Analysis of cloud environment for implementing machine learning model comparative to the local server

MSc Research Project
Cloud Computing

Devaraj Devegowda
Student ID: 20125429

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Devaraj Devegowda

Student ID: 20125429

Programme: Msc cloud computing **Year:** 2022

Module: Msc Research project

Lecturer: Sean Heeney

Submission Due Date: 28/04/2022

Project Title: Analysis of cloud environment for building machine learning model comparative to the local server

Word Count: **4558** **Page Count: 21**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Devaraj Devegowda

Date: 26/04/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Analysis of cloud environment for implementing machine learning model comparative to the local server

Devaraj Devegowda
20125429

Abstract

Machine learning implementation is becoming very popular in all industries. In agriculture, machine learning is implemented for different problems. some problems are crop recommendation and yield prediction. This study shows the implementation of crop recommendation and yield prediction using the two best algorithms with high accuracy. This research also proves which is the best platform to build, train and deploy machine learning models. Machine learning models are developed in two different environments one local machine and the other AWS SageMaker. After the deployments, this study proves that using AWS SageMaker provides more benefits compared to local machine development. Performance testing for both environments is done using JMeter. Results are evaluated on different samples.

1 Introduction

Cloud computing is essential for building highly scalable and highly efficient applications. High-performance cloud computing resources are often underutilized. The technology stack provided by the cloud helps the industry to build and deploy the highly scalable application in the latest environment. Services like EC2, S3, Sage maker, and much more help in building high-performance applications. (Agavanakis *et al.*, 2019)

Nowadays in all industries cloud is becoming the dominant technology. In the recent study of CNCF (Cloud Native Computing Foundation) cloud-based projects have increased up to 200 percent from 2017. To build an effective machine learning model one of the most important factors is scientific expertise. Even though cloud and machine learning will not converge together, it will help all the machine learning applications to grow effectively.

This study shows the use of machine learning in the agriculture industry. One of the main problems in the agricultural industry is crop recommendation for the soil type and yield prediction according to rainfall and temperature. A lot of studies are done to solve these issues and machine learning is one of them. Using machine learning some of the problems can be fixed like crop recommendation, disease prediction, yield prediction, water controlling system, etc.

This research aims to build machine learning models for two major problems. One crop recommendation and second yield prediction. Crop recommendation is done based on soil characteristics like Nitrogen, ph, potassium, rainfall, humidity, and phosphorous. Yield prediction is based on area, rainfall, pesticide, and temperature. The below diagram shows machine learning applications using cloud services.

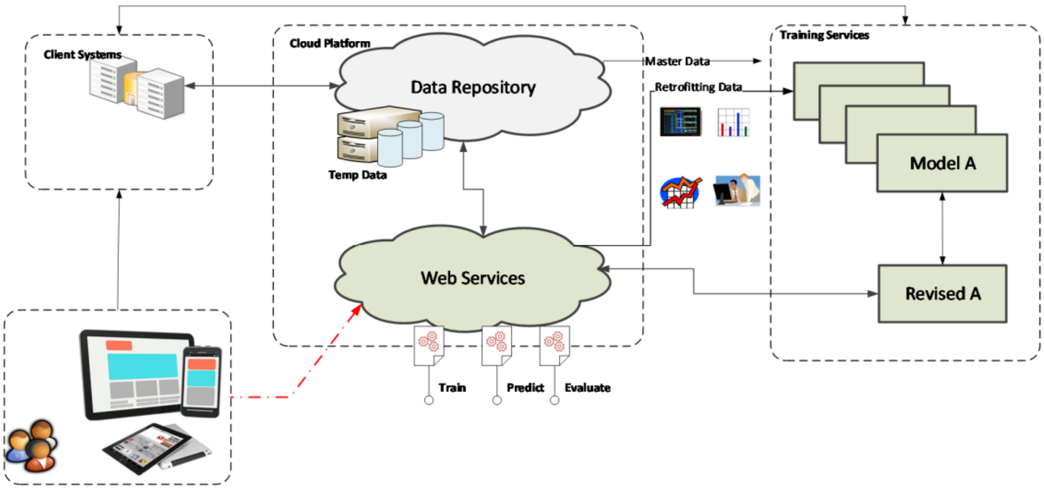

Figure 1: Machine learning with cloud services (Agavanakis *et al.*, 2019)

Two supervised learning algorithms are used, one is for classification and the other is for regression. For crop recommendation, a random forest algorithm is used and for yield, prediction used a decision tree regressor. Using these algorithms, a web-based application was built using python flask and deployed in AWS elastic beanstalk.

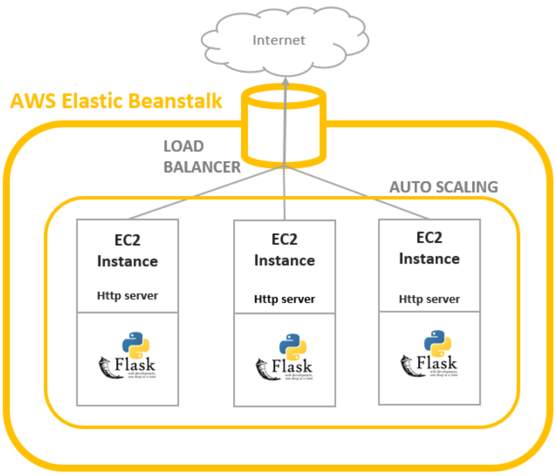

Figure 2: Deployment of ML models in Ec2 [1]

Same machine learning models are deployed in AWS SageMaker and created API using Lambda functions and API Gateway. This research also shows the benefits of using cloud services for building and deploying machine learning applications effectively. Later, try to compare the performance testing of both environments with different settings. In end, cloud see benefits of utilizing cloud services for machine learning-enabled applications.

[1] https://github.com/ScalarPy/AWS-Sagemaker-Deploy

1.1 Motivation

Machine learning is one of the most important aspects of all business. It is used in all types of industries including agriculture. Applications that are using machine learning can give users better search results and better recommendations based on their interests. Developing machine models and using them for different applications is a very big task.

Before creating any machine learning models one should be scientific expertise in their domain and should have gathered a lot of data to train the models. And after creating these models they must be implemented in real-world applications. Later these models are deployed in a local environment or virtual machine. Challenges in these environments are, that models can't be updated regularly, and storage for datasets keeps on expanding every day due to which models should learn accordingly. If the number of users increases latency and throughput of the applications are also reduced.

To overcome all these problems machine learning models should be built, trained, and deployed independently. This research shows the benefits of using cloud services. Cloud computing provides different services to make machine learning models run independently. Cloud also provides infinite storage for datasets. These model endpoints can be used in any application. Also because of auto-scaling options in the cloud latency and throughput of the application will increase. Along with it this study also shows, build, train and deploy two machine learning model one for crop recommendation and other yield prediction using AWS SageMaker.

1.2 Research Question

To what extent machine learning models can effectively implement using cloud services when compared to local servers?

Cloud computing is becoming more flexible for building and deploying machine learning applications. To build, train and deploy machine learning models in the local environment is very complex and lot of dependencies. This study shows how to build, train, and deploy machine learning models effectively using cloud services. The web application is developed and integrated with machine learning models. This application is deployed in an EC2 instance. This research also shows the results of performance testing running the machine learning models in SageMaker vs EC2. It concludes that SageMaker gives a better platform for Developing crop recommendation and yield prediction algorithms.

1.3 Structure of the Paper

The next part of the paper is designed as follows. Section 2 consists of related work on machine learning on crop recommendation and yield prediction, and cloud services like SageMaker, EC2, and S3. Section 3 gives an overview of the methodology for obtaining the intended results. Section 4 provides design specifications of the whole system. Section 5 provides an implementation description. At last sections 6 and 7 provide evaluation and conclusion respectively.

2 Related Work

Several research works are being carried out to emphasize the benefits of cloud services across all IT industries. Machine Learning is one such field that can leverage the services provided by cloud platforms like AWS for the deployment of these ML models that would indeed improve the performance, efficiency, and scalability of the system.

2.1 Machine Learning

Data is the new oil in current days and to make use of a large amount of available data everywhere, it needs to be analysed and useful information needs to be extracted. (Angra and Ahuja, 2017)states, that techniques like data mining and Machine Learning(ML) help to find the historical relationships within the data as well as train a system to learn these relationships and patterns to predict the future.

Machine Learning is applied across all industries including product recommendation, sales prediction, agriculture advisory, transportation network, demand prediction, Bots, anomaly detection and the list goes on. (Ray, 2019) in her recent work mentions that the types of problems that Machine learning techniques commonly deal with are 1. Regression 2. Classification and 3. Clustering. These problems can be solved by choosing one of the Machine Learning techniques of supervised learning, unsupervised learning, or semi-supervised learning.

(Ray, 2019)writes in her work, about the most widely used ML algorithms to solve issues in different fields. Linear Regression, Logistic Regression, and Multivariate Regression analysis are a few are the regression algorithms explained. Decision Tree, Support Vector Machine, Bayesian learning, and K-Nearest Neighbour are a few of the algorithms that could be used for classification type problems. On the other hand, Clustering algorithms like K-Means can be used for applications like document classification, customer segmentation, etc.

Decision Trees or Random Forest is one such algorithm within supervised ML technique, that works by selecting the best features and creating the best tree split point that well defines the data in question. Random Forest consists of a large number of Decision Trees where each of these unique trees classifies the data to a possible output class and the class with the most votes is the resulting class prediction. (Geetha *et al.*, 2020) tries to construct an ML model using the Random Forest algorithm for crop prediction, where she demonstrates Random Forest as the best algorithm among other ML algorithms. More the number of Decision Trees inside the Random Forest more the accuracy of the ML model.

2.2 Use Case: Crop Recommendation and yield prediction using Machine Learning

To help farmers, several machine learning models are being developed. (Doshi *et al.*, 2018)provided a crop recommendation system using machine learning. This article considers various environmental factors such as temperature, precipitation, area, and soil characteristics before recommending crops to users. Here, the authors use different classification algorithms, some are decision trees, KNNs, neural networks, and random forests to have high accuracy.

After applying different machine learning to this dataset, it is clear that the neural network provides the highest accuracy of 91%. As a result, the system shows the best use of soil data and rainfall predictions for crop recommendations.

Precision Agriculture, according to(Pudumalar *et al.*, 2017), delivers improved crop recommendations. Precision agriculture, per this study, takes use of soil data and area-specific factors. It is a modern farming technique that employs a variety of machine learning algorithms, including Nave Bayes, CHAID, Random Forest, and K-Nearest Neighbor. For a better result, this study incorporates site-specific factors. The data sets for this system were obtained from a Tamil Nadu soil testing lab. This system displays the Random Forest, with an accuracy of 88 percent according to CHAID.

(Pande *et al.*, 2021)offer a mobile application that allows users to enter input parameters and get crop recommendations. Machine learning algorithms here will recommend the most cost-effective crop to produce based on the soil type and are provided by the user. Artificial Neural Network, K-Nearest Neighbor, Random Forest, and Support Vector Machine are among the machine learning algorithms used by the author. The author employs two sets of data, one for soil type and the other for crop kind. For better results, these data sets are pre-processed. The Random Forest algorithm used here gives 95 percent accuracy on the provided data set, according to this system.

(Mishra, Mishra and Santra, 2016)have done similar work on understanding the applicability of machine learning in the agricultural sector. The goal of this study is to see if these advanced machine learning approaches have a meaningful link with their applications to the various attributes in the dataset. (Mishra, Mishra and Santra, 2016)outline the strategy and methodology behind each machine algorithm, such as the Markov chain model, Artificial Neural Network (ANN), Decision tree, Information Fuzzy Network, Regression analysis, and Bayesian analysis. It also gives a comprehensive analysis of which machine learning approaches are best for specific crops, such as cotton, corn, grains, sugar, soybean, olive, and jowar, and which ML techniques produce the best results, as well as other issues such as seasonal and inter-annual climate predictions.

2.3 Usage of Cloud services for deploying Machine Learning models.

(Agavanakis *et al.*, 2019) says the most important component in developing effective machine learning systems is scientific knowledge. Even though machine learning and cloud computing aren't inherently merging, the cloud platform is increasingly becoming a tool for ML applications. They designed a platform offered as a bouquet of cloud-hosted web services that make extensive use of PaaS and SaaS features like workflows, big data management, ML training models, etc. Leveraging the flexibility of the cloud resources it favours global accessibility, high availability, scalability, performance, and high-security standards. Besides being responsible for acquiring, modelling, training, and using the results either online or offline, it also provides automation mechanisms to connect diverse systems that may span the boundaries of the organization itself.

In this study, the practical aspects and capabilities of ML services have been explored, building a pragmatic cloud-based solution that provides efficient integration in a scalable infrastructure for shared knowledge.

Amazon SageMaker is a machine learning service that is fully managed. Data scientists and developers can use SageMaker to construct and train machine learning models fast and easily, then deploy them directly into a production-ready hosted environment. You don't have to manage servers because it has an integrated Jupyter writing notebook instance for easy access to your data sources for exploration and analysis. It also includes common machine learning methods that have been improved for use in a distributed setting with exceptionally huge data sets. SageMaker offers versatile distributed training alternatives that adapt to your individual workflows thanks to native support for bring-your-own-algorithms and frameworks. Launch a model from SageMaker Studio or the SageMaker console in a safe and scalable environment with just a few clicks.

From the above research, we can conclude it is seen that Crop recommendation and yield prediction are not done in the same application. This search uses a random forest and decision tree algorithm for crop recommendation and yield prediction in the same application. This research also provides python based web application to check the results of the trained model. This research also concludes the usage of cloud services for the development and deployment of machine learning models. In the end, the study shows that which is a better platform for running machine learning models.

3 Research Methodology

This section gives an overview of the methodologies used in this research. 3.1 provides information on process flow, and 3.2 provides information on what tools and technologies are used in this research. This study provides the two best machine learning models for crop recommendation and yield prediction. Using these machine learning models web applications also provided a better use case. This research also shows how to build, train, and deploy machine learning models in AWS SageMaker.

This research shows how to utilise all cloud services for developing Web applications and creating ML models without worrying about local environment setup. This study also shows deploying the machine learning models in two different environments one is Elastic Beanstalk and other SageMaker. The end of the research results shows which is a better platform for the development and deployment of machine learning models.

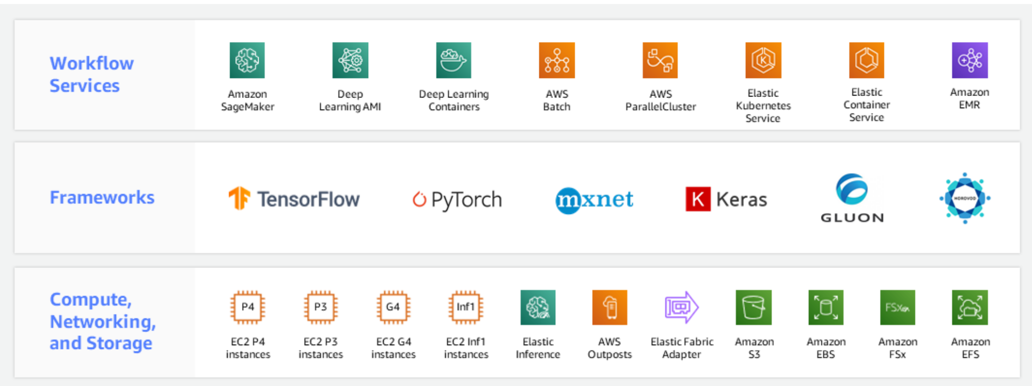

Figure 3: Aws infrastructure for Machine learning[2]

[2] https://aws.amazon.com/machine-learning/infrastructure

3.1 Process Flow of Research

To solve the crop recommendation problem Random Forest Classifier algorithm is used, and for yield prediction Decision Tree regressor is used. Created Web application using Python Flask framework for model testing. For deployment of web application AWS t2.micro instance is used. Two Datasets are used for developing the machine learning in this research. For crop recommendation, https://www.kaggle.com/atharvaingle/crop-recommendation-dataset consists of 2200 instances. For yield prediction https://www.kaggle.com/patelris/crop-yield-prediction-dataset again, contains 28242 instances.

For developing ML models in SageMaker created a notebook instance of ml.t2.medium and stored all the datasets in S3 storage. Created Lambda function for accessing SageMaker endpoint. This function helps in testing Deployed ML model. Created API Gateway for accessing the deploying the lambda function.

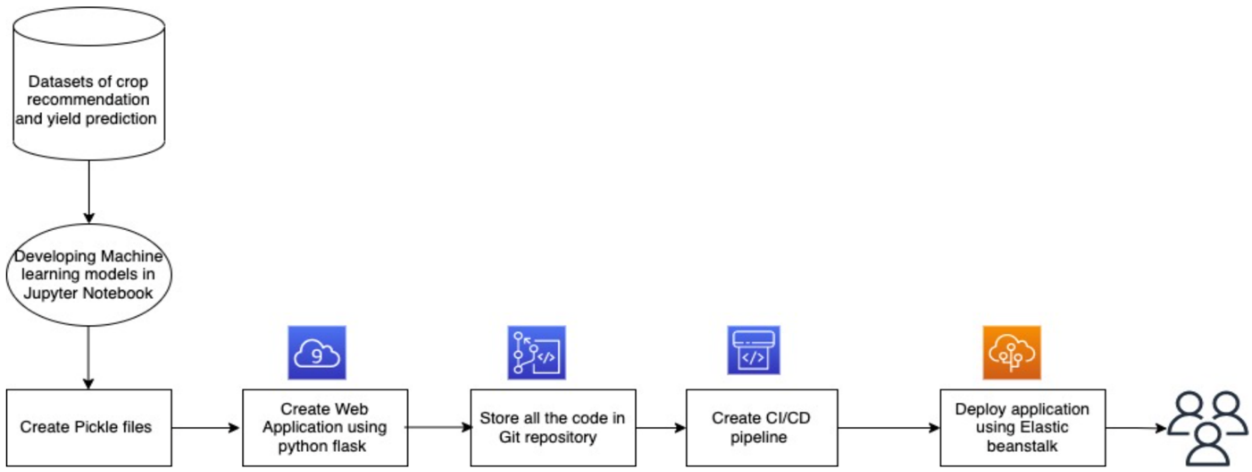

Figure 4: Process flow for Developing machine learning in the local environment

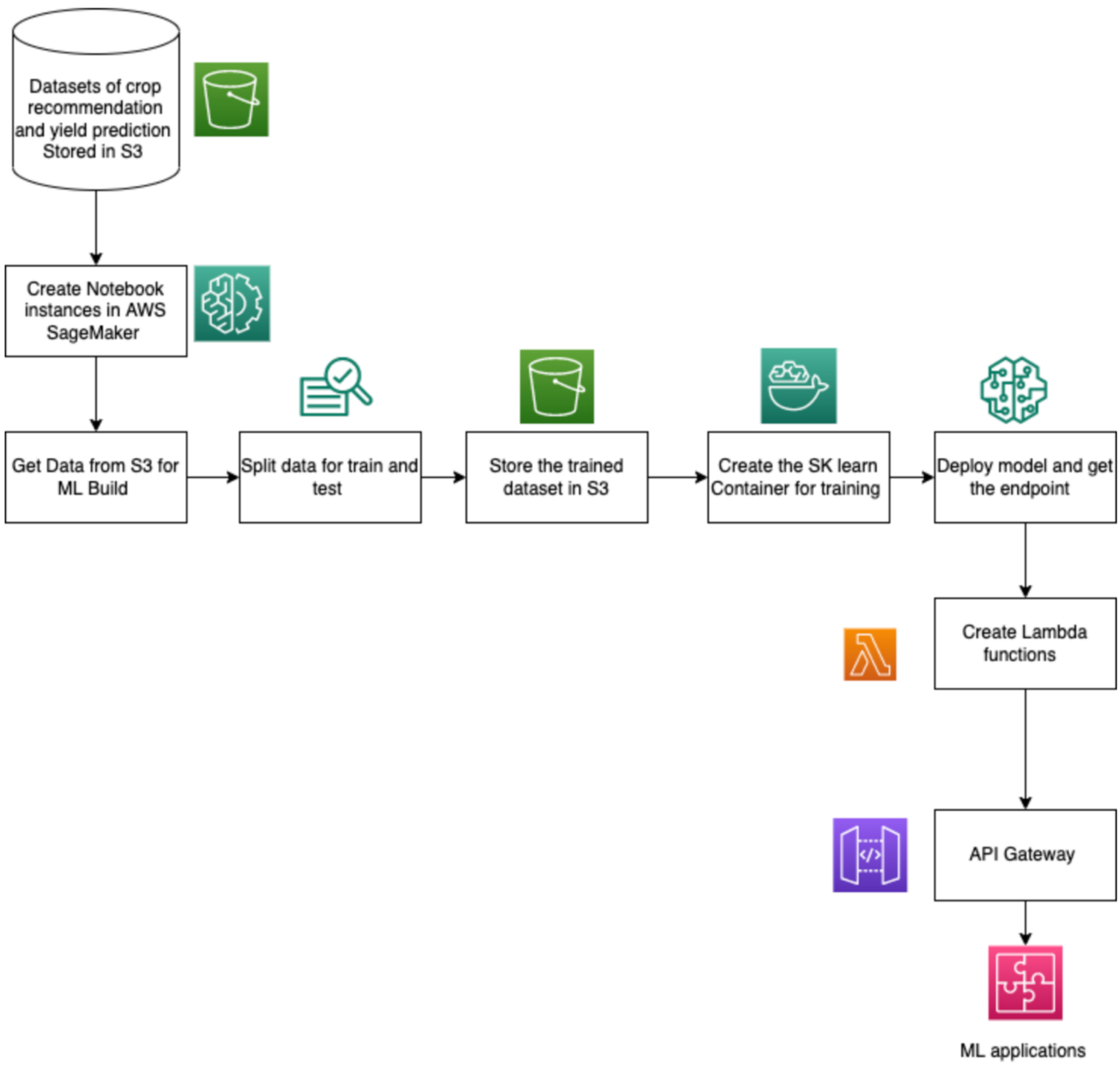

Figure 5: Process flow of Developing machine learning models in AWS Sage Maker

3.2 Tools and Technologies Used in Research

This research was conducted on AWS EC2 t2.micro and ml.t2.medium instances. Machine learning algorithms Random Forest and Decision Tree are built and deployed in AWS SageMaker. The web application is developed using the python flask framework.

The following tools and technologies are used in carrying out this research:

- Jupyter Notebook: In this research for developing and testing machine learning algorithms jupyter notebook was used as a workspace.
- Python: For building the web application and creating the machine learning models python was used as a primary language.
- Cloud 9: This research uses Cloud 9 IDE for writing code and used push the code for GitHub.
- GitHub: GitHub is used for storing and accessing the code.
- AWS SageMaker: This research makes use of SageMaker to build, train and deploy machine learning models.

- Elastic Beanstalk: To deploy a python web application with an auto-scaling option Elastic Beanstalk is used.
- S3: The Datasets used in this research are stored and accessed through the S3 bucket.
- Code pipeline: This research makes use of a Code pipeline for CI/CD implementation.
- Lambda Function: Lambda functions are used to create a function to test machine learning models.

3.3 Assessment Carried Out in This Research

For recommendation and prediction, this research used six different algorithms out that chosen two algorithms with the best accuracy. This study shows the best use of the random forest algorithm for crop recommendation and the Decision Tree for crop yield prediction.

These algorithms are used in python web applications for better usage. This application is deployed in the AWS cloud using Elastic Beanstalk.

Same algorithms are developed using AWS SageMaker and deployed. It has been examined machine learning models' development is good in the cloud or local environment. Validation is done using JMeter on both applications for the same model.

4 Design Specification

This section provides the overall design specification of this research. Specification of cloud systems is described in 4.1. Architectural design for the Local environment is described in 4.2. Architectural design for Cloud environment 4.3.

4.1 Specification of the system

As part of this research lot of Cloud services are used to build and deploy machine learning models. The web application is developed using the python flask framework. Machine learning models are trained in jupyter notebooks in the local environment and stored in pickle files. These pickle files are then integrated with a web application. To build an application cloud 9 online IDE is used. Cloud 9 IDE runs on EC2 t2.micro instance. To deploy the application Elastic Beanstalk is used. Elastic Beanstalk provides an automated process for deploying web applications and autoscaling options. It also stores all the logs in a cloud watch for future use. Elastic beanstalk also uses EC2 t2.micro instances. To automate the code deployment Code pipeline is used. Code pipeline provides CI/CD integration for the system.

Cloud services	**Use case**
Elastic beanstalk (t2.micro)	Deploy ML application
Cloud 9(t2.mciro)	Online IDE
Code pipeline	CI/CD
ELB	storage

Table 1: Cloud services of local machine learning environment

To build and deploy machines using cloud providers this research uses AWS SageMaker studio. SageMaker studio uses notebook instances for developing machine learning models.

This research uses ml.t2.medium instance for the notebook. All the data sets are stored in the S3 bucket. To train the machine learning models ml.m4.xlarge instance is used in SageMaker. SageMaker deploys the trained model ml.m4.xlarge instance and provides the endpoint. Lambda function and API gateway are used to access the machine learning endpoint.

Cloud services	**Use case**
SageMaker Notebook(ml.t2.medium)	Building and deploying ML models
S3	Data sets storage
Lambda functions and API Gateway	Creating API for ML model
Sklearn container	ML model training

Table 2: Cloud Services for machine learning in the cloud environment

4.2 Architecture for the local machine learning environment

The high-level architecture of crop recommendation and yield prediction ML models is shown in figure 6. Machine learning models are trained and built-in Jupytper notebook in local systems. After building these models are stored in a pickle file. Using cloud 9 online IDE web app development is done. After the development one feature or function code was pushed to the GitHub repository. AWS Code pipeline was integrated with GitHub and Elastic beanstalk to achieve CI/CD. Application is deployed in AWS Elastic beanstalk. After the deployment user can test the application by providing soil characteristics.

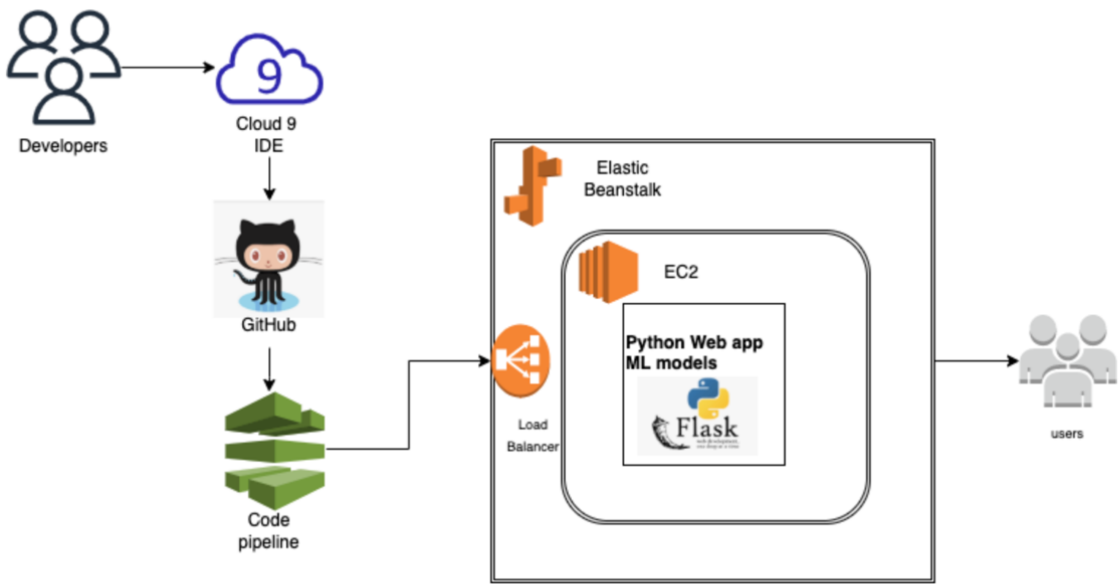

Figure 6: High-level architecture of the system

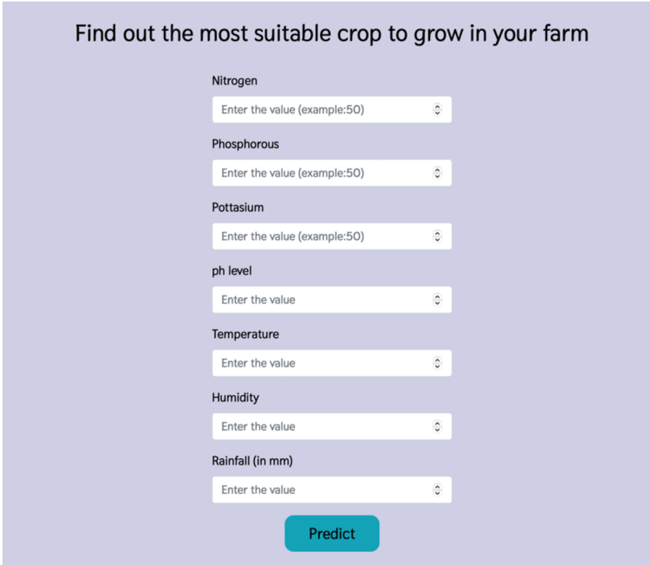

Figure 7: Form for Crop recommendation

4.3 Architecture for machine learning in a cloud-native environment

Crop recommendation and yield prediction models built using AWS SageMaker Figure 8 displays the high-level architecture of machine learning model implementation in the AWS cloud.

SageMaker is a fully cloud-native service that provides a platform for machine learning to run and scale. SageMaker helps the developers and data scientists to develop and deploy the machine models quickly for production-ready environments. It provides an inbuilt Jupytper notebook for easily accessing data sources and exploring them. SageMaker provides a platform for building its algorithm models from the Sklearn library.

In this research, SageMaker is used to build the random forest and decision tree machine learning models. These models are deployed and maintained in a secure environment. To access this model's lambda function and API Gateway is used.

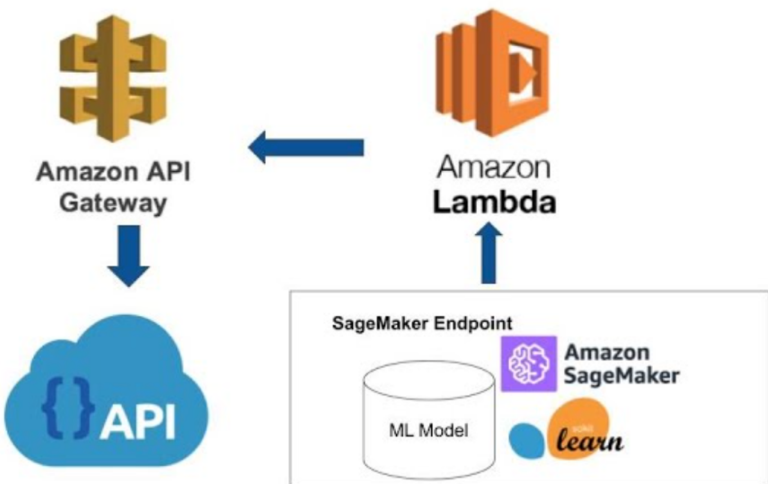

Figure 8: Machine learning models in SageMaker.

5 Implementation

The research implementation is discussed in this section. 5.1 describes the implementation of machine learning algorithms. In 5.2 it is explained the implementation of AWS SageMaker is.

5.1 Implementation of machine learning models

In this research, two main topics are crop recommendation and yield prediction. One is a classification problem, and the other is a regression problem.

5.1.1 Crop recommendation

The main goal of this topic is to provide insightful information for the farmers about what crops to grow before cultivation. The data sets for the algorithm are taken from open source Kaggle website. Datasets consist of Nitrogen, Phosphorous, Potassium, and pH values of the soil. It also contains the humidity, temperature, and rainfall required for a particular crop.

Data frame head

	N	P	K	temperature	humidity	ph	rainfall	label
0	90	42	43	20.879744	82.002744	6.502985	202.935536	rice
1	85	58	41	21.770462	80.319644	7.038096	226.655537	rice
2	60	55	44	23.004459	82.320763	7.840207	263.964248	rice
3	74	35	40	26.491096	80.158363	6.980401	242.864034	rice
4	78	42	42	20.130175	81.604873	7.628473	262.717340	rice

This is a classification problem to get the right output needs to find the right algorithm and train that. In this research, three algorithms are chosen to solve the problem. One Decision Tree, second Logistic Regression, and last random Forest.

This data set is divided into two parts train and test. Train datasets are 70 percent and test is 30 percent. After this data set is trained and tested with each algorithm.

Results are:

Decision Tree --> 0.8590909090909091
Logistic Regression --> 0.9621212121212122
RF --> 0.9893939393939394

It is shown that Random Forest is more accurate compared to the other two. Hence Random Forest is chosen for crop recommendation.

5.1.2 Crop yield prediction

Agricultural production is primarily influenced by meteorological conditions (rain, temperature, etc.), pesticides, and reliable information about crop yield history is critical for making agricultural risk management and future projections decisions. The basic components that keep us alive are the same. Corn, wheat, rice, and other basic crops make up a large part of our diet. This is a regression problem.

To get the right yield first it was considered using three different algorithms, Gradient Boosting Regressor, Random Forest Regressor, and Decision Tree Regressor.

After the data pre-processing below table shows how data sets look.

	Area	Item	Year	hg/ha_yield	average_rain_fall_mm_per_year	pesticides_tonnes	avg_temp
0	Albania	Maize	1990	36613	1485.0	121.0	16.37
1	Albania	Potatoes	1990	66667	1485.0	121.0	16.37
2	Albania	Rice, paddy	1990	23333	1485.0	121.0	16.37
3	Albania	Sorghum	1990	12500	1485.0	121.0	16.37
4	Albania	Soybeans	1990	7000	1485.0	121.0	16.37

The data set is divided into two parts one for training and the other for testing.

After running through all the algorithms, the Decision tree regressor provides more accuracy.

Results:
['GradientBoostingRegressor', 0.8965746504314166]
['RandomForestRegressor', 0.6842575852659857]
['DecisionTreeRegressor', 0.9591910674748293]

Hence decision tree regressor is chosen for crop yield prediction.

5.2 Implementation of a Web application in Python

After creating the machine learning right machine learning models web-based application is developed using python. The framework used for development is Flask. Machine learning models for crop recommendation and yield prediction are stored as pickle files in the application.

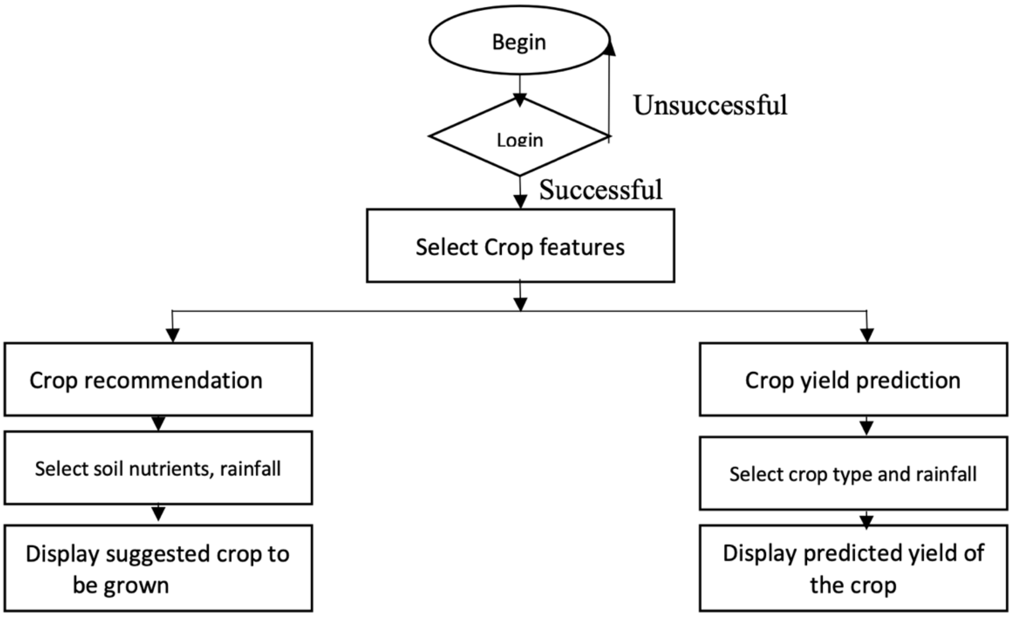

Figure 9: Flow diagram for web application

5.3 Machine learning Implementation using SageMaker

At first data sets are stored in S3 storage. S3 provides infinite storage and quick access for data within the AWS services. AWS SageMaker provides a scalable platform for building and developing machine learning models.

5.3.1 Model building

SageMaker provides a Jupyter notebook for building, training, and deploying ML models. Many of the users use these notebooks for data analysis and model building. SageMaker provides GPU or Big CPU for model building.

5.3.2 Model Training

Same notebooks are used for training the models and later store the model and files in S3.

The below code shows how to save the trained file in s3

Upload the data for training

```
train_input = sagemaker_session.upload_data("data")
```

```
train_input
```

```
's3://sagemaker-eu-west-1-796640696573/data'
```

To train the model Sklearn estimator is used with instance type ml.m4. large. The later fit method is used to train the model

Create SageMaker Scikit Estimator

```
[8]: from sagemaker.sklearn.estimator import SKLearn

script_path = 'startup_prediction.py'

sklearn = SKLearn(
    entry_point=script_path,
    instance_type="ml.m4.xlarge",
    framework_version="0.20.0",
    py_version="py3",
    role=role,
    use_spot_instances=True,
    max_run=300,
    max_wait=600)
    sagemaker_session=sagemaker_session)
```

Train SKLearn Estimator on Startup data

```
[9]: sklearn.fit({'train': train_input})
```

5.3.3 Models deploy

Even the small model which is developed locally must be deployed somewhere. Common issues are scalability and latency in response due to the number of requests increased. To overcome this SageMaker provides scalable ML deployment with the endpoint. We can choose what type of instance for deployment.

The below code is used for deployment

deployment = sklearn.deploy(initial_instance_count=1, instance_type="ml.m4.xlarge")

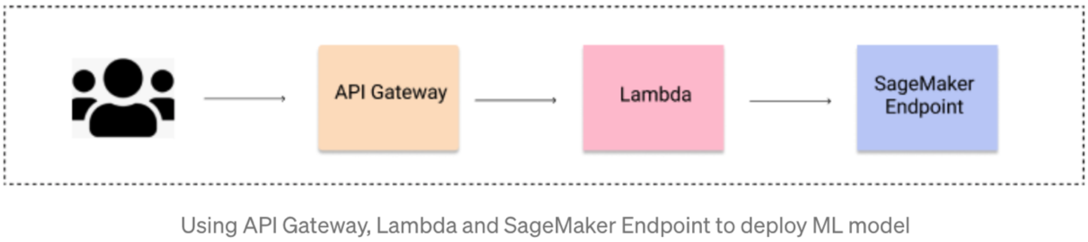

Figure 10: ML model deployment using SageMaker

6 Evaluation

Performance evaluation of ML model in Elastic Beanstalk and ML model in AWS SageMaker are done through JMeter test cases. In this research, three experiments are conducted. Each experiment tries to increase the number of requests at the same time. For this experiment, Elastic beanstalk with auto-scaling enabled has EC2 t2.micro instance. ML model deployed using SageMaker is ml.m4.xlarge.

Evaluation in this research has three experiments

First test ML application in both Elastic beanstalk and SageMaker with 10 requests at the same time.
Second, send 100 requests at the same time for both ML application
Third, send 500 requests at the same time for ML applications

Jmeter load testing is measured through average response time and throughput.

6.1 Experiment 1

In the first Experiment, 10 sample requests for both applications are sent. Then observe the average response time, error rate, and throughput on both applications. Results are shown below.

Applications	No of Samples	Average response time in milliseconds	Throughput	The error rate in percent
ML model in EC2	10	84	14.0/sec	0
ML model in SageMaker	10	299	13.3/sec	0

Table 3: Results of comparison after experiment 1

From the above experiment, we can observe that the ML model in EC2 is having less response time compared to the ML model in SageMaker.

6.2 Experiment 2

In this experiment, the number of samples is increased to 100.

Applications	No of Samples	Average response time in milliseconds	Throughput	The error rate in percent
ML model in EC2	100	808	49.0/sec	0
ML model in SageMaker	100	764	43.3sec	0

Table 4: Results of comparison after experiment 2

In this experiment, we can see that ML in EC2 has more throughput then compared to others. The error rate for both is the same zero.

6.3 Experiment 3

In this experiment, we can observe that number of samples is increased to 1000. Here we test the scalability of both systems.

Applications	No of Samples	Average response time in milliseconds	Throughput	The error rate in percent
ML model in EC2	1000	8971	13.30/sec	63.6
ML model in SageMaker	1000	18225	9.9/sec	31.8

Table 5: Results of comparison after experiment 3

From all the above experiments it is observed that the ML model in EC2 is getting more error rate compared to other. And ML in EC2 has less average response time.

6.4 Discussion

After conducting all three experiments for different settings, results are varied in every stage.

In this research, Machine learning models are built, trained, and deployed in different environments. First in EC2, built ML model in local machine and integrated with web application and deployed in EC2. Second, ML models are built, trained, and deployed in SageMaker with help of SageMaker Studio.

During the first experiment, it is observed that both systems are working perfectly without any error. The average response time for EC2 is less compared to ML in SageMaker. And throughput is also slightly more. So, in this experiment, both systems are working perfectly.

During the second experiment, both systems are working perfectly. No sample is increased to 100, both systems are showings results almost the same, there is no error in all requests, average response time is both are almost similar. Throughput is a little more in the EC2 system.

During the third experiment, here No of samples is increased to 1000. The error rate for both systems increased due to many requests at the same time.

The below graphs show the results of all the experiments conducted. Here n is a number of sample requests sent to the system.

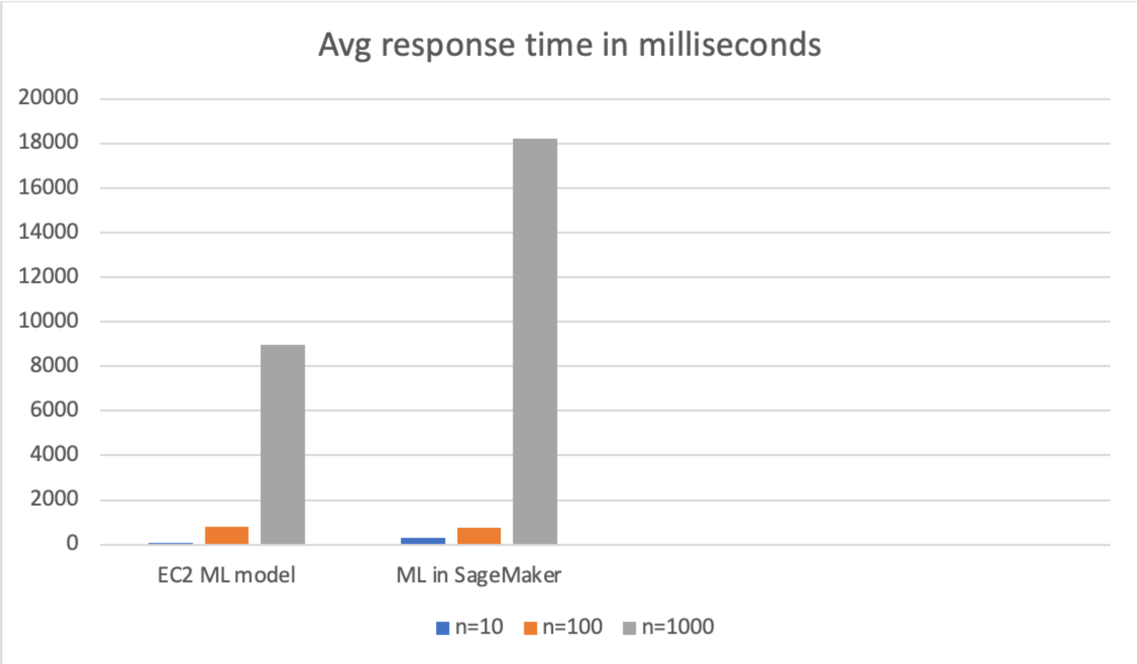

Figure 11: Average response time of both system

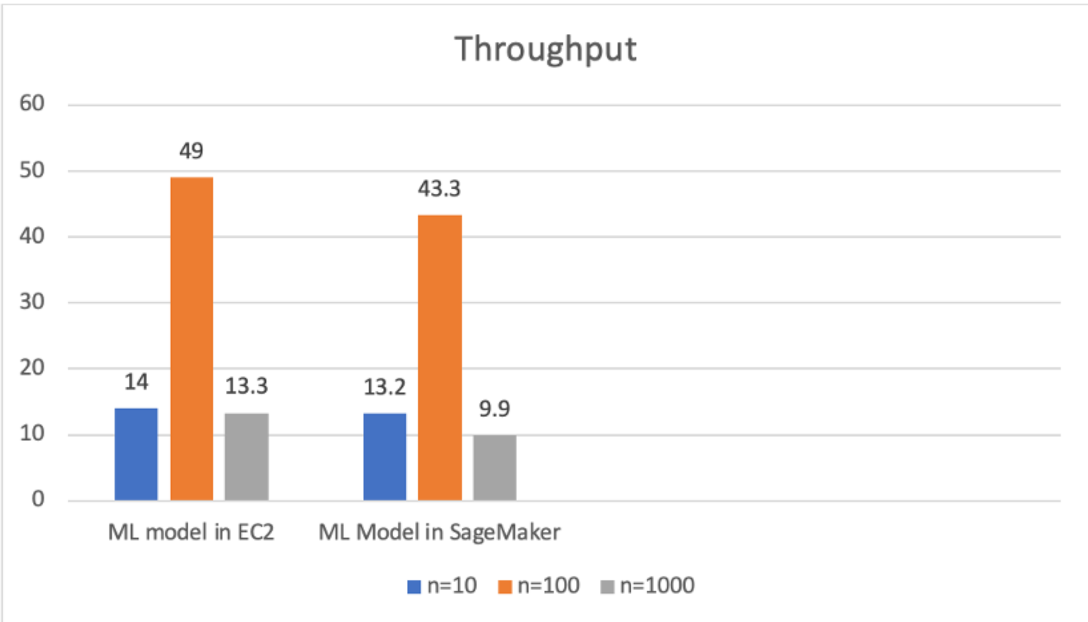

Figure 12: Throughput for both system

Figure 13: Error rate for both system

During experiments 1 and 2 there is a zero-error rate system that works fine. But in experiment 3 we can that as the load increases error rate also increases. EC2 ML model has more error rate compared to the SageMaker ML model.

7 Conclusion and Future Work

To conclude this research, crop recommendation and yield prediction are done using the two best algorithms. The research question analysed in this study is to compare the performance of an ML model in SageMaker and an ML model in EC2. Experiments include building, training, and deploying machine learning models in different AWS environments. This study shows using sage maker has a lot of benefits compared to ML in local servers.

SageMaker provides flexibility for developers and data scientist to run their workload without worrying about data and scalability. The limitation of this study is that scheduled training jobs are not done and SageMaker was not implemented for large datasets.

Since there are no scheduled jobs in SageMaker this could be looked into future. In addition, not all the algorithms are present in AWS SageMaker, it must be tested how effectively it works compared to inbuilt algorithms.

References

Agavanakis, K. *et al.* (2019) *Practical machine learning based on cloud computing resources*.

Angra, S. and Ahuja, S. (2017) “Machine learning and its applications: A review,” in *2017 International Conference on Big Data Analytics And Computational Intelligence (ICBDAC)*, pp. 57–60. doi:10.1109/ICBDACI.2017.8070809.

Doshi, Z. *et al.* (2018) “AgroConsultant: Intelligent Crop Recommendation System Using Machine Learning Algorithms,” in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1–6. doi:10.1109/ICCUBEA.2018.8697349.

Geetha, V. *et al.* (2020) “An Effective Crop Prediction Using Random Forest Algorithm,” in *2020 International Conference on System, Computation, Automation, and Networking (ICSCAN)*, pp. 1–5. doi:10.1109/ICSCAN49426.2020.9262311.

Mishra, S., Mishra, D. and Santra, G.H. (2016) “Applications of machine learning techniques in agricultural crop production: A review paper,” *Indian Journal of Science and Technology*, 9(38). doi:10.17485/it/2016/v9i38/95032.

Pande, S.M. *et al.* (2021) “Crop Recommender System Using Machine Learning Approach,” in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*. Institute of Electrical and Electronics Engineers Inc., pp. 1066–1071. doi:10.1109/ICCMC51019.2021.9418351.

Pudumalar, S. *et al.* (2017) “Crop recommendation system for precision agriculture,” in *2016 8th International Conference on Advanced Computing, ICoAC 2016*. Institute of Electrical and Electronics Engineers Inc., pp. 32–36. doi:10.1109/ICoAC.2017.7951740.

Ray, S. (2019) “A Quick Review of Machine Learning Algorithms,” in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 35–39. doi:10.1109/COMITCon.2019.8862451.