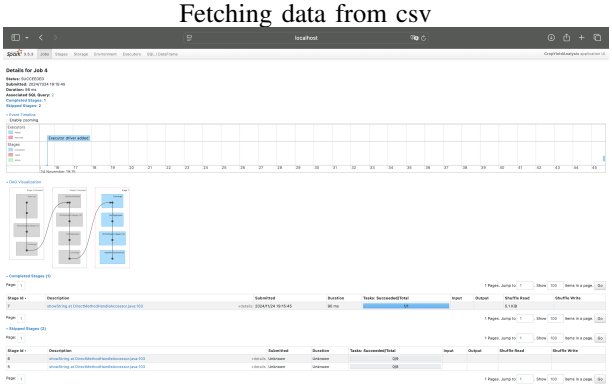
Data Pre-processing and Cleaning

The DAG Visualization graph as shown below is the output visible when you run the program step by step. It is possible to encounter error and rerun which might increase the job no or stage no but in the end the result will be similar as shown below.



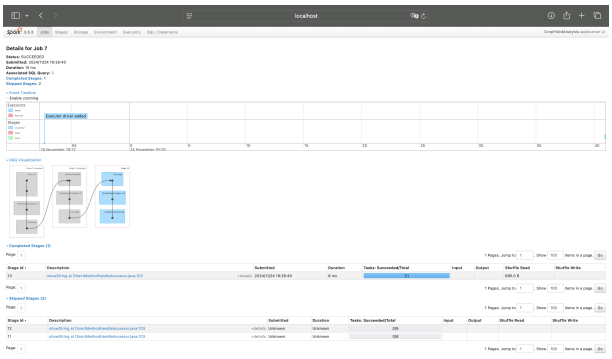
1)

Fetching data from csv



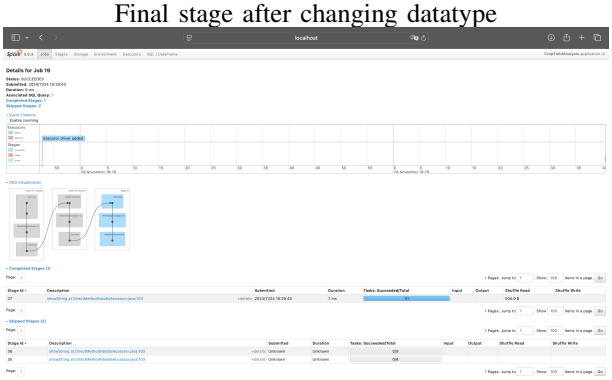
2)

Final stage after performing drop duplicates



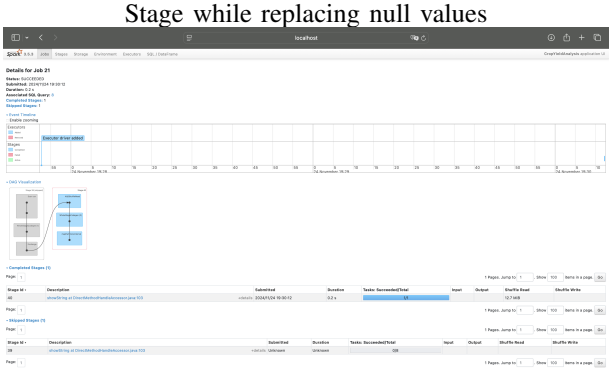
3)

Final stage after changing datatype



4)

Stage while replacing null values



5)

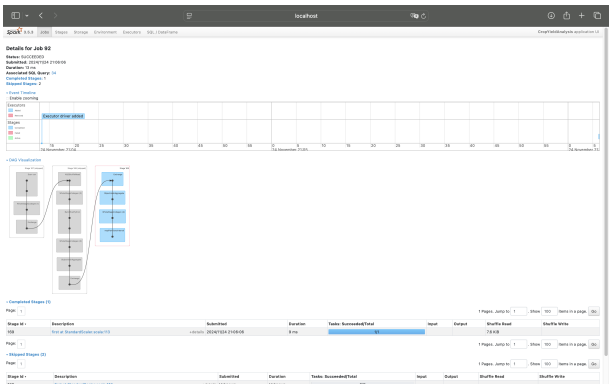
Stage while removing white spaces

6)

Stage while removing outliers and performing one-hot-encoding

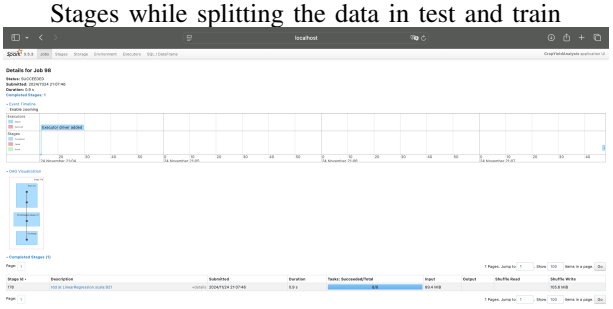
Machine Learning Models

The DAG Visualization graph as shown is the output visible when you run the program stepwise running one model at a time. It is possible to encounter error and rerun the block which might increase the job number or stage number but in the end the result will be similar what is shown below.

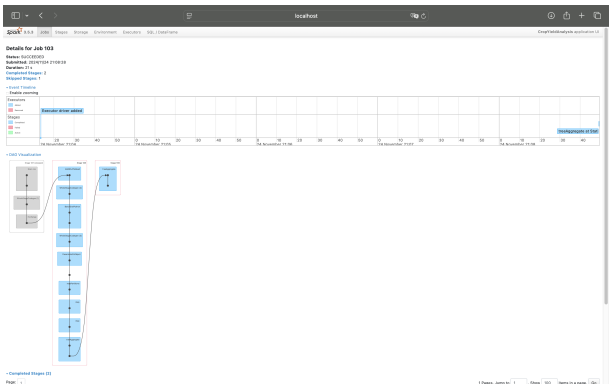


1) Stages while selecting the features

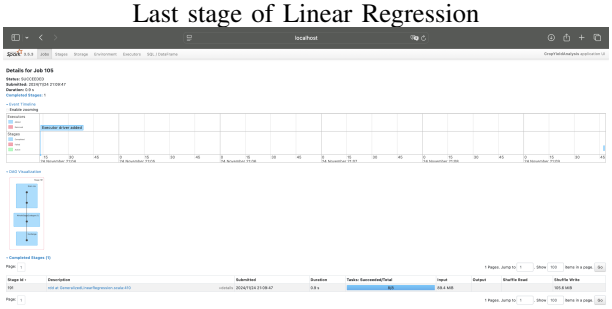
2) Stages while splitting the data in test and train



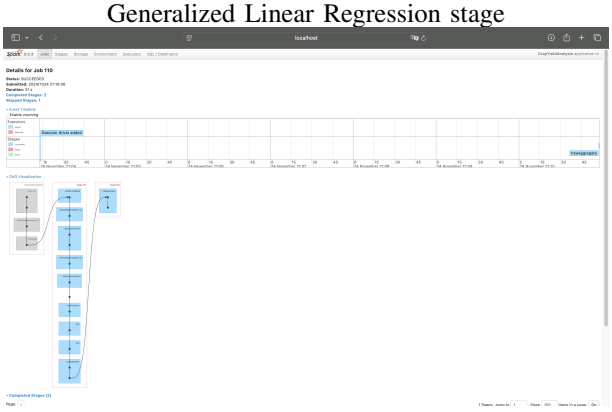
3) Linear Regression stage



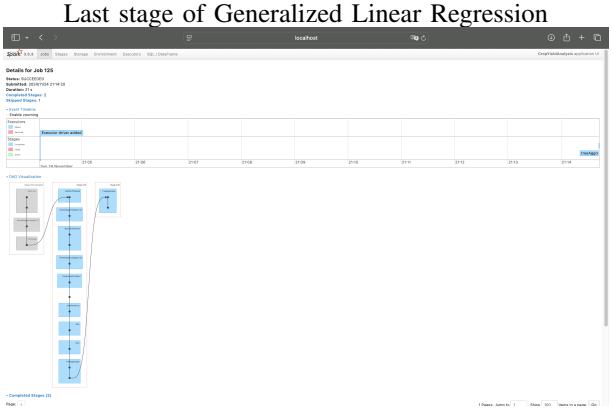
4) Last stage of Linear Regression



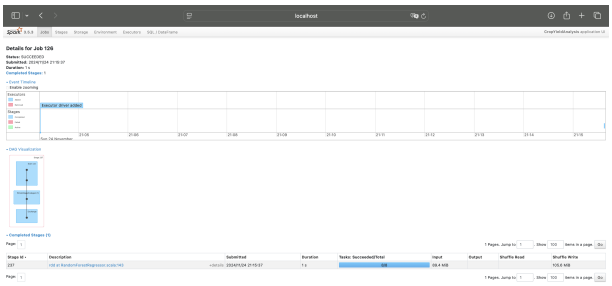
5) Generalized Linear Regression stage



6) Last stage of Generalized Linear Regression

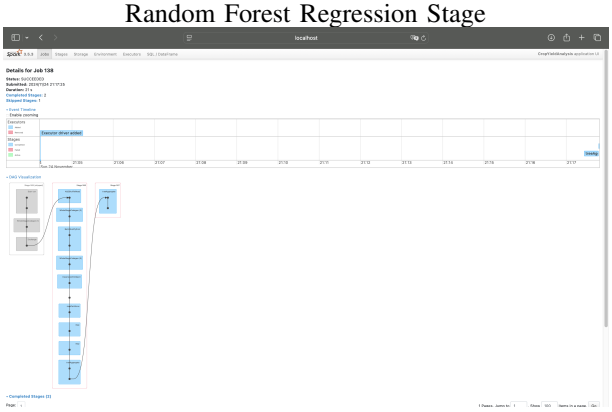


7) Last stage of Decision Tree



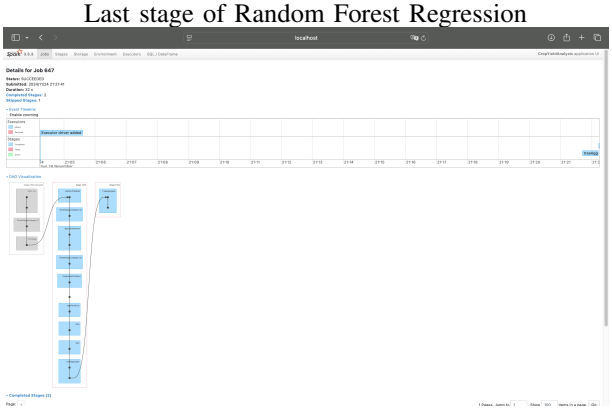
8)

Random Forest Regression Stage



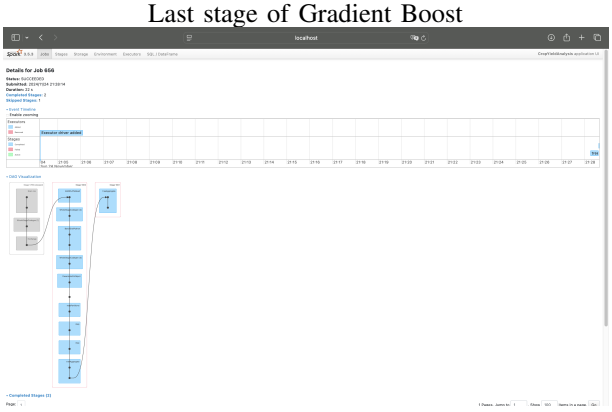
9)

Last stage of Random Forest Regression



10)

Last stage of Gradient Boost



11)

Last stage of Elastic Net Regression

VIII. Comparison with Phase 2 output

A. Time comparison in seconds

Model name	Phase 2	Phase 3
Linear Regression	0.25	52.50
Generalized Linear Regression	Not used	29.86
Decision Tree	5.64	101.44
Random Forest	419.55	94.96
GBT Regression	Not used	164.66
Elastic-Net Regression	0.48	65.99

As above table shows that only Random Forest runs faster when using pyspark library to run the models else most other models run faster without the pyspark library.

B. R^2 comparison

Model name	Phase 2	Phase 3
Linear Regression	0.88	0.83
Generalized Linear Regression	Not used	0.83
Decision Tree	0.75	0.82
Random Forest	0.87	0.81
GBT Regression	Not used	0.84
Elastic-Net Regression	0.76	0.76

Best accuracy is given by models without using pyspark library. Whileusing pyspark library we get best accuracy with GBT(Gradient Boosting Trees) Regression.

C. Effectiveness:

PySpark is noted for its efficiency in handling large quantities of data because it has inherent scalability and speed, based on in-memory computation, performing several tasks much faster compared to traditional big data frameworks like Hadoop MapReduce. With the Python API, it's more approachable and less complex for a big data solution. That makes it very easy for data scientists to create sophisticated analytics-from real-time processing to machine learning and graph processing-all within a single framework. PySpark's integration with a wide range of big data tools and its fault-tolerant architecture make it exceptionally robust and versatile for enterprises desiring to gain the full value of their data, with smooth data processing and minimal time wasted in case of failure, even in a distributed environment.

D. Advantages:

PySpark provides significant benefits for big data because of its great distributed computing capabilities. In this regard, it allows exceptional scalability and speed in handling large datasets efficiently. PySpark is part of the Apache Spark ecosystem that has benefited from in-memory computing, which considerably reduces the execution time of data processing operations compared to disk-based systems like Hadoop. It is especially attractive to data scientists because it also provides an easy-to-use Python API, allowing them to express complex data transformations and analytics using syntax and libraries they are used to. In addition, PySpark also fits well with other big data technologies and platforms, making it even more useful in a variety of contexts. It also grants robust fault

tolerance because of the utilization of Resilient Distributed Datasets; hence, it is reliable and effective for enterprises that want to capitalize on the insights derived from large-scale analytics.

REFERENCES

- https://www.kaggle.com/datasets/samuelotiattakorah/agriculture-crop-yield
  Dataset is clean, had to make it unclean using python script, before performing data cleaning operations.
- https://ieeexplore.ieee.org/document/9432236/figuresfigures
- https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222
- Links used to refer code:
  1. https://seaborn.pydata.org
  2. https://matplotlib.org/stable/index.html
  3. https://www.geeksforgeeks.org
  4. https://www.databricks.com/spark/getting-started-with-apache-spark/machine-learning
  5. https://www.geeksforgeeks.org/pyspark-linear-regression-using-apache-mllib/
  6. https://www.machinelearningplus.com/pyspark/pyspark-decision-tree/
  7. https://www.machinelearningplus.com/pyspark/pyspark-random-forest/
  8. https://www.machinelearningplus.com/pyspark/pyspark-gradient-boosting-model/
  9. https://www.machinelearningplus.com/pyspark/pyspark-lasso-regression/
  10. https://www.geeksforgeeks.org/pyspark-window-functions/