COMPUTER HARDWARE SOFTWARE WORKSHOP
(COCSC19)

Title: Project Synopsis

Submitted by:-

Ashish Kumar(2019UCO1518)
Sandeep Jain(2019UCO1522)
Nishant Goel(2019UCO1529)

TINY ML

Background

Water scarcity has been a hot topic all around the world. We always wondered how we reduce water wastage around our society. Agriculture makes up around 70% of all water usage. So we wondered how we can make the usage of water in agriculture more efficient. So we have tried to make a smart plant automated water system using tiny ml so that we can reduce the wastage of water.

Theoretical framework

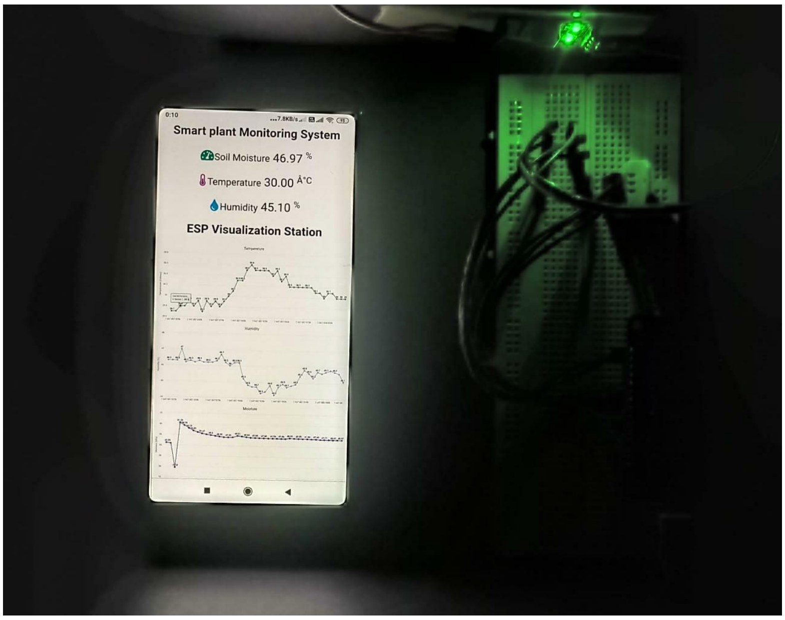
TinyML is a field of study in ML that explores the types of models you can run on small powered devices like microcontrollers. This low power consumption enables the TinyML devices to run unplugged on batteries for weeks, months, and in some cases, even years, while running ML applications on edge.Successful deployment in this field requires intimate knowledge of applications, algorithms, hardware, and software.

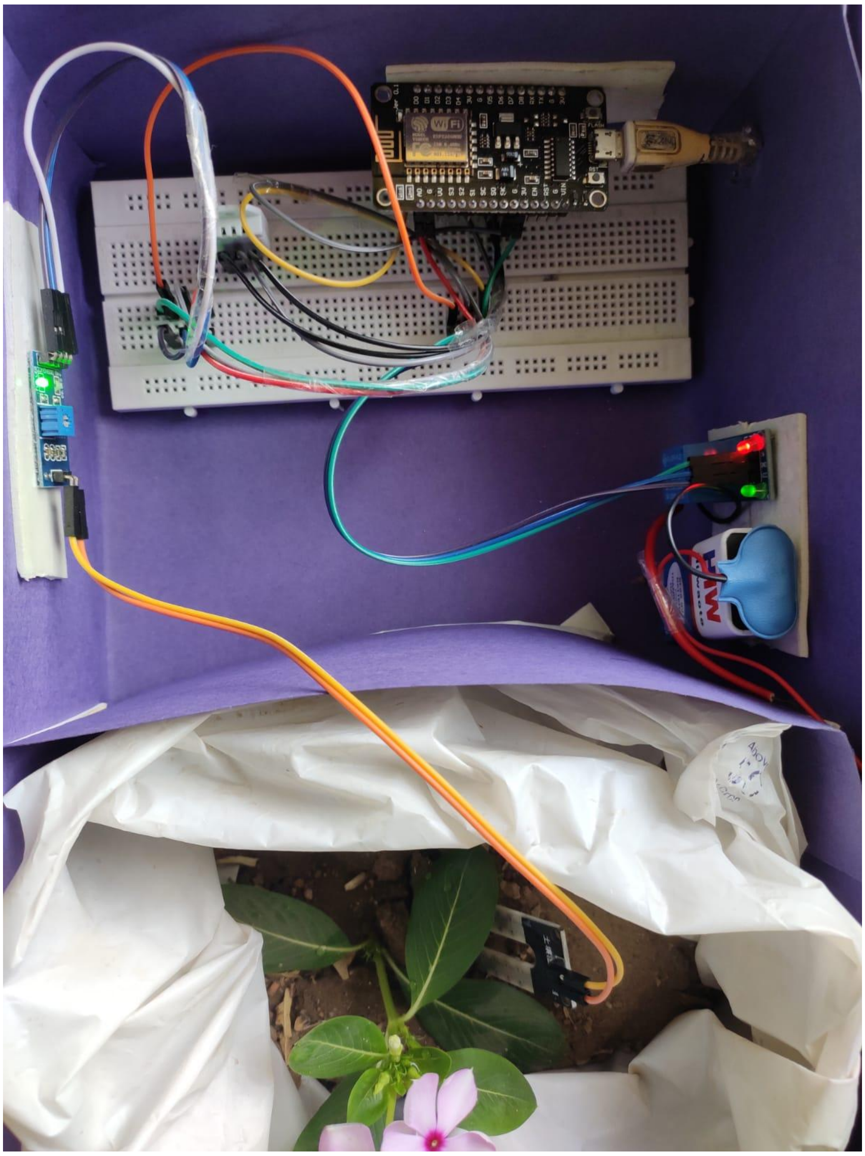
Methodology

We have made a system that will detect plant moisture and temperature and if moisture is less than threshold value it will automatically start watering plants. In this way we have tried to make optimal use of water . We have collected moisture sensor data and then we trained it on ANN model and tried to predict the optimal threshold value.

For theoretical purposes,
we have taken a dataset from kaggle:->
here

Project Images





Data Visualization in R

Background

Recently I watched a movie on netflix and after some time, netflix started recommending me different movies and TV shows. I wondered how the movie streaming platform could suggest content that appealed to me. On further research i found out, all of these recommendations are made possible by the implementation of recommender systems. Recommender systems encompass a class of techniques and algorithms that can suggest “relevant” items to users. They predict future behavior based on past data through a multitude of techniques . The main goal of this project is to build a recommendation engine that recommends movies to users. This R project is designed to understand the functioning of a recommendation system.

Theoretical framework

Data visualization is the technique used to deliver insights in data using visual cues such as graphs, charts, maps, and many others. This is useful as it helps in intuitive and easy understanding of the large quantities of data and thereby make better decisions regarding it. The popular data visualization tools that are available are Tableau, Plotly, R, Google Charts, Infogram, and Kibana. The various data visualization platforms have different capabilities, functionality, and use cases. They also require a different skill set. We are using **R Programming Language for our work.** R is a language that is designed for statistical computing, graphical data analysis, and scientific research. It is usually preferred for data visualization as it offers flexibility and minimum required coding through its packages.

Methodology

In this section of the data science project, we developed the *Item Based Collaborative Filtering System*. This type of collaborative filtering finds similarity in the items based on the people's ratings of them. The algorithm first builds a similar-items table of the customers who have purchased them into a combination of similar items. This is then fed into the recommendation system. The similarity between single products and related products can be determined with the following algorithm:

- For each Item i1 present in the product catalog, purchased by customer C.
- And, for each item i2 also purchased by the customer C.
- Create record that the customer purchased items i1 and i2.
- Calculate the similarity between i1 and i2.

I built this filtering system by splitting the dataset into 80% training set and 20% test set.

Data Preparation

This is conducted in three steps:

1. Selecting useful data
2. Normalizing data
3. Binarizing the data

Data Selection: Through this I visualised the top users and movies through a heatmap. Then I visualized the distribution of the average ratings per user.

Data Normalization: In the case of some users, there can be high ratings or low ratings provided to all of the watched films. This will act as a bias while implementing the model. In order to remove this, I normalized the data. Normalization is a data preparation procedure to standardize the numerical values in a column to a common scale value. This is done in such a way that there is no distortion in the range of values. Normalization transforms the average value of our ratings column to 0. I then plotted a heatmap that portrays our normalized ratings.

Data Binarization: In the final step of the data preparation, in this data science project, I binarized the data. Binarizing the data means that we have two discrete values 1 and 0, which will allow the recommendation system to work more efficiently. I defined a matrix that will consist of 1 if the rating is above 3 and otherwise it will be 0.

Building the recommendation system

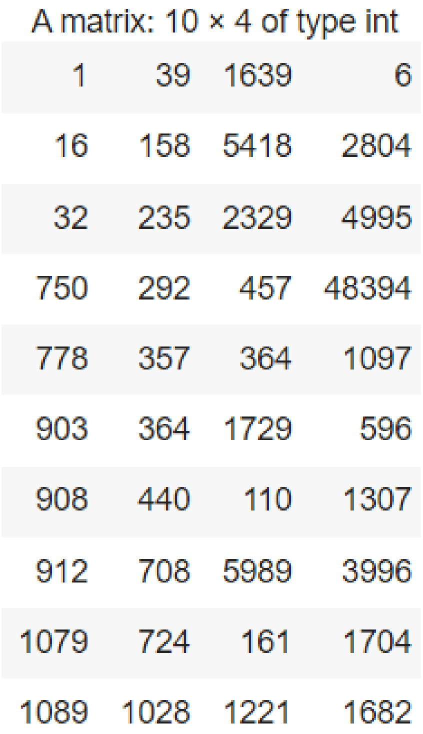
Now, I explored the various parameters of the *Item Based Collaborative Filter*. These parameters are default in nature. In the first step, k denotes the number of items for computing their similarities. Here, k is equal to 30. Therefore, the algorithm will now identify the k most similar items and store their number.

Exploring data science recommendation system model

Using the `getModel()` function, I retrieved the `recommen_model`. I then found the class and dimensions of the similarity matrix, that is, contained within `model_info`. Finally, I generated a heatmap, that will contain the top 20 items and visualize the similarity shared between them.

Collab link:- here

FInal output:-



A matrix: 10 × 4 of type int

1	39	1639	6
16	158	5418	2804
32	235	2329	4995
750	292	457	48394
778	357	364	1097
903	364	1729	596
908	440	110	1307
912	708	5989	3996
1079	724	161	1704
1089	1028	1221	1682

POWER BI

Background

Usa is the biggest economy in the world. American companies are part of our day to day life. America now has more than 500 unicorns. But we wondered which companies are biggest revenue wise and sector wise if compared globally. We not only tried to compare revenue but we analysed various sectors. We have also compared cities with the most number of companies. We have also analysed which companies lead sector wise.

Theoretical Framework

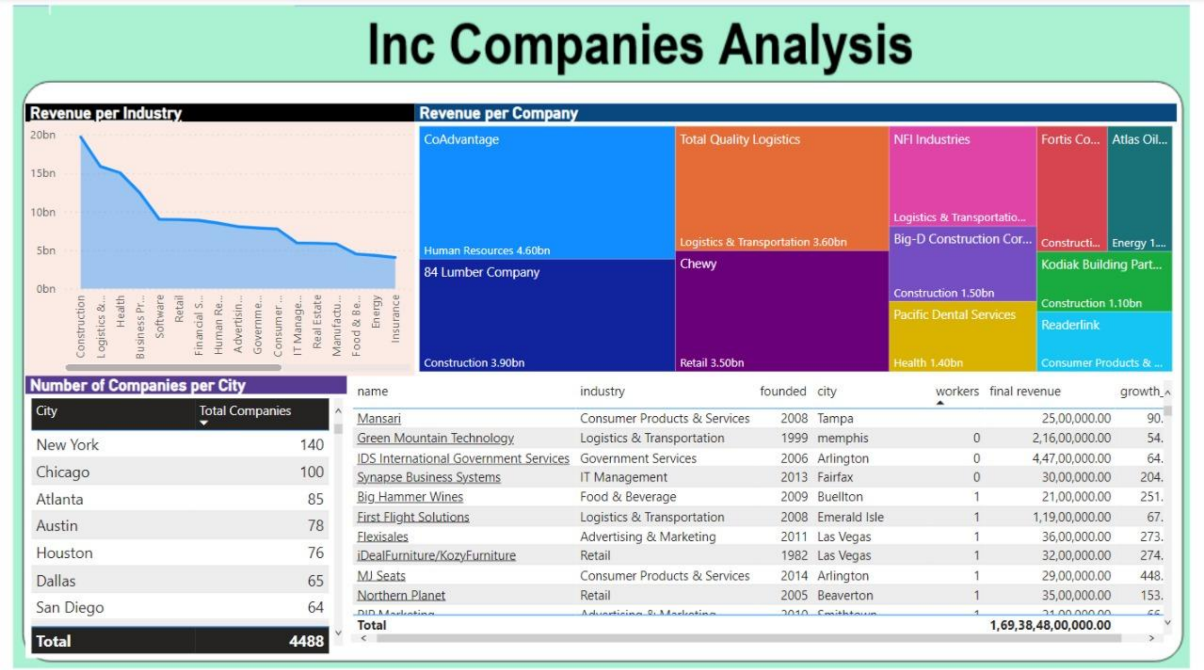
Microsoft power bi is business intelligence that provides nontechnical business users with tools for aggregation, analysing, visualising and sharing data. MIcrosoft Power BI is used to find insights within an organisation's data. Power BI can help connect disparate data sets, transform and clean the data into a data model and create charts or graphs to provide visuals of the data. All of this can be shared with other Power BI users within the organisation. The data models created from power bi can be used in several ways in organisations including telling stories through charts and data visualisations. Power Bi can also provide dashboards for administrators or managers giving management more insight into how departments are doing.

Methodology

Dataset : We have taken the dataset from kaggle and analysed it using power bi tool.

Link for data set ->here

Screenshot of our analysis



Inc Companies Analysis

Revenue per Industry

Revenue per Company

Number of Companies per City

City	Total Companies
New York	140
Chicago	100
Atlanta	85
Austin	78
Houston	76
Dallas	65
San Diego	64
Total	**4488**

name	industry	founded	city	workers	final revenue	growth_
Mansari	Consumer Products & Services	2008	Tampa		25,00,000.00	90.
Green Mountain Technology	Logistics & Transportation	1999	memphis	0	2,16,00,000.00	54.
IDS International Government Services	Government Services	2006	Arlington	0	4,47,00,000.00	64.
Synapse Business Systems	IT Management	2013	Fairfax	0	30,00,000.00	204.
Big Hammer Wines	Food & Beverage	2009	Buellton	1	21,00,000.00	251.
First Flight Solutions	Logistics & Transportation	2008	Emerald Isle	1	1,19,00,000.00	67.
Flexisales	Advertising & Marketing	2011	Las Vegas	1	36,00,000.00	273.
iDealFurniture/KozyFurniture	Retail	1982	Las Vegas	1	32,00,000.00	274.
MJ Seats	Consumer Products & Services	2014	Arlington	1	29,00,000.00	448.
Northern Planet	Retail	2005	Beaverton	1	35,00,000.00	153.
PIP Marketing	Advertising & Marketing	2010	Smithtown	1	21,00,000.00	66.
Total					**1,69,38,48,00,000.00**	

Limitations

- As power bi does not handle the large data sources properly so we can only do analysis on small data set.
- This data set contains only American companies. A wholistic dataset of worldwide companies would be better.

APACHE SPARK

Background

Flight delay is a significant problem that negatively impacts the aviation industry and costs billions of dollars each year. Most existing studies investigated this issue using various methods based on applying machine learning methods to predict the flight delay. However, due to the highly dynamic environments of the aviation industry, relying only on single route of airport may not be sufficient and applicable to forecast the future of flights. The purpose of this project is to analyze a broader scope of factors which may potentially influence the flight delay it compares several machine learning-based models in designed generalized flight delay prediction tasks. In this project we have used flight delay dataset from US Department of Transportation (DOT) to predict flight delays. We have used supervised learning algorithms to predict flight departure delay and then model evaluation is done to get best model and our model can identify which features were more important when predicting flight delays.

Theoretical Framework

Apache Spark (Spark) is an open-source data-processing engine for large data sets. It is designed to deliver the computational speed, scalability, and programmability required for Big Data—specifically for streaming data, graph data, machine learning, and artificial intelligence (AI) applications.

Spark's analytics engine processes data 10 to 100 times faster than alternatives. It scales by distributing processing work across large clusters of computers, with built-in parallelism and fault tolerance. It even includes APIs for programming languages that are popular among data analysts and data scientists, including Scala, Java, Python, and R.

Spark is often compared to Apache Hadoop, and specifically to MapReduce, Hadoop’s native data-processing component. The chief difference between Spark and MapReduce is that Spark processes and keeps the data in memory for subsequent steps—without writing to or reading from disk—which results in dramatically faster processing speeds.

Methodology

The goal of this project is to create a machine learning model (logistic regression and random forest) and predict if a flight will be delayed over 15 minutes using Apache Spark.

The steps we are going to perform are:

1. Loading the data into Apache Spark
2. Reading and processing data with Pyspark
3. Using Logistic Regression and Random Forest to predict delayed flights

We also aim to build a machine learning pipeline with PySpark. For that we have used pyspark.ml.At the core of the pyspark.ml module are the Transformer and Estimator classes.

Transformer classes have a.transform() method that takes a Data Frame and returns a new Data Frame; usually the original one with a new column appended. For example, you might use the class Bucketizer to create discrete bins from a continuous feature or the class PCA to reduce the dimensionality of your dataset using principal component analysis.

Estimator classes all implement a .fit() method. These methods also take a DataFrame, but instead of returning another DataFrame they return a model object. This can be something like a StringIndexerModel for including categorical data saved as strings in your models, or a RandomForestModel that uses the random forest algorithm for classification or regression.

Pipeline is a class which now can be used in the pyspark.ml module that combines all the Estimators and Transformers

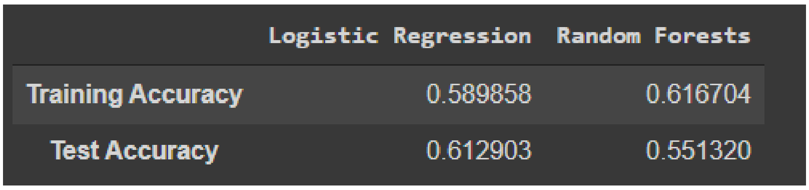
Results:

LINK TO CODE: [(here)]()

LINK TO COLLAB: [(here)]()

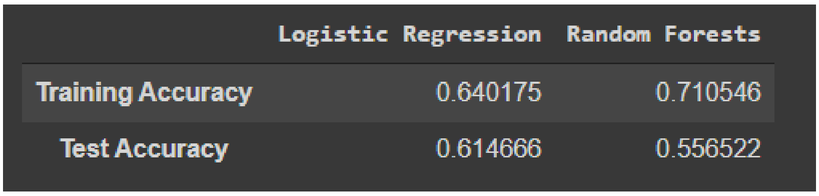
OUTPUT:

Without cross validation:



	Logistic Regression	Random Forests
Training Accuracy	0.589858	0.616704
Test Accuracy	0.612903	0.551320

With cross validation:



	Logistic Regression	Random Forests
Training Accuracy	0.640175	0.710546
Test Accuracy	0.614666	0.556522

DEVOPS

Background:

DevOps is an integration of development and operations teams. It is the union of people, ,processes and technology to continually provide value to customers. DevOps enables formerly siloed roles—development, IT operations, quality engineering and security—to coordinate and collaborate to produce better, more reliable products. By adopting a DevOps culture along with DevOps practices and tools, teams gain the ability to better respond to customer needs, increase confidence in the applications they build and achieve business goals faster.

Theoretical Framework:

DevOps:

Beyond establishing a DevOps culture, teams bring DevOps to life by implementing certain practices throughout the application lifecycle. Some of these practices help accelerate, automate and improve a specific phase. Others span several phases, helping teams create seamless processes that help improve productivity.

Continuous integration and continuous delivery (CI/CD)

Configuration management refers to managing the state of resources in a system including servers, virtual machines and databases. Using configuration management tools, teams can roll out changes in a controlled, systematic way, reducing the risks of modifying system configuration. Teams use configuration management tools to track system state and help avoid configuration drift, which is how a system resource’s configuration deviates over time from the desired state defined for it.

Github Actions:

GitHub Actions makes it easy to automate all your software workflows, now with world-class CI/CD. Build, test, and deploy your code right from GitHub. Make code reviews, branch management, and issue triaging work the way you want.GitHub Actions help you automate your software development workflows from within GitHub. You can deploy workflows in the same place where you store code and collaborate on pull requests and issues.

Machine Learning:

While AI is the broad science of mimicking human abilities, machine learning is a specific subset of AI that trains a machine how to learn. Machine Learning (ML) is a subset of Artificial Intelligence.

We are using Random forests as our model to make predictions of the wine quality.

Random Forests:

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Methodology:

So, in order to introduce features of devOps into our project , we have used Github actions to provide CD/CI to our project.

Steps:

->We have created a new repository with our dataset and our model and also requirements.txt which will tell the deployer which files to be included during deployment.

->Under the Github Actions tab , we have chosed a workflow for our project and created yml file which will describe the build and test jobs to github .

->Add the trigger event to push , so that whenever anything is pushed to main branch , CD will come into action.

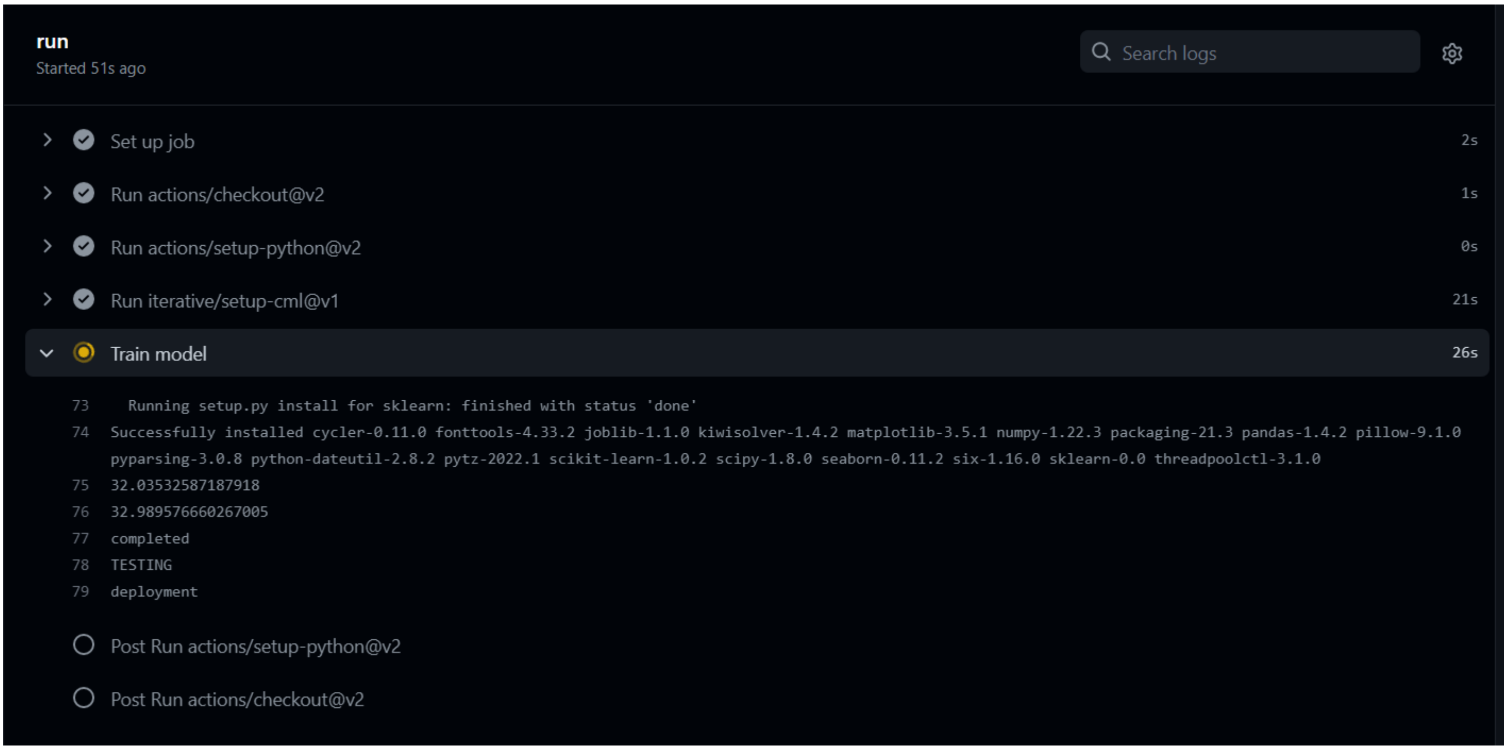
Objective of the Study

We aimed to implement the concepts of DevOps to our machine learning model.We want automation in the production and deployment part of our project.Github actions made it possible by introducing the continuous delivery to our ML project by creating a separate CI/CD pipeline.

Repository Link:

https://github.com/nishantGcode10/devops-project

CD in actions:



run
Started 51s ago

Search logs

✓ Set up job 2s

✓ Run actions/checkout@v2 1s

✓ Run actions/setup-python@v2 0s

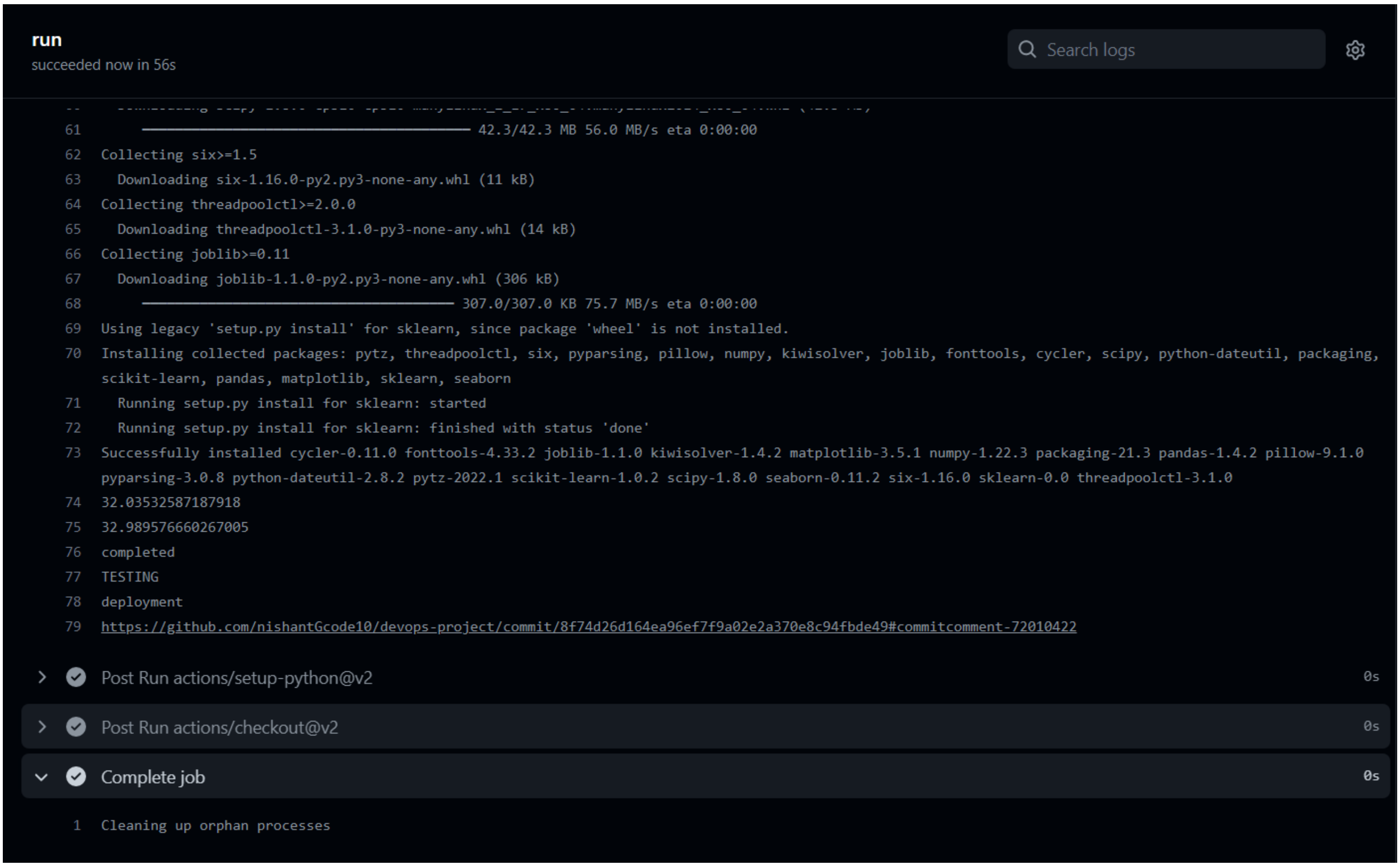
✓ Run iterative/setup-cml@v1 21s

● Train model 26s

```
  Running setup.py install for sklearn: finished with status 'done'
Successfully installed cycler-0.11.0 fonttools-4.33.2 joblib-1.1.0 kiwisolver-1.4.2 matplotlib-3.5.1 numpy-1.22.3 packaging-21.3 pandas-1.4.2 pillow-9.1.0
pyparsing-3.0.8 python-dateutil-2.8.2 pytz-2022.1 scikit-learn-1.0.2 scipy-1.8.0 seaborn-0.11.2 six-1.16.0 sklearn-0.0 threadpoolctl-3.1.0
32.03532587187918
32.989576660267005
completed
TESTING
deployment
```

○ Post Run actions/setup-python@v2

○ Post Run actions/checkout@v2



run
succeeded now in 56s

Search logs

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.3/42.3 MB 56.0 MB/s eta 0:00:00
Collecting six>=1.5
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Collecting threadpoolctl>=2.0.0
  Downloading threadpoolctl-3.1.0-py3-none-any.whl (14 kB)
Collecting joblib>=0.11
  Downloading joblib-1.1.0-py2.py3-none-any.whl (306 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 307.0/307.0 KB 75.7 MB/s eta 0:00:00
Using legacy 'setup.py install' for sklearn, since package 'wheel' is not installed.
Installing collected packages: pytz, threadpoolctl, six, pyparsing, pillow, numpy, kiwisolver, joblib, fonttools, cycler, scipy, python-dateutil, packaging,
scikit-learn, pandas, matplotlib, sklearn, seaborn
  Running setup.py install for sklearn: started
  Running setup.py install for sklearn: finished with status 'done'
Successfully installed cycler-0.11.0 fonttools-4.33.2 joblib-1.1.0 kiwisolver-1.4.2 matplotlib-3.5.1 numpy-1.22.3 packaging-21.3 pandas-1.4.2 pillow-9.1.0
pyparsing-3.0.8 python-dateutil-2.8.2 pytz-2022.1 scikit-learn-1.0.2 scipy-1.8.0 seaborn-0.11.2 six-1.16.0 sklearn-0.0 threadpoolctl-3.1.0
32.03532587187918
32.989576660267005
completed
TESTING
deployment
https://github.com/nishantGcode10/devops-project/commit/8f74d26d164ea96ef7f9a02e2a370e8c94fbde49#commitcomment-72010422
```

✓ Post Run actions/setup-python@v2 0s

✓ Post Run actions/checkout@v2 0s

✓ Complete job 0s

```
Cleaning up orphan processes
```