**School of Engineering and Applied Science (SEAS)**
**Ahmedabad University**

**SUBJECT: Algorithms and Optimization of Big Data (CSE511)**

**Project Report**

# WILD BLUEBERRY YIELD PREDICTION USING REINFORCEMENT LEARNING

**Submitted by : Team Narsimha**

| | |
|---|---|
| Akshay Bhimani | AU1841126 |
| Manav Patel | AU1841037 |
| Rahul Parmar | AU2044001 |
| Rushil Patel | AU1841008 |
| Shubh Shah | AU1841122 |

# Contents

# 1 INTRODUCTION

## 1.1 Problem Definition

Given a dataset consisting of a variety of honeybee characteristics and weather conditions as well as data on wild blueberries; create a **Reinforcement Learning** model that would be able to predict the yield of wild blueberries when new data (that was not present in the original dataset) is provided.

## 1.2 Problem Overview / Specifications

| | |
|---|---|
| How data was acquired | Generated from Simulation Modelling of Wild Blueberry Pollination by an open source GAMA simulation platform V1.7 (http://gama-platform.org), using GAML modelling programming language. |
| Data Format | XLSX, Raw |
| Parameters for Data Collection | 1) The Average size of Blueberry clones within a field;<br>2) Foraging Density of each bee taxon group; and<br>3) Weather information such as temperature, precipitation and wind speed. |
| Description of data collection | A total of 77,700 simulations were conducted to achieve both an extensive and intensive sampling effort and this resulted in a dataset consisting of 777 records, each of which is an average of 100 simulation runs. |
| Data Source Location | Institution : The University of Maine.<br>State/Region : Maine.<br>Country : USA. |
| Data Accessibility | Data was provided with [2]. |

Table 1: Specifications Table

| Features | Unit | Description |
|---|---|---|
| Clonesize | $m^2$ | The average Blueberry clone size in the field. |
| Honeybee | Bees/$m^2$/min | Honeybee density in the field. |
| Bumblebee | Bees/$m^2$/min | Bumblebee density in the field. |
| Andrena | Bees/$m^2$/min | Andrena bee density in the field. |
| Osmia | Bees/$m^2$/min | Osmia bee density in the field. |
| MaxOfUpperTRange | ℃ | The highest record of the upper band daily air temperature during the bloom season. |
| MinOfUpperTRange | ℃ | The lowest record of the upper band daily air temperature. |
| AverageOfUpperTRange | ℃ | The average of the upper band daily air temperature. |
| MaxOfLowerTRange | ℃ | The highest record of the lower band daily air temperature. |
| MinOfLowerTRange | ℃ | The lowest record of the lower band daily air temperature. |
| AverageOfLowerTRange | ℃ | The average of the lower band daily air temperature. |
| RainingDays | Day | The total number of days during the bloom season, each of which has precipitation larger than zero. |
| AverageRainingDays | Day | The average of raining days of the entire bloom season. |
| Fruitset | N/A | Fraction of flowers in an inflorence that actually form fruits. [7] |
| Fruitmass | g | Average mass of all blueberries [8]. |
| Seeds | N/A | Average number of seeds in wild blueberries. [1] |
| Yield | g/$m^2$ | The yield of wild blueberries [8]. |

Table 2: Features and their Description

With the help of the above given data features, the problem wants us to create a model that can predict the yield of blueberries, given a feature set (which was not already included in the data).

# 2 LITERATURE SURVEY / REVIEW

## 2.1 Previous approaches used in Agriculture

### 2.1.1 Simulation-based modeling of wild blueberry pollination. [1]

As the spatial and genetic characteristics of wild blueberry plants differ to a large extent, bee foraging behaviour as well as the complexity of interaction of these factors, are major obstacles to understanding the pollination wild blueberries. This paper present a spatially-explicit agent-based simulation model that enables exploration of how various factors, including spatial arrangements of plants, bee compositions and weather conditions interacts with each other, in isolation and combination, and how they affect pollination efficiency throughout a blueberry bloom season. Their analysis suggested that fruit set and measures resulting from it, like fruit mass and viable seeds per fruit were sensitive to the amount of blueberry plant cover in the field.

### 2.1.2 Wild blueberry yield prediction using machine learning algorithms. [2]

The most challenging task in the agriculture sector is to accurately predict crop yield. In this article they used data generated by the Wild Blueberry Pollination Model, and have attempted to reveal how bee species composition and weather affect yield and to predict optimal bee species composition and weather conditions that achieve the best yield using computer simulation and machine learning algorithms. Multiple Linear Regression (MLR), Boosted Decision Trees (BDT), Random Forest (RF), and Extreme Gradient Boosting (XGBoost) were evaluated as predictive tools.

The results showed that the Extreme Gradient Boosting (XGBoost) outperformed other algorithms in all measures of model performance for predicting the yield of wild blueberry. The results are consistent with previous work on predicting wild blueberry fruit yield and this study shows that crop yield predictions can be based on compter simulation modelling datasets.

### 2.1.3 Crop Yield Prediction Using Deep RL Model. [3]

Though deep learning models are broadly used to extract significant crop features for prediction, they are unable to create a direct non-linear or linear mapping between the raw data and crop yield values and the performance of these models highly relies on the quality of the extracted features.

Thus, using a combination of reinforcement learning and deep learning i.e, deep reinforcement learning, a complete crop yield prediction framework can be built that can map the raw data to the crop prediction values. A Deep Recurrent Q-Network model is proposed, i.e. a Recurrent Neural Network deep learning algorithm is constructed over the Q-Learning reinforcement learning algorithm which efficiently predicts the crop yield outperforming existing models.

### 2.1.4 Deep RL-Based Irrigation Scheduling. [4]

In this article, a deep reinforcement learning based approach is proposed that can automate the irrigation process that results in higher simulated net return. Using this approach, the irrigation controller can automatically determine the optimal or near-optimal water application amount considering soil moisture level, evapotranspiration, forecast precipitation, and crop growth stage.

Traditional reinforcement learning fails to accurately represent a real-world irrigation environment due to its limited state space. The deep reinforcement learning method can better model a real-world environment based on multi-dimensional observations. Simulations for various weather conditions and crop types show that the proposed deep reinforcement learning irrigation scheduling can increase net return.

### 2.1.5 Design And Implementation Of Crop Yield Prediction Model. [5]

As there isn't any franework in location to suggest farmers what plants to grow, this study proposes a machine learning approach that aims at predicting best yield crop for a particular region by analyzing various atmospheric factors like rainfall, temperature, humidity etc., as well as land factors like soil pH, using suitable machine learning algorithms must be applied to compute efficiency and capability of the model.

In the proposed model, random forest gives better yield prediction amongst random forest, polynomial regression and decision tree algorithms. The implementation of this system would help in better cultivation of the agricultural practices of our country.

#### 2.1.6 Reinforcement Learning Control for Agricultural Irrigation. [6]

In this work, a **Reinforcement Learning** based control technique is investigated to increase water-use efficiency in irrigation. Here, the reward of crop yield is handled by the temporal difference technique. Furthermore, the learning process they employed can be trained by both offline simulation and real data from the sensors. Finally, they found out that the method they proposed can **significantly** increase net return (considering both crop yield and water expense) and simulations for various geographic locations and crop types back that claim.

## 2.2 Importance of Crop Yield Prediction in today's world

Increasing the accuracy of agricultural forecasting is an important application of earth observation.

The influence of climate change and its unpredictability, has caused majority of the agricultural crops to be affected in terms of their production and maintenance. Forecasting or predicting the crop yield well ahead of its harvest time would assist farmers for taking suitable measures for selling and storage. Accurate prediction of crop development stages plays an important role in crop production management. Such predictions will also support the allied industries for strategizing the logistics of their business. [9]

The average crop yield per acre also serves as the evaluation of a farmer's agricultural output on a particular field over a specified time period. It is considered to be probably the most important measure of each farmer's performance, as it embodies the result of all the efforts and resources invested by agrarians in the development of plants on their fields. This is the reason why most farmers find themselves in a constant quest called "How to increase the average crop yield per acre?". [10]

Crop yield prediction is of great importance to global food production. Policy makers rely on accurate predictions to make timely import and export decisions to strengthen national food security. Seed companies need to predict the performances of new hybrids in various environments to breed for better varieties. [11]

# 3 SYSTEM ANALYSIS & DESIGN

## 3.1 Proposed System

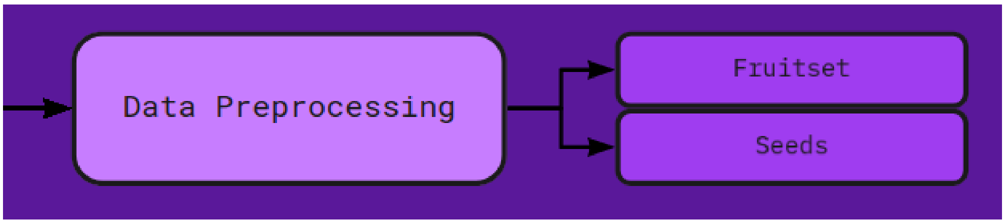
### 3.1.1 Attributes of the Model

**Environment** is a simulation/task/scenario that an agent interacts with. In our case, as we are trying to predict the yield of wild blueberries, hence, the provided wild blueberry data-set plays the role of environment.

**State** refers to the current situation returned by the environment. For our model, we have chosen the ’Seeds’ column as our States.

**Action** is something an agent does to change states and maximize reward. For our model, we have chosen the ’Fruit Set’ column as our Actions.

### 3.1.2 Data Preprocessing



In order to make the dataset more useful, we have to perform some pre-processing tasks. The data includes 16 independent variables and in a dataset. There may be features that are not completely relevant and thus not explanatory of blueberry yield. The contribution of these types of features is often low for predictive modelling compared to the most significant features obtained as the result of feature selection.

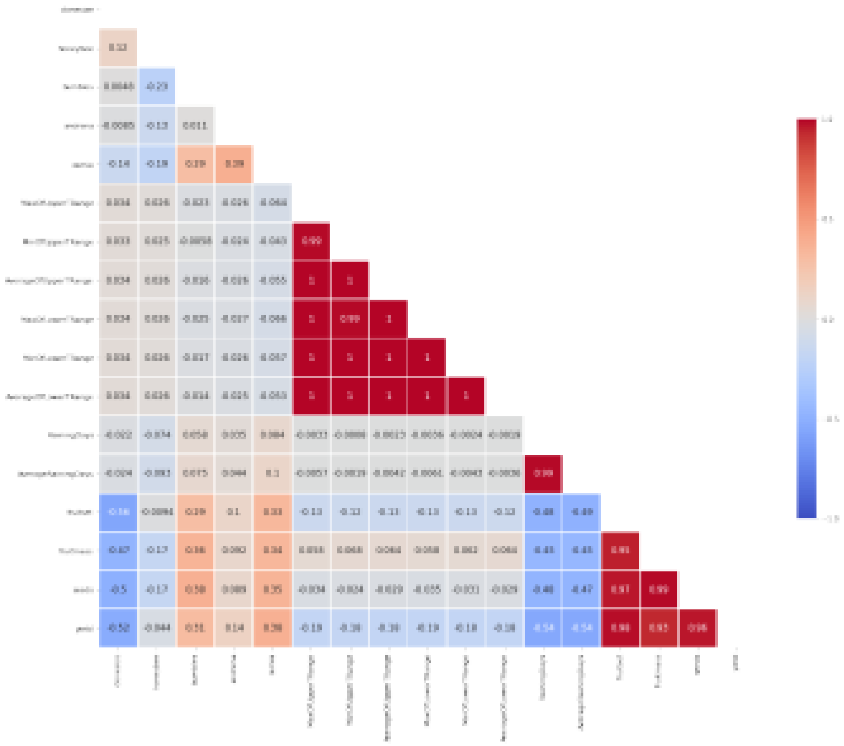
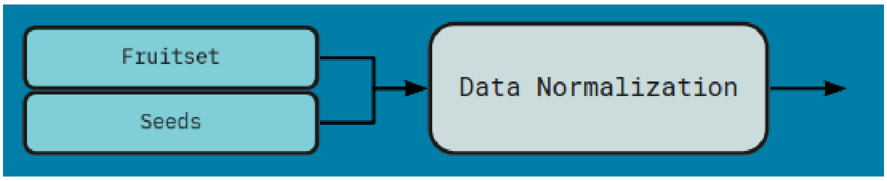


Figure 1: Correlation Matrix of our feature set

Hence, in order to disregard unnecessary features, we constructed a correlation matrix (shown above) with the help of Python's *numpy* and *pandas* library. We can clearly see that most of the features are weakly correlated, whereas *fruitset* and *seed* features are strongly correlated with the goal of our model, i.e. yield of wild blueberries. As a result, the weakly correlated features were ignored and the strongly correlated features were later applied to train and build the Reinforcement Learning model.

### 3.1.3 Data Normalization



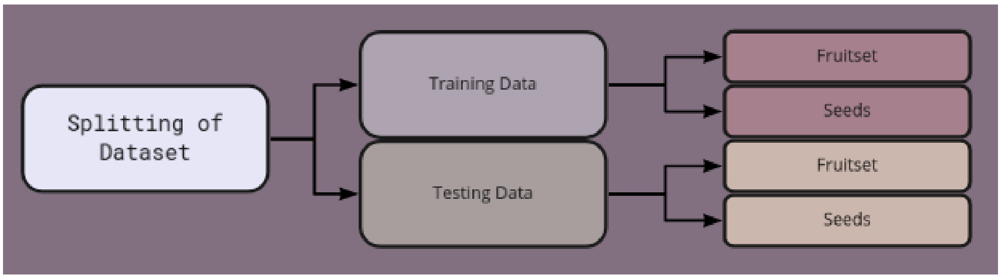
When the algorithm compares features that are on different scales, the feature on the larger scale may dominate the ones on lower scales completely. Hence we use normalization to ensure that every data point has the same scale and that each feature is equally important.

Here, we've used one of the most common methods to normalize data, the min-max normalization. For every feature, the minimum value gets transformed into **0**; similarly the maximum value gets transformed into **1** and every other value gets transformed into a value between **0** and **1**. The formula is as follows:

$$\frac{value - min}{max - min}$$

In our data set we have chosen 'Seeds' column as our state and normalized it by dividing all values (of 'Seeds') by the maximum seed value. Also, for classification part in DNN-1 we have normalized the action (here 'Fruitset') in a discrete fashion.

### 3.1.4 Training and Testing dataset generation



Since we usually need unbiased evaluation to properly assess the predictive performance of your model and validate the model, we can't evaluate the predictive performance of a model with the same data that was used for training the model. *Fresh data* is needed to evaluate the model, and this can be achieved by splitting the model before it is used.

**The training set** is applied to train, or *fit*, the model.

**The test set** is needed for unbiased evaluation of the final model, and should not be used for fitting or validation of the model. [12]

### 3.1.5 DNN-1 : Classification of Fruit Set

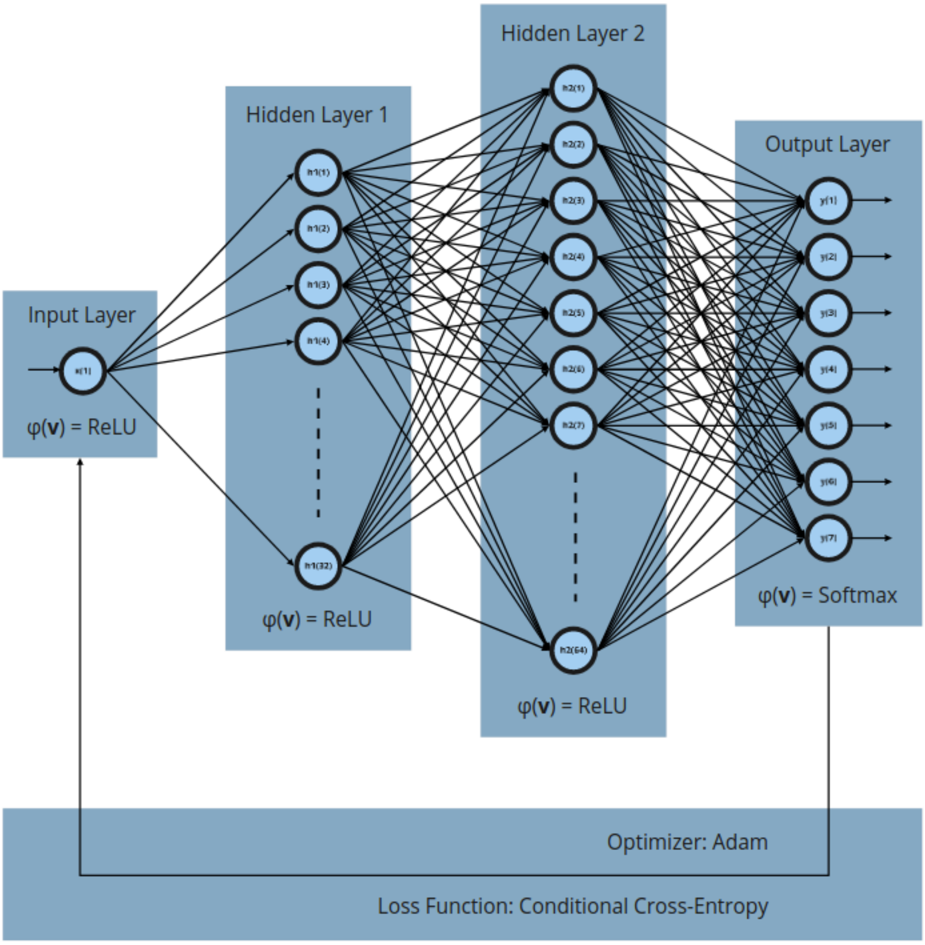


Figure 2: Architecture (visualization) of Deep-Neural-Network-1

| Layer | Neurons | Activation Function | Optimizer | Loss Function |
|---|---|---|---|---|
| Input | 1 | ReLU | Adam | Categorical Cross-Entropy |
| Hidden-1 | 32 | ReLU | | |
| Hidden-2 | 64 | ReLU | | |
| Output | 7 | Softmax | | |

Table 3: Architecture (summary) of Deep-Neural-Network-1

This deep neural network is a classification model that will determine the fruit set of blueberries (action) on the basis of average number of seeds (state); in other words, it maps states of our environment (seeds), to actions of our environment (fruit set).

**Activation Function(s) :**

**ReLU** is defined as:

$$f(x) = \begin{cases} x & \text{, if } x > 0 \\ 0 & \text{, otherwise} \end{cases}$$

where x is the input to the neuron.

Why use ReLU? Because, in 2011, it was found to enable better training of deeper networks, compared to the widely used activation functions prior to 2011. Also, ReLU offers many advantages like sparse activation, better gradient propagation (fewer vanishing gradient problems compared to sigmoidal activation functions), efficient computation, and it is scale invariant.

**Softmax** function takes as input a vector z of K real numbers, and normalizes it into a probability distribution consisting of K probabilities proportional to the exponential of the input numbers.

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{ for } i = 1, \cdots, K \text{ and } \mathbf{z} = (z_1, \cdots, z_K) \in \mathbb{R}^K$$

In other words, prior to applying softmax, some vector components could be negative, or greater than one; and might not sum to 1; but after applying softmax, each component will be in the interval $(0, 1)$ and the components will add up to 1, so that they can be interpreted as probabilities and hence, as we are trying to model a classification problem, use Softmax activation function.

**Optimizer :**

**Why use ADAM?** Because it is computationally efficient and combines the advantages of two other extensions of stochastic gradient descent, namely, *Adaptive Gradient ALgorithm (Adagrad)* and *Root Mean Square Propagation (RMSProp)*

**Loss Function :**

**Why use Categorical Cross-Entropy?** Because this is a classification model and Cross-entropy loss measures the performance of a classification model, it's natural that we use this loss function.

### 3.1.6 DNN-2 : Prediction of Yield

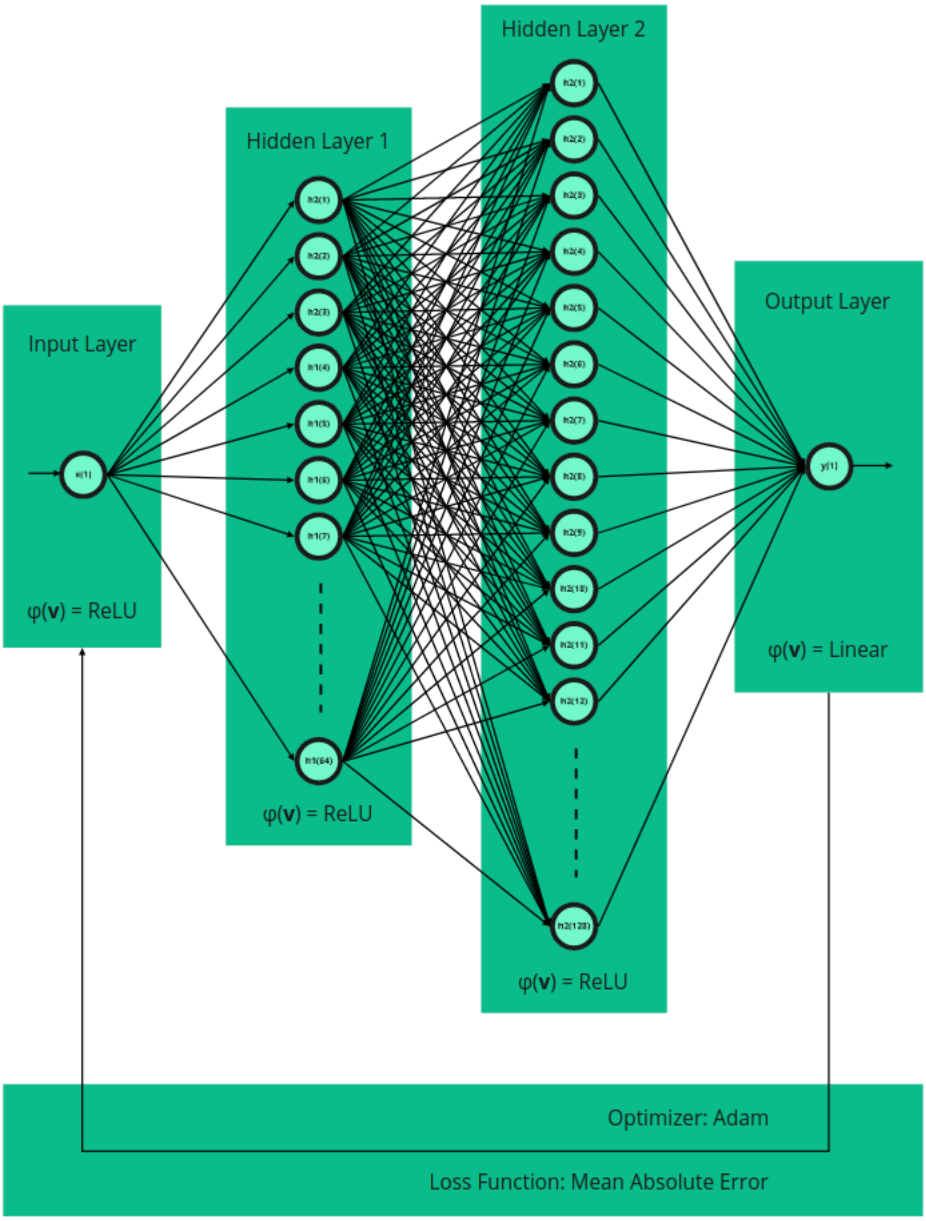


Figure 3: Architecture (visualization) of Deep-Neural-Network-2

| Layer | Neurons | Activation Function | Optimizer | Loss Function |
|---|---|---|---|---|
| Input | 1 | ReLU | Adam | Mean Absolute Error |
| Hidden-1 | 64 | ReLU | | |
| Hidden-2 | 128 | ReLU | | |
| Output | 1 | Linear | | |

Table 4: Architecture (summary) of Deep-Neural-Network-2

This deep neural network is a regression model that will predict the yield of blueberries (goal) on the basis of fruit set (action); which was obtained from the previous classification deep neural network.

**Activation Function(s) :**

**ReLU** is defined as:

$$f(x) = \begin{cases} x & , \text{if } x > 0 \\ 0 & , \text{otherwise} \end{cases}$$

where x is the input to the neuron.

Why use ReLU? Because, in 2011, it was found to enable better training of deeper networks, compared to the widely used activation functions prior to 2011. Also, ReLU offers many advantages like sparse activation, better gradient propagation (fewer vanishing gradient problems compared to sigmoidal activation functions), efficient computation, and it is scale invariant.

**Linear** is commonly used when predicting values using a regression model. Therefore, we used a linear activation model here.

**Optimizer :**

**Why use ADAM?** Because it is computationally efficient and combines the advantages of two other extensions of stochastic gradient descent, namely, *Adaptive Gradient ALgorithm (Adagrad)* and *Root Mean Square Propagation (RMSProp)*

**Loss Function :**

**Mean Absolute Error (MAE)** is another loss function used for regression models. MAE is the sum of absolute differences between our target and predicted variables.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - y_i^p|}{n}$$

Advantages of MAE:

- It is a linear measure its meaning is more intuitive.
- It is useful if the training data is corrupted with outliers.

### 3.1.7 Working of the system

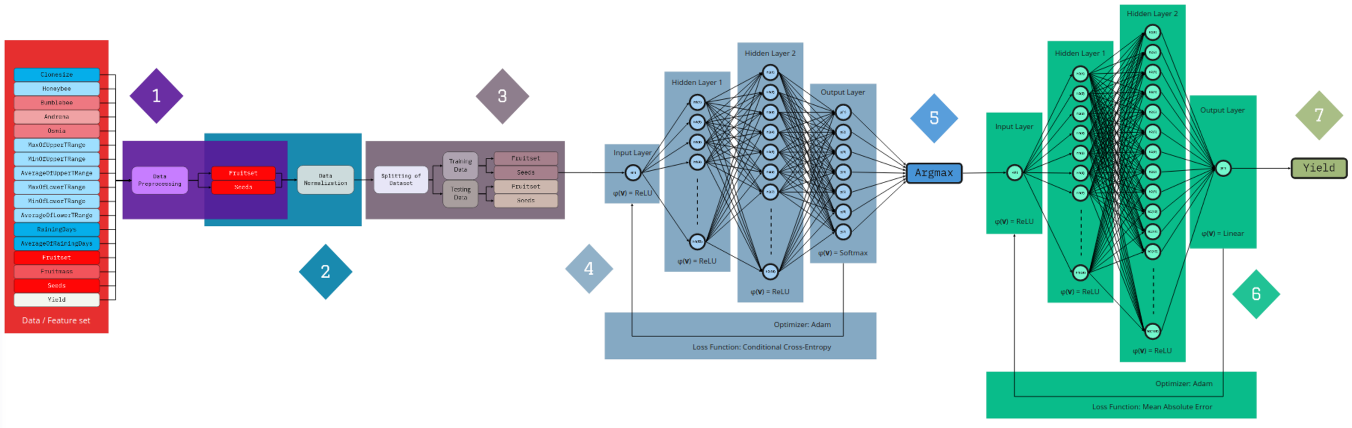
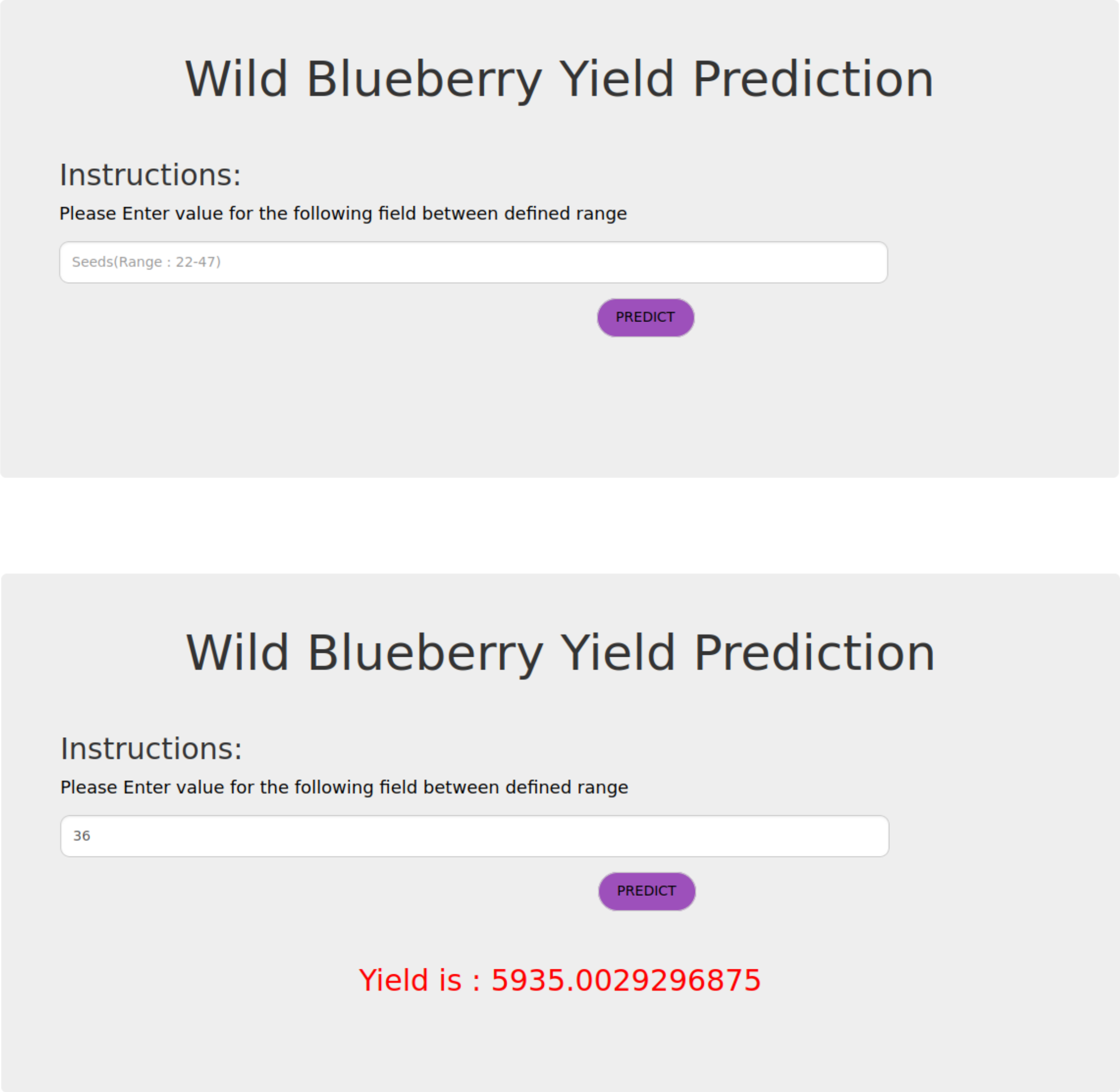


Figure 4: Working of the entire system

1. First of all, take the dataset and preprocess it to discard features that are weakly correlated with yield and keep those features that are strongly correlated with it.

2. After the strongly correlated features are extracted from the dataset, normalize those features (with min-max normalization).
3. When features have been normalized, split the dataset into training and testing datasets (both will have different instances / rows of same features).
4. Now, from the training dataset, pass the state of our environment (seeds) in classification deep neural network to obtain the probabilities of each action (fruit set).
5. Afterwards, pass these probabilities into the *argmax* function to obtain the action (fruit set) having the largest probability (of corresponding to that particular state).
6. Then, pass the output of *argmax* function (fruit set) into the regression deep neural network to obtain the predicted value of the yield of wild blueberries.
7. In this way we can predict the yield of wild blueberries based on state (seeds) and action (fruit set) of the environment (dataset).

### 3.1.8 Results



Wild Blueberry Yield Prediction

Instructions:

Please Enter value for the following field between defined range

Seeds(Range : 22-47)

Wild Blueberry Yield Prediction

Instructions:

Please Enter value for the following field between defined range

36

Yield is : 5935.0029296875

Figure 5: Interface of app deployed on Heroku

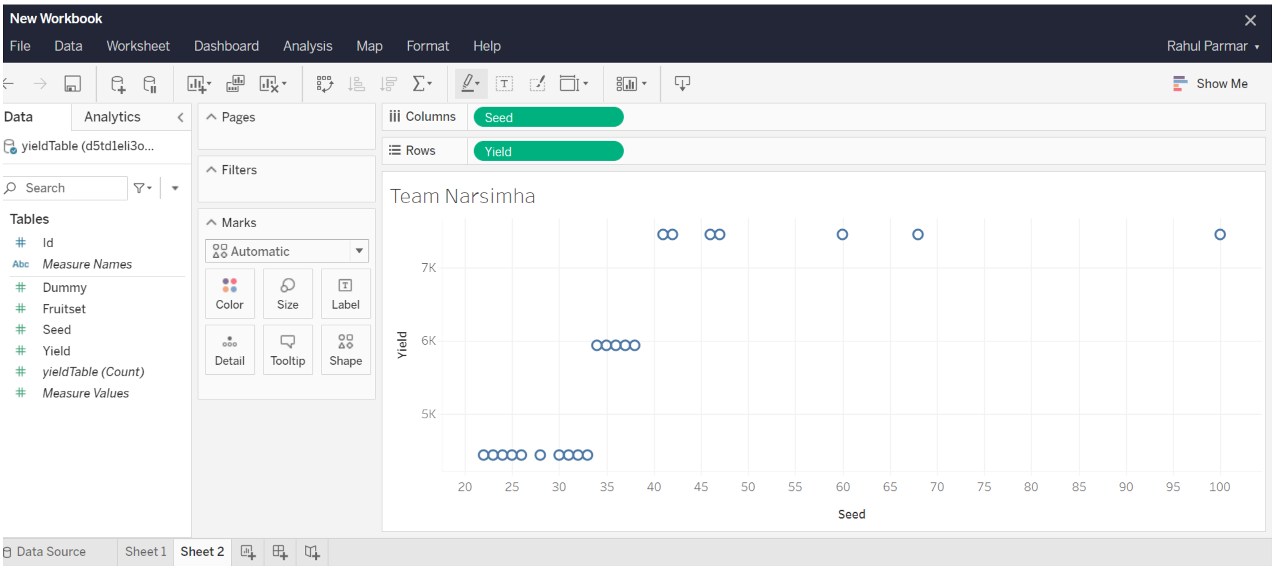
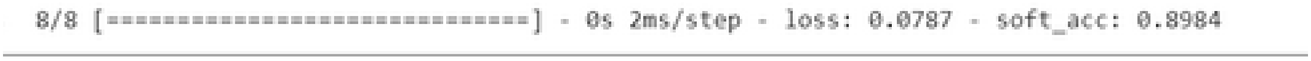


Figure 6: Visualization of Data in Tableau

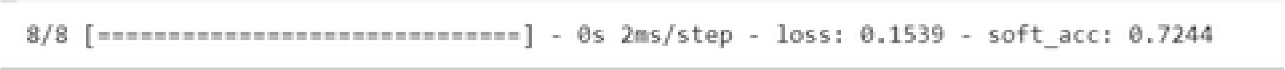
## 3.2 Testing the viability of our model

In order to test if the model is viable, we randomize the order of actions (fruit set) and then feed these actions into the second (regression) deep neural network and observe the accuracy of the model (this is in contrast to the previous approach where successive actions are determined on the basis of how previous actions performed).



```
8/8 [==============================] - 0s 2ms/step - loss: 0.0787 - soft_acc: 0.8984
```

Figure 7: Accuracy and loss of the original approach



```
8/8 [==============================] - 0s 2ms/step - loss: 0.1539 - soft_acc: 0.7244
```

Figure 8: Accuracy and loss of the ”randomized” approach

Evidently, the ”randomized” approach results in far worse loss (which doubles in this case) and accuracy (which decreases as well) than the loss and accuracy obtained by the original approach. Hence, we can confidently say that the strategy we adopted works better and makes our model more viable than randomly plugging actions into the deep neural network for prediction.

# 4 CONCLUSION

In this project we used various reinforcement learning concepts in order to create a prediction model that could predict the yield of wild blueberries with the help of data provided. Intuitively, we chose actions as fruitset, states as seeds and environment as the entire dataset. The model is divided into three parts.

The first part deals with the preprocessing, normalization and splitting of the dataset. It would be unwise to use the dataset without any ”cleaning”, as most of the features provided do not correlate strongly with the yield.

Next, the second part is nothing but a classification deep neural network that deals with the classification of action based on the states provided as input. The model takes advantage of the fact that actions can be roughly classified into a finite number of classes, which implies, every state's corresponding action must lie in one of these classes. After obtaining probabilities that a particular class of action corresponds to the given state, we take the action having the maximum probability and pass it ahead to the third part.

Finally, the third part also a deep neural network, which performs regression in order to predict the yield based on the actions provided as input. In this way, we predict the yield of wild blueberries from the states and actions presented. Although the provided data-set was relatively small, the model could predicted the blueberry yield with good accuracy compared to other models that perform prediction on the with the help of humongous amounts of data.

Apart from this, not only have we deployed this model on **Heroku** (for a live demo, please visit `https://blueberry-yield1.herokuapp.com/`), but also stored it's responses in a **PostgreSQL** database (also on **Heroku**) as well as interactively visualized that data in **Tableau Online**. In this way, we attempted to create an end-to-end system using services offered by **Heroku** and **Tableau**.

# 5 WORK DISTRIBUTION

| Task | Members |
|---|---|
| Model creation | Manav & Rahul |
| Testing | |
| Development | Akshay |
| Deployment | |
| Illustrations | Rushil & Shubh |
| Report creation | |

Table 5: Work Distribution Table

# References

[1] Hongchun Qu and Frank Drummond. Simulation-based modeling of wild blueberry pollination. *Computers and Electronics in Agriculture*, 144:94–101, 2018.

[2] Efrem Yohannes Obsie, Hongchun Qu, and Francis Drummond. Wild blueberry yield prediction using a combination of computer simulation and machine learning algorithms. *Computers and Electronics in Agriculture*, 178:105778, 2020.

[3] Dhivya Elavarasan and P. M. Durairaj Vincent. Crop yield prediction using deep reinforcement learning model for sustainable agrarian applications. *IEEE Access*, 8:86886–86901, 2020.

[4] Yanxiang Yang, Jiang Hu, Dana Porter, Thomas Marek, Kevin Heflin, and Hongxin Kong. Deep reinforcement learning-based irrigation scheduling. *Transactions of the ASABE*, 63(3):549–556, 2020.

[5] Shruthi Gowda and Sangeetha Reddy. Design and implementation of crop yield prediction model in agriculture. *International Journal of Scientific & Technology Research*, VOLUME 8,:544–549, 01 2020.

[6] Lijia Sun, Yanxiang Yang, Jiang Hu, Dana Porter, Thomas Marek, and Charles Hillyer. Reinforcement learning control for water-efficient agricultural irrigation. pages 1334–1341, 12 2017.

[7] Written by, Caesweb, and posted in Bunch Grapes Grapes. https://site.extension.uga.edu/viticulture/2017/05/fruit-set-what-it-is-and-what-can-affect-it/, May 2017.

[8] Vences C. Valleser. Planting density influenced the fruit mass and yield of 'sensuous pineapple'. *International Journal of Scientific and Research Publications (IJSRP)*, 8(7), 2018.

[9] Teresa Priyanka, Pratishtha Soni, and C. Malathy. Agricultural crop yield prediction using artificial intelligence and satellite imagery. http://www.eurasianjournals.com/Agricultural-Crop-Yield-Prediction-Using-Artificial-Intelligence-and-Satellite-Imagery,105697,0,2.html, Mar 2019.

[10] Crop yield: Increased productivity with precision technologies. https://eos.com/blog/crop-yield-increase/, Mar 2021.

[11] Saeed Khaki and Lizhi Wang. Crop yield prediction using deep neural networks. https://www.frontiersin.org/articles/10.3389/fpls.2019.00621/full, Apr 2019.

[12] Real Python. Split your dataset with scikit-learn's train_test_split(). https://realpython.com/train-test-split-python-data/, Feb 2021.