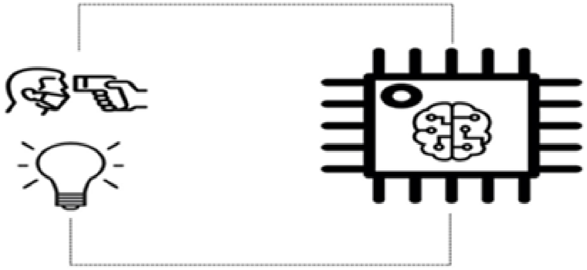
Computer Hardware Software Workshop
Course Code: COCSE19
Practical LAB
Projects Synopsis Submission

Team Members

Rhythm Narula (2019UCO1585)
Abhishek (2019UCO1588)
Mukund Maheshwari (2019UCO1591)
Pulkit Barwal (2019UCO1604)

TinyML

Tiny machine learning (TinyML) is an embedded software technology that can be used to build low power consuming devices to run machine learning models. TinyML is used to build applications that require “full-stack” (hardware, system, software, and applications) solutions, including machine learning architectures, techniques, tools, and approaches capable of performing on-device data analytics at the very edge of the cloud. Tiny ML enables low power, low latency, and low bandwidth model inference at small IoT devices that can run on batteries for months or years without an internet connection. It gives small devices the ability to make smart decisions without needing to send the data to the cloud.



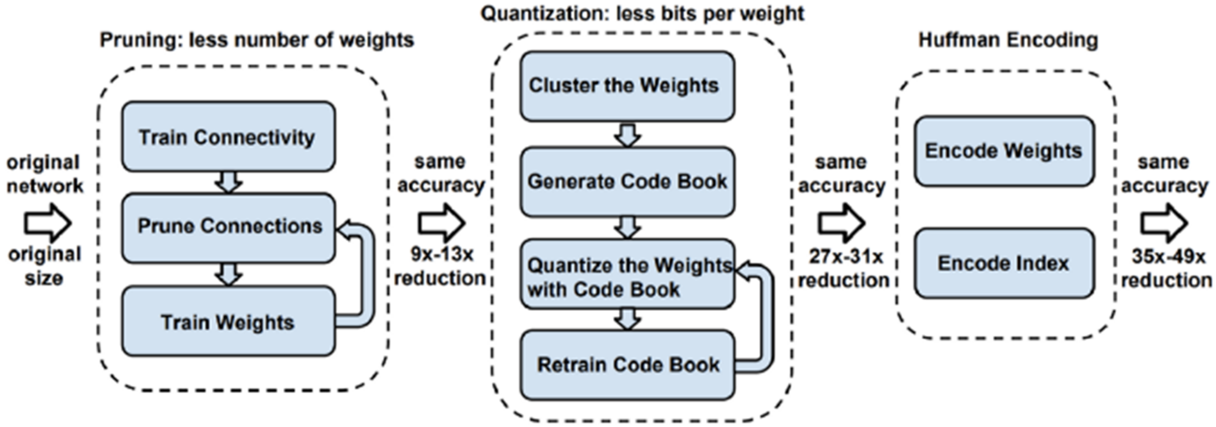
IoT Device – AI in the Microcontroller

Why choose TinyML

1. ML models consume a lot of processing power while TinyML does not require much processing power.
2. Machine learning models uses Tensorflow for creating models while TensorflowLite, CoreML and PyTorch Mobile can be used for TinyML. TensorflowLite offers small binaries capable of running on low power embedded systems. With TensorFlowLite, we can use C, C ++, or Java to create machine learning models.
3. No Internet necessary: TinyML model can run without internet while ML require internet to receive the data on which we have to run the model. If internet goes down, it will be impossible to send data to server in ML.
4. Tiny Machine Learning is a machine learning technique that integrates reduced and optimized machine learning applications that require “full-stack” solutions and approaches capable of performing on-device analytics at the very edge of the cloud. While ML model run on cloud.
5. Usual IoT approach is to collect data and send it to a centralized registration server, and then, you can use machine learning to conclude. On the other hand, TinyML perform the processing on the microprocessor itself so no requirement of data transfer.
6. Data Security: As there is no need to transfer information to external environments, data stays secure and privacy is more guaranteed.
7. Energy savings: Transferring information requires an extensive server infrastructure. When there is no data transmission, energy and resources are saved, consequently in costs. Microcontrollers use low power and can run on batteries for long time.

8. Latency: Data Transfer is not required and thus it takes less time and reduces output latency.

How TinyML Works



Step to build a TinyML Application

1. Obtain a simple dataset.
2. Train a deep learning model.
3. Evaluate the model’s performance.
4. Convert the model to run on-device. (TensorFlow Lite, which is a set of tools for running TensorFlow models on “edge devices”—meaning everything from mobile phones down to microcontroller boards.)
5. Write code to perform on-device inference.
6. Build the code into a binary.
7. Deploy the binary to a microcontroller.

ML frameworks that support TinyML

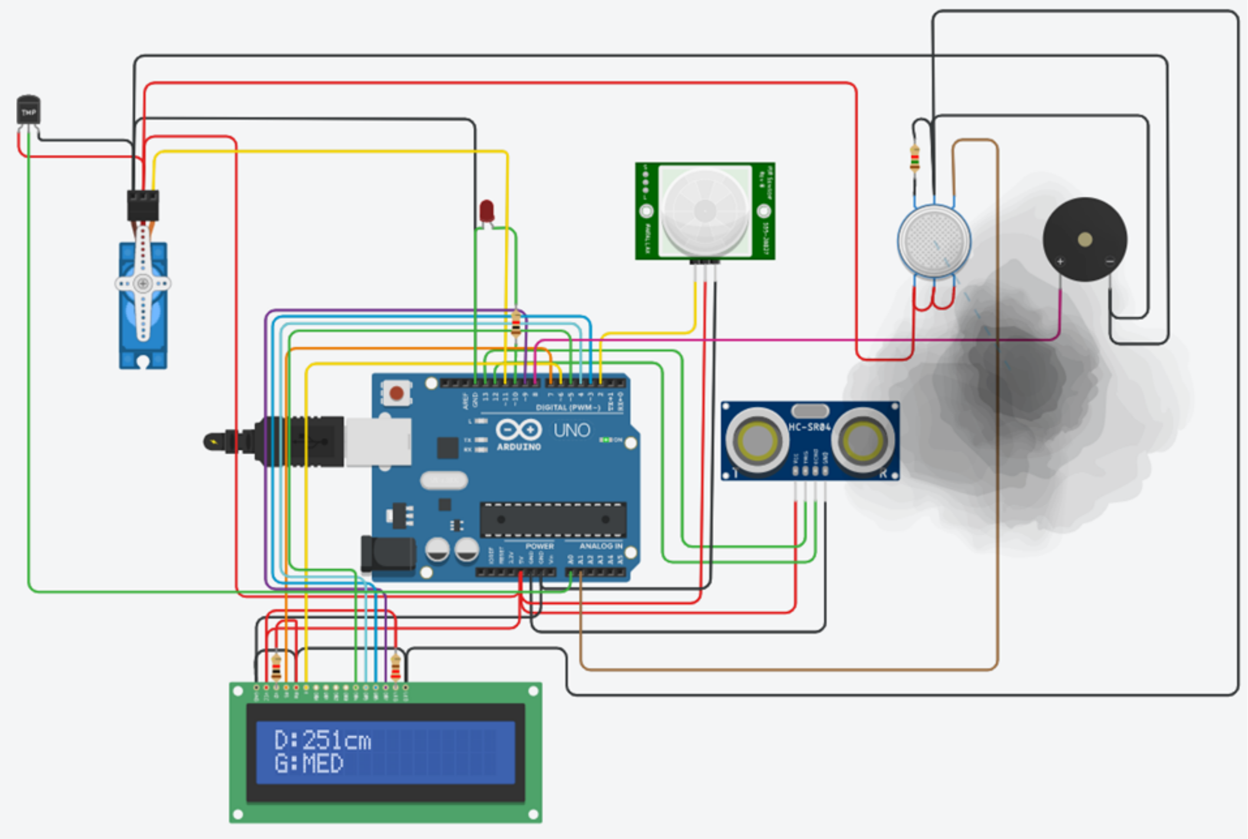
8. TensorFlow Lite
9. PyTorch Mobile (mobile version of Facebook’s PyTorch deep learning library)
10. Edge Impulse
11. CoreML (Apple library)

Applications of TinyML

1. Detect if any plant is having some disease by using the picture of it.
2. It can answer audio commands to execute actions like Google Assistant and Alexa. The devices are always on and analyze your voice to detect the wake word.

3. Predictive maintenance - TinyML can be used to detect faults in a machine ahead of time constantly. It implies maintenance based on predictions.
4. Livestock wearables to monitor vital parameters which can help in predicting illnesses.
5. Fit bands - TinyML can be used to monitor critical as well as palliative care patients in real-time, sending out alerts when emergency action is required.

Application - Home Automation using Tinkercad



R

R is a programming language and software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. R is an integrated suite of software facilities for data manipulation, calculation, and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy,

and a well-developed, simple and effective programming language that includes conditionals, loops, user-defined recursive functions, and input, and output facilities.

The R language is widely used among statisticians and data miners for developing statistical software and data analysis. R is an implementation of the S programming language. R was created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team, of which Chambers is a member. R is named partly after the first names of the first two R authors and partly as a play on the name of S. R is a GNU project. The source code for the R software environment is written primarily in C, Fortran, and R. R is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems. While R has a command-line interface, there are several graphical front-ends available.

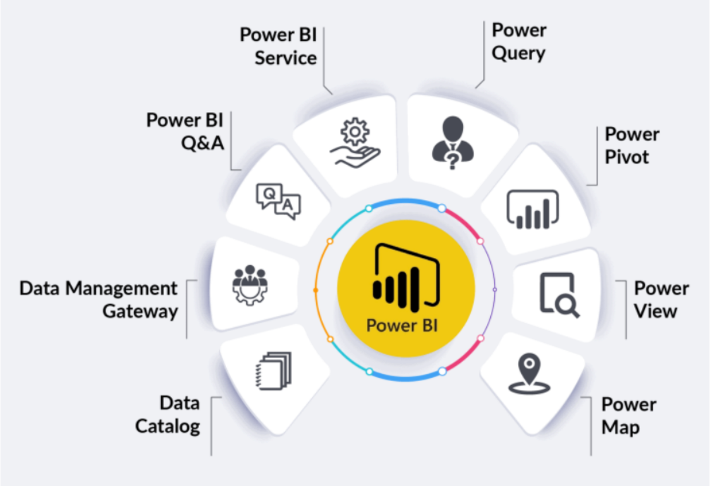
RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging, and workspace management. RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro (Debian/Ubuntu, RedHat/CentOS, and SUSE Linux). RStudio is a free and open-source integrated development environment (IDE) for R, a programming language for statistical computing and graphics. JJ Allaire, the creator of the programming language ColdFusion, founded RStudio.

PowerBI

Microsoft Power BI is a business intelligence platform that provides non-technical business users with tools for aggregating, analyzing, visualizing, and sharing data. Power BI's user interface is fairly intuitive for users familiar with Excel and its deep integration with other Microsoft products makes it a very versatile self-service tool that requires little upfront training.

Why choose Power BI

Microsoft Power BI is one of the commonly used and trusted data visualization and manipulation tools in spite of many other successful tools in the market today. Here are some of the reasons why is it still a preferred visualization tool for businesses:



Types of Power BI Services

To meet each of its customers' role and comfort in creating, consuming and sharing business insights, reports, or visualizations, Microsoft delivers its Power BI Platform in 3 forms of services:

Power BI Desktop: Microsoft Windows application for desktop users. You can install the application from the official site and start using Power BI Desktop on your local machine.

Power BI Web App (Power BI Service): It is an online SaaS (Software as a Solution) Service also which can be accessed via any browser. It provides the same environment and feel like the Power BI Desktop, which an advantage that you can access and use in the browser itself without downloading it.

Mobile Power BI: Available for tablets and mobile phones.

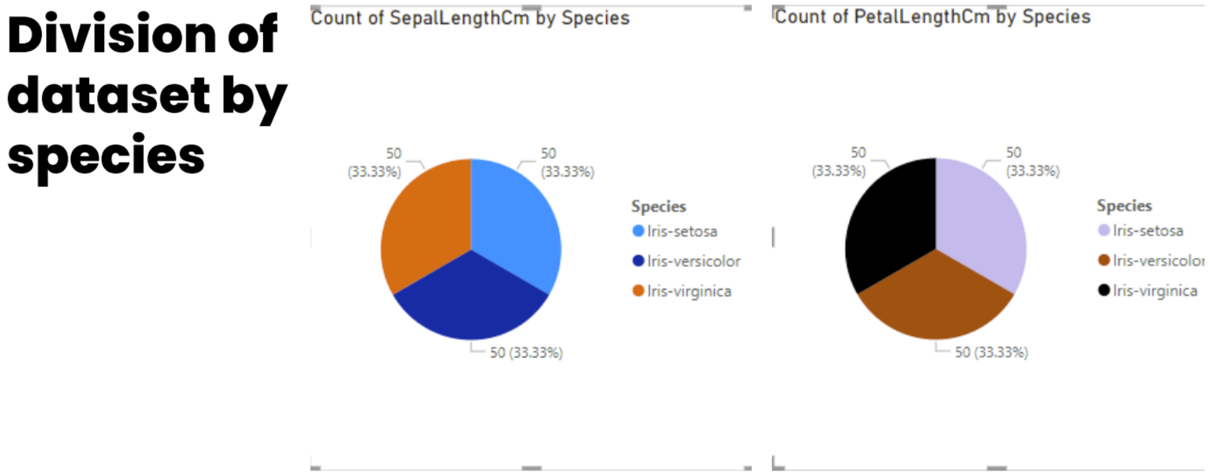
PowerBi visualization on IRIS dataset

Dataset: https://archive.ics.uci.edu/ml/datasets/iris

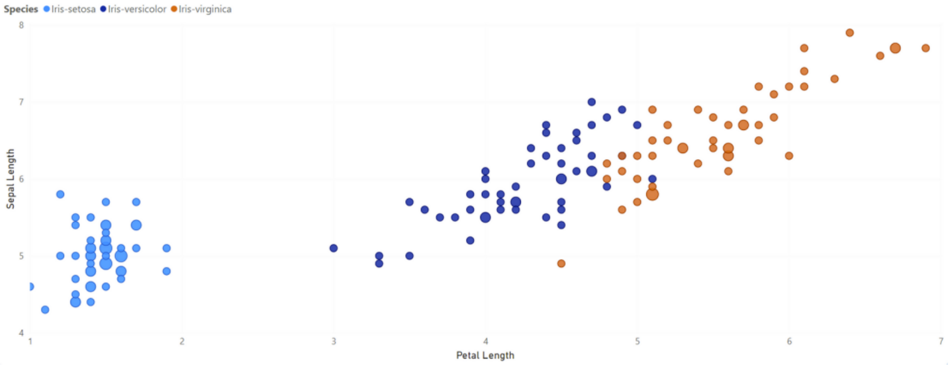
Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
-- Iris Setosa
-- Iris Versicolour
-- Iris Virginica

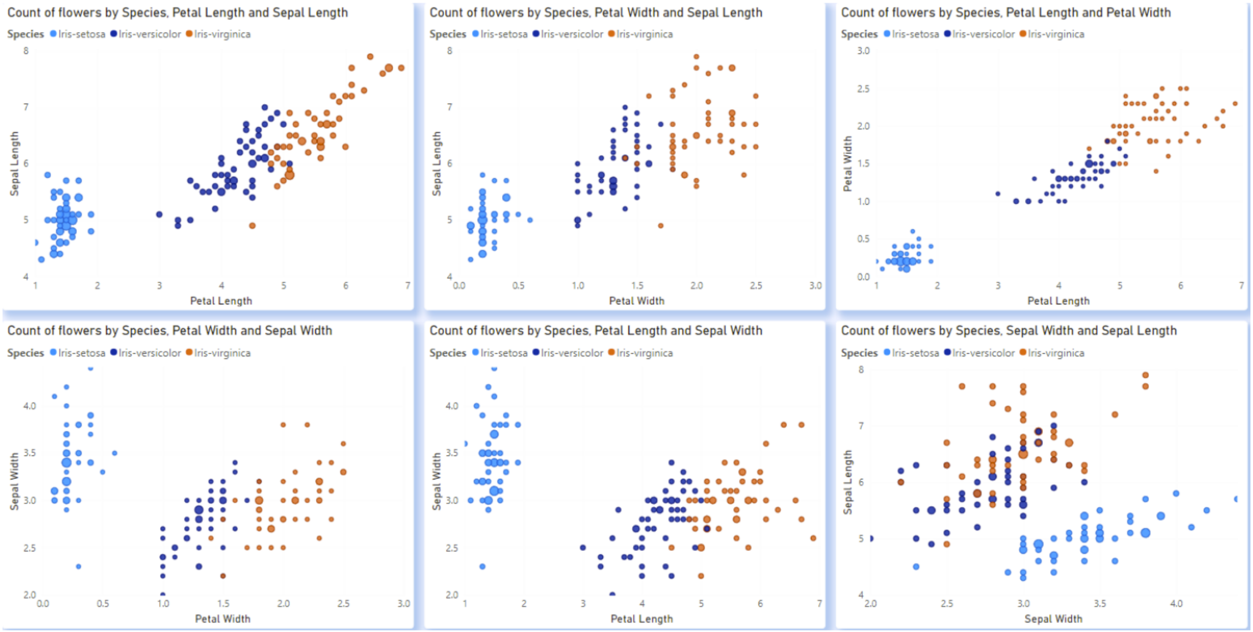
PowerBI visualisation:



Division of dataset by species



Count of flower by Sepal Length, Petal length, species


Count of flowers by Species, Petal Length and Sepal Length
Species ● Iris-setosa ● Iris-versicolor ● Iris-virginica
Sepal Length
Petal Length
Count of flowers by Species, Petal Width and Sepal Length
Species ● Iris-setosa ● Iris-versicolor ● Iris-virginica
Sepal Length
Petal Width
Count of flowers by Species, Petal Length and Petal Width
Species ● Iris-setosa ● Iris-versicolor ● Iris-virginica
Petal Width
Petal Length
Count of flowers by Species, Petal Width and Sepal Width
Species ● Iris-setosa ● Iris-versicolor ● Iris-virginica
Sepal Width
Petal Width
Count of flowers by Species, Petal Length and Sepal Width
Species ● Iris-setosa ● Iris-versicolor ● Iris-virginica
Sepal Width
Petal Length
Count of flowers by Species, Sepal Width and Sepal Length
Species ● Iris-setosa ● Iris-versicolor ● Iris-virginica
Sepal Length
Sepal Width

Apache Spark

Apache Spark is a data processing framework that can quickly perform processing tasks on very large data sets. It can also distribute data processing tasks across multiple computers, either on its own or in company with other distributed computing tools. It is one of the key big data distributed processing frameworks in the world.

Why should Apache spark be opted if Hadoop is already present?

- **Performance** : The main difference between Spark and MapReduce is that Spark runs computations in memory during the later on the hard disk. It allows high-speed access and data processing, reducing time from hours to minutes.
- **Processing** :Hadoop performs batch processing while Spark does both batch as well as real time processing.
- **Execution Time** : Due to more lines of code in Hadoop which is written in Java Execution time increases while Spark is written in Scala having fewer lines of code.
- **Scalability**: When data volume rapidly grows, Hadoop quickly scales to accommodate the demand via Hadoop Distributed File System (HDFS). In turn, Spark relies on the fault tolerant HDFS for large volumes of data.

Features

- **Dynamic in Nature** :We can easily develop a parallel application, as Spark provides 80 high-level operators.
- **Reusability** : Spark code can be reused for batch-processing, join stream against historical data or run ad-hoc queries on stream state.
- **Fault Tolerance in Spark** :Spark RDDs are designed to handle the failure of any worker node in the cluster. Thus, it ensures that the loss of data reduces to zero
- **Support Multiple Languages** : In Spark, there is Support for multiple languages like Java, R, Scala, Python. Thus, it provides dynamicity and overcomes the limitation of Hadoop that it can build applications only in Java.
- **Support for Sophisticated Analysis**: Spark comes with dedicated tools for streaming data, interactive/declarative queries, machine learning which add-on to map and reduce.
- **Integrated with Hadoop**: Spark can run independently and also on Hadoop YARN Cluster Manager and thus it can read existing Hadoop data. Thus, Spark is flexible.

Components of Apache Spark

Spark Core: It is the heart of Spark and performs the core functionality. It holds the components for task scheduling, fault recovery, interacting with storage systems and memory management.

Spark SQL: The Spark SQL is built on the top of Spark Core to provide support for structured and semi-structured data processing.

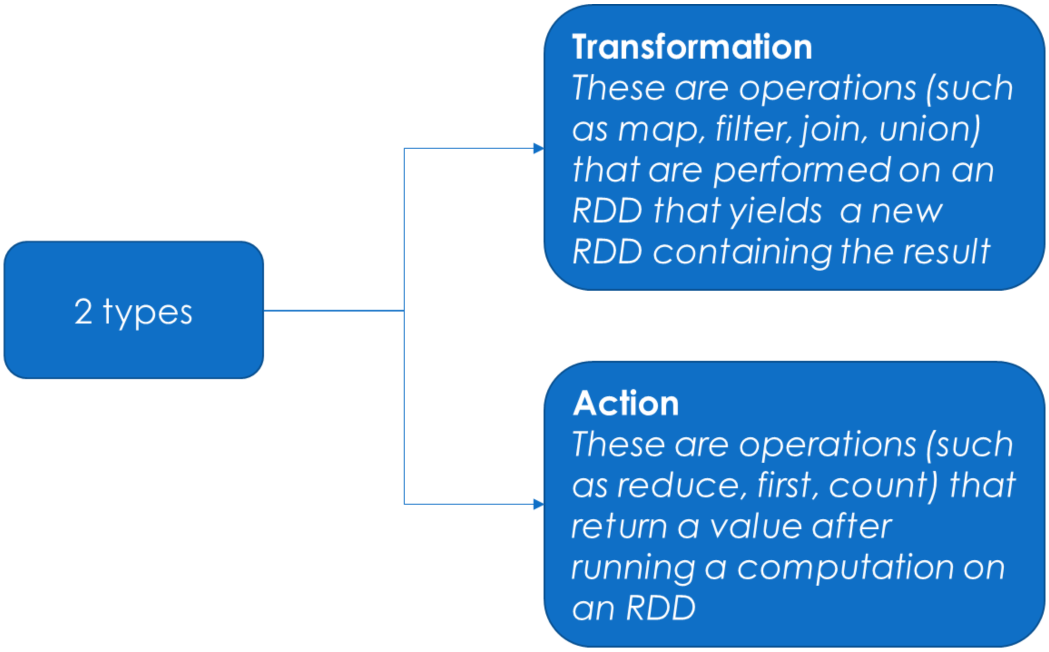
Spark Streaming: It is a Spark component that supports scalable and fault-tolerant processing of streaming data and uses Spark Core's fast scheduling capability to perform streaming analytics.

MLlib: It is a library that contains various machine learning algorithms including correlations and hypothesis testing, classification and regression, clustering, and principal component analysis.

GraphX: It is a library that is used to manipulate graphs and perform graph-parallel computations. It facilitates the creation of a directed graph with arbitrary properties attached to each vertex and edge.

What is a RDD?

The RDD (Resilient Distributed Dataset) is the Spark's core programming abstraction.It represents an immutable collection of objects that can be split across a computing cluster.Operations on the RDDs can also be split across the cluster and executed in a parallel batch process, leading to fast and scalable parallel processing.

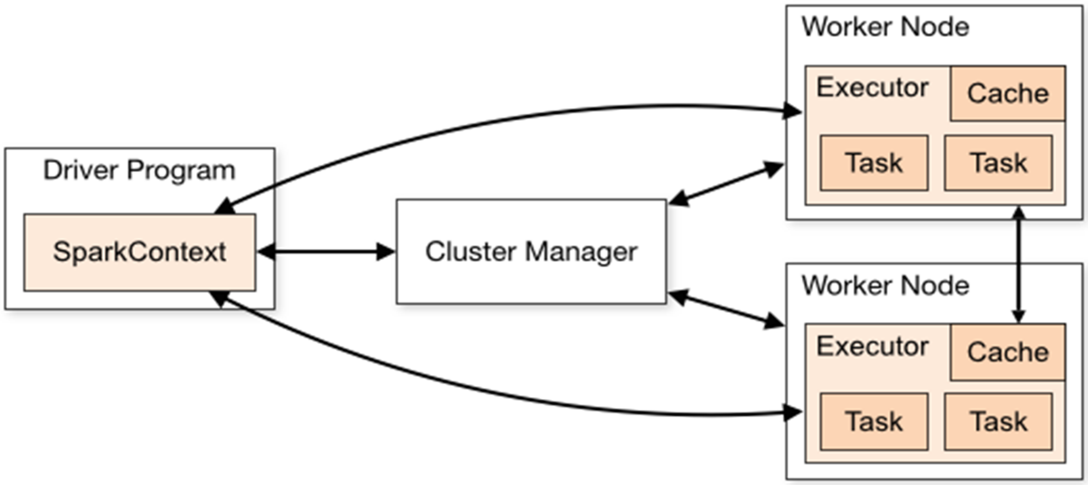


2 types

Transformation
These are operations (such as map, filter, join, union) that are performed on an RDD that yields a new RDD containing the result

Action
These are operations (such as reduce, first, count) that return a value after running a computation on an RDD

Spark Architecture



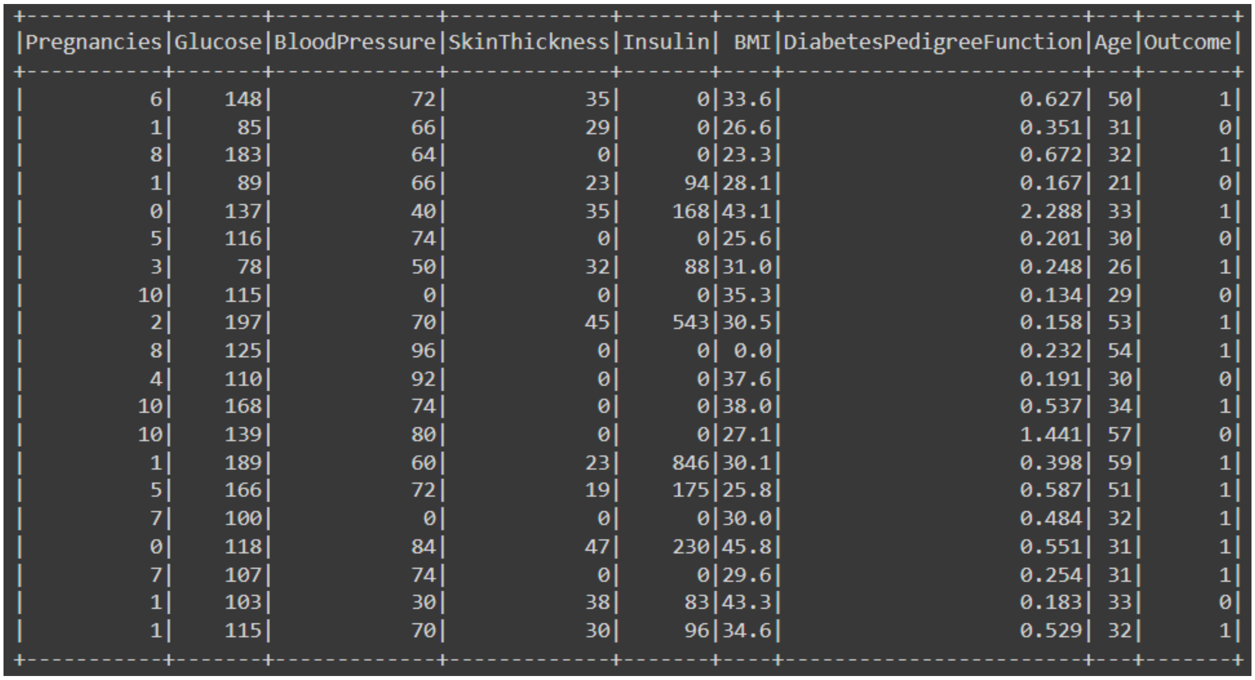
- **Driver Program** :
  It is the controller of the execution of a Spark Application and maintains all of the states of the Spark cluster (the state and tasks of the executors). It must interface with the cluster manager in order to actually get physical resources and launch executors. It is just a process on a physical machine that is responsible for maintaining the state of the application running on the cluster.

- **The Spark executors** :
  Spark executors are the processes that perform the tasks assigned by the Spark driver. Executors have one core responsibility: take the tasks assigned by the driver, run them, and report back their state (success or failure) and results. Each Spark Application has its own separate executor processes.

- **Cluster Manager** :
  The cluster manager is responsible for maintaining a cluster of machines that will run your Spark Application.Whenever a Spark Application is to be executed, we request resources from the cluster manager to run it. Depending on how our application is configured, this can include a place to run the Spark driver or might be just resources for the executors for our Spark Application. Over the course of Spark Application execution, the cluster manager will be responsible for managing the underlying machines that our application is running on.

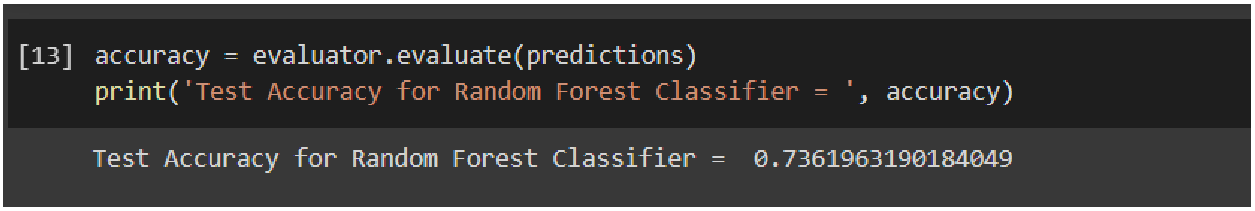
Spark implemented project :

We have used a diabetes dataset available on kaggle to run a distributed database analysis model on it and be able to predict whether a certain person with given qualities has diabetes or not using PySpark.

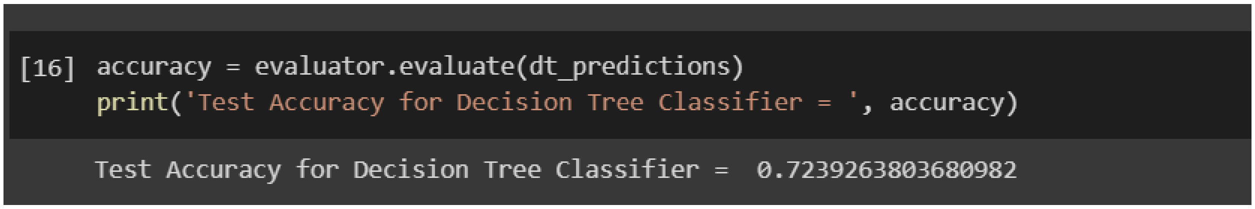


```
+-----------+-------+-------------+-------------+-------+----+------------------------+---+-------+
|Pregnancies|Glucose|BloodPressure|SkinThickness|Insulin| BMI|DiabetesPedigreeFunction|Age|Outcome|
+-----------+-------+-------------+-------------+-------+----+------------------------+---+-------+
|          6|    148|           72|           35|      0|33.6|                   0.627| 50|      1|
|          1|     85|           66|           29|      0|26.6|                   0.351| 31|      0|
|          8|    183|           64|            0|      0|23.3|                   0.672| 32|      1|
|          1|     89|           66|           23|     94|28.1|                   0.167| 21|      0|
|          0|    137|           40|           35|    168|43.1|                   2.288| 33|      1|
|          5|    116|           74|            0|      0|25.6|                   0.201| 30|      0|
|          3|     78|           50|           32|     88|31.0|                   0.248| 26|      1|
|         10|    115|            0|            0|      0|35.3|                   0.134| 29|      0|
|          2|    197|           70|           45|    543|30.5|                   0.158| 53|      1|
|          8|    125|           96|            0|      0| 0.0|                   0.232| 54|      1|
|          4|    110|           92|            0|      0|37.6|                   0.191| 30|      0|
|         10|    168|           74|            0|      0|38.0|                   0.537| 34|      1|
|         10|    139|           80|            0|      0|27.1|                   1.441| 57|      0|
|          1|    189|           60|           23|    846|30.1|                   0.398| 59|      1|
|          5|    166|           72|           19|    175|25.8|                   0.587| 51|      1|
|          7|    100|            0|            0|      0|30.0|                   0.484| 32|      1|
|          0|    118|           84|           47|    230|45.8|                   0.551| 31|      1|
|          7|    107|           74|            0|      0|29.6|                   0.254| 31|      1|
|          1|    103|           30|           38|     83|43.3|                   0.183| 33|      0|
|          1|    115|           70|           30|     96|34.6|                   0.529| 32|      1|
+-----------+-------+-------------+-------------+-------+----+------------------------+---+-------+
```

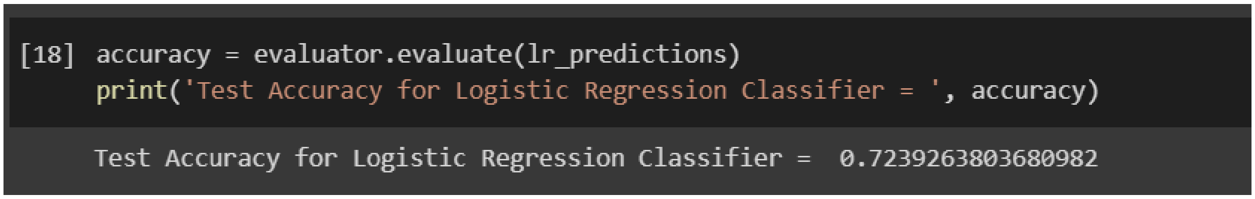
Result:



```
[13] accuracy = evaluator.evaluate(predictions)
     print('Test Accuracy for Random Forest Classifier = ', accuracy)

     Test Accuracy for Random Forest Classifier =  0.7361963190184049
```



```
[16] accuracy = evaluator.evaluate(dt_predictions)
     print('Test Accuracy for Decision Tree Classifier = ', accuracy)

     Test Accuracy for Decision Tree Classifier =  0.7239263803680982
```

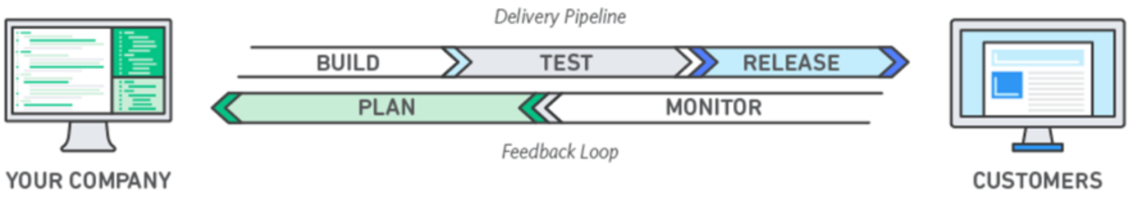


```
[18] accuracy = evaluator.evaluate(lr_predictions)
     print('Test Accuracy for Logistic Regression Classifier = ', accuracy)

     Test Accuracy for Logistic Regression Classifier =  0.7239263803680982
```

The best accuracy was achieved by using a Random Forest Classifier (73.6%)

DevOps

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.



In some DevOps models, quality assurance and security teams may also become more tightly integrated with development and operations and throughout the application lifecycle. When security is the focus of everyone on a DevOps team, this is sometimes referred to as DevSecOps.

Why choose DevOps

1) **Speed**

Move at high velocity so you can innovate for customers faster, adapt to changing markets better, and grow more efficiently at driving business results. The DevOps model enables your developers and operations teams to achieve these results. For example, microservices and continuous delivery let teams take ownership of services and then release updates to them quicker.

2) Rapid Delivery

Increase the frequency and pace of releases so you can innovate and improve your product faster. The quicker you can release new features and fix bugs, the faster you can respond to your customers' needs and build competitive advantage. Continuous integration and continuous delivery are practices that automate the software release process, from build to deploy.

3) **Reliability**

Ensure the quality of application updates and infrastructure changes so you can reliably deliver at a more rapid pace while maintaining a positive experience for end users. Use practices like continuous integration and continuous delivery to test that each change is functional and safe. Monitoring and logging practices help you stay informed of performance in real-time.

4**) Improved Collaboration**

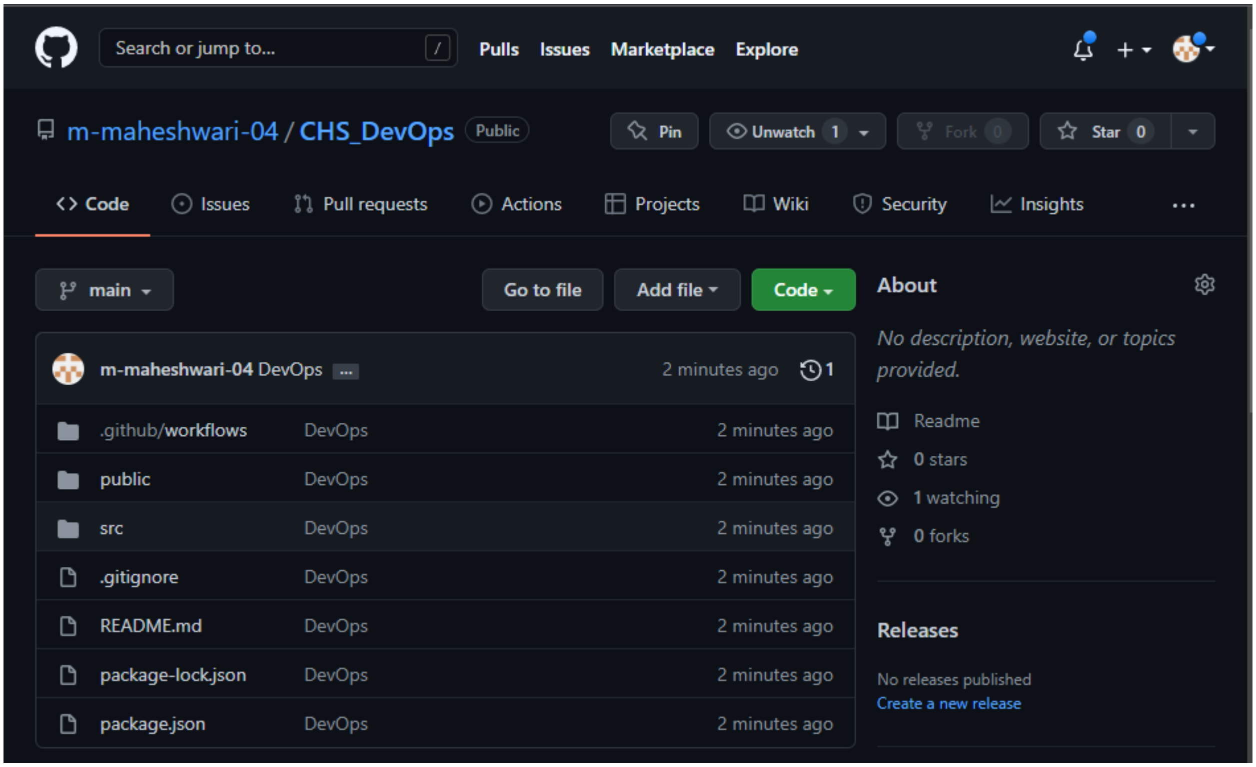
Build more effective teams under a DevOps cultural model, which emphasizes values such as ownership and accountability. Developers and operations teams collaborate closely, share many responsibilities, and combine their workflows. This reduces inefficiencies and saves time (e.g. reduced handover periods between developers and operations, writing code that takes into account the environment in which it is run).

5) **Security**
Move quickly while retaining control and preserving compliance. You can adopt a DevOps model without sacrificing security by using automated compliance policies, fine-grained controls, and configuration management techniques. For example, using infrastructure as code and policy as code, you can define and then track compliance at scale.

Devops Using Github actions

We have performed continuous integrity features using github actions.

1) We first need to make a project that needs continuous integration.



m-maheshwari-04 / CHS_DevOps

m-maheshwari-04 DevOps ... 2 minutes ago 1

.github/workflows	DevOps	2 minutes ago
public	DevOps	2 minutes ago
src	DevOps	2 minutes ago
.gitignore	DevOps	2 minutes ago
README.md	DevOps	2 minutes ago
package-lock.json	DevOps	2 minutes ago
package.json	DevOps	2 minutes ago

About

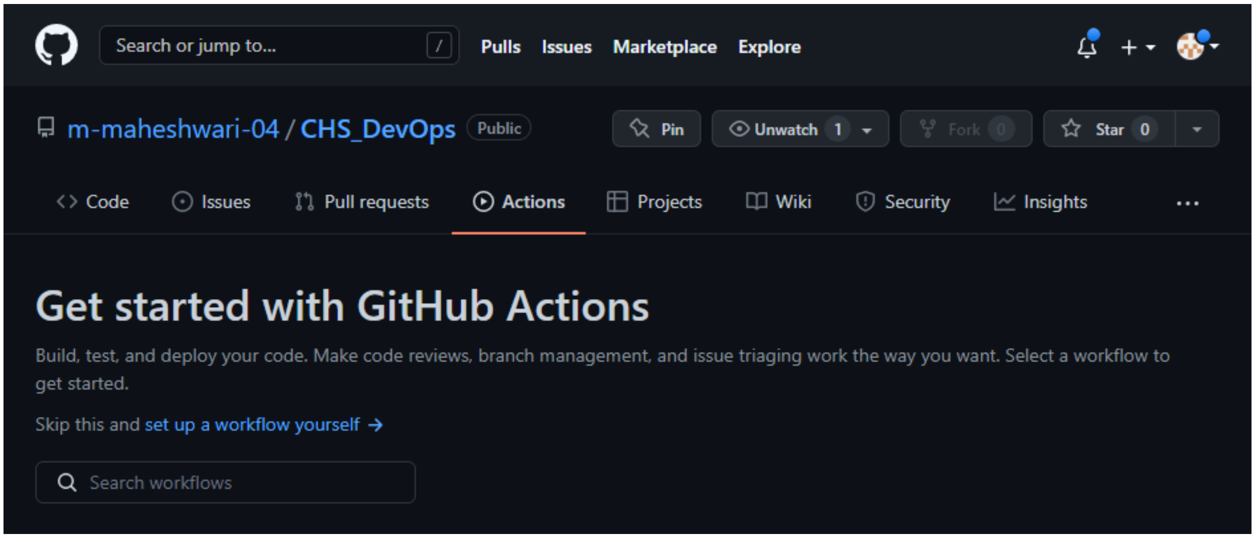
No description, website, or topics provided.

Readme
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

2) Setting up the workflow

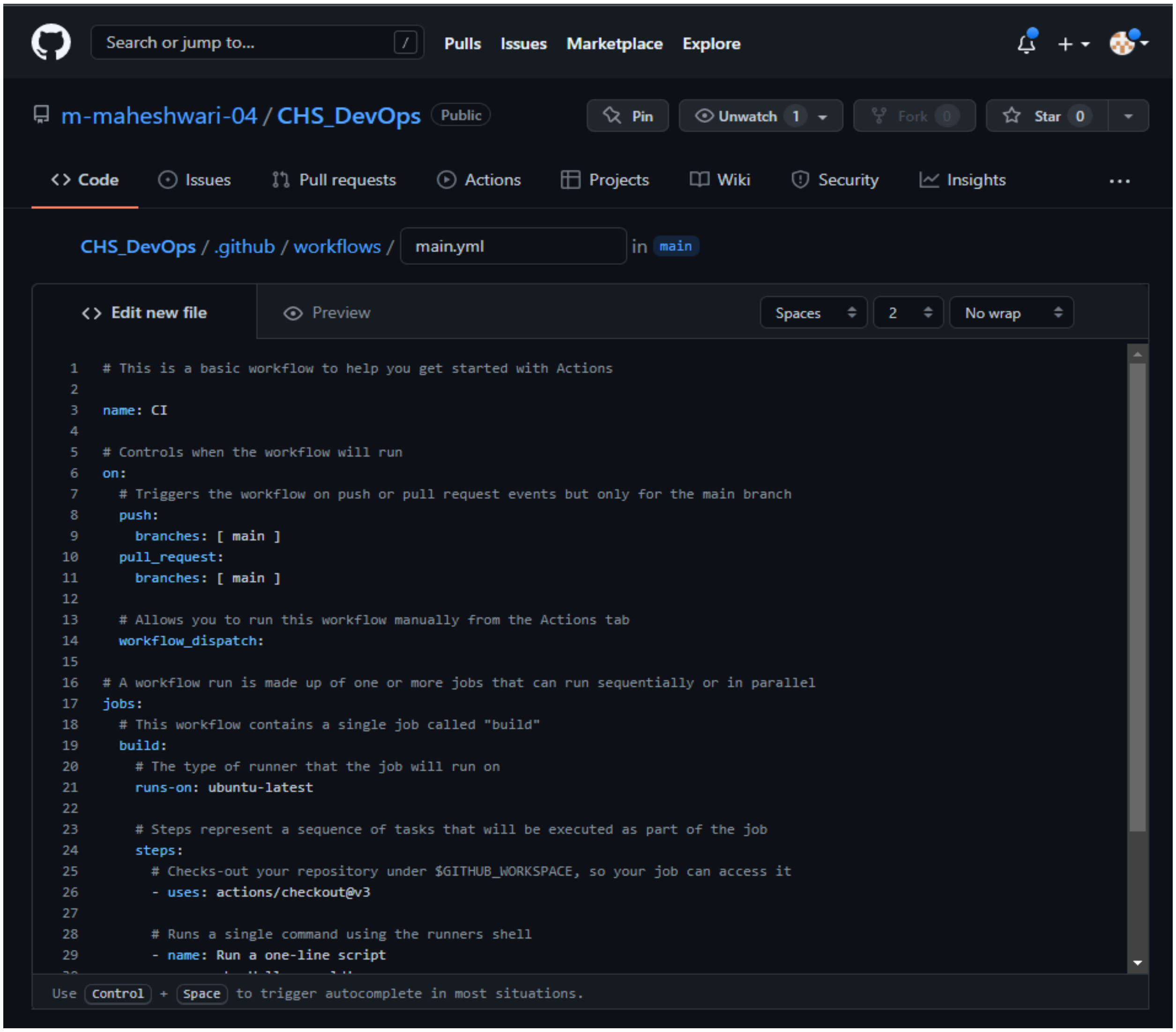


Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

Skip this and set up a workflow yourself →

3) Writing the main.yaml file



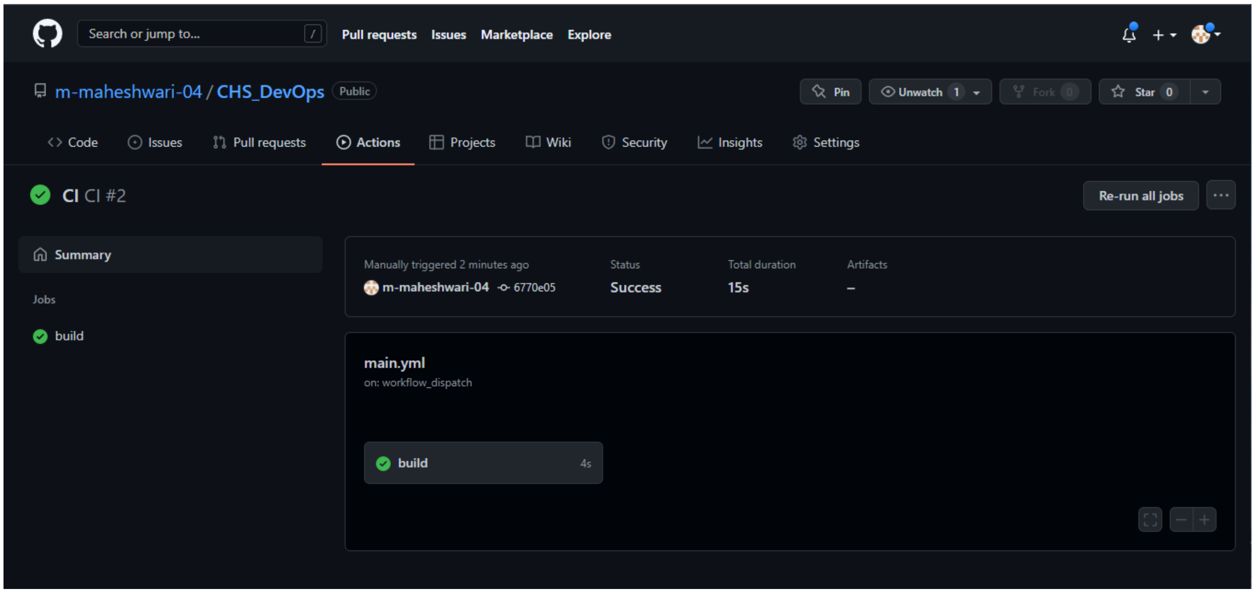
CHS_DevOps / .github / workflows / main.yml in main

Spaces 2 No wrap

```yaml
# This is a basic workflow to help you get started with Actions

name: CI

# Controls when the workflow will run
on:
  # Triggers the workflow on push or pull request events but only for the main branch
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]

  # Allows you to run this workflow manually from the Actions tab
  workflow_dispatch:

# A workflow run is made up of one or more jobs that can run sequentially or in parallel
jobs:
  # This workflow contains a single job called "build"
  build:
    # The type of runner that the job will run on
    runs-on: ubuntu-latest

    # Steps represent a sequence of tasks that will be executed as part of the job
    steps:
      # Checks-out your repository under $GITHUB_WORKSPACE, so your job can access it
      - uses: actions/checkout@v3

      # Runs a single command using the runners shell
      - name: Run a one-line script
```

Use Control + Space to trigger autocomplete in most situations.

4.) Checking status



CI CI #2

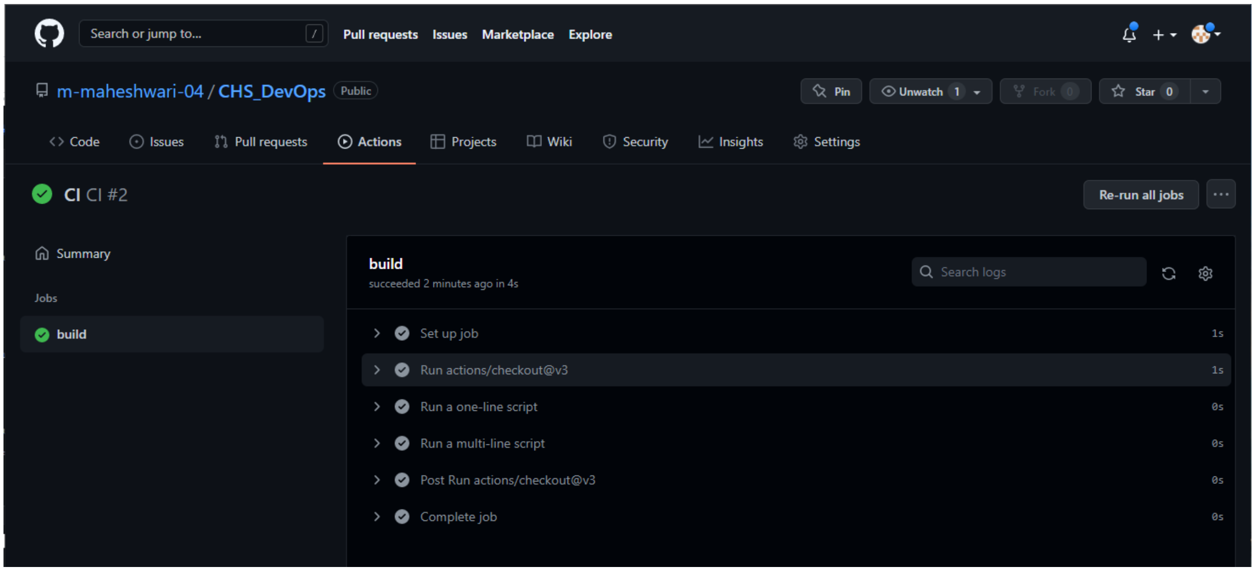
Manually triggered 2 minutes ago
m-maheshwari-04 -○- 6770e05

Status	Total duration	Artifacts
Success	15s	–

main.yml
on: workflow_dispatch

✓ build 4s

5.) Build status



CI CI #2

build
succeeded 2 minutes ago in 4s

```
> ✓ Set up job                     1s
> ✓ Run actions/checkout@v3        1s
> ✓ Run a one-line script          0s
> ✓ Run a multi-line script        0s
> ✓ Post Run actions/checkout@v3   0s
> ✓ Complete job                   0s
```