# DISEASE DETECTION WITH PATTERN RECOGNITION

# 1. Introduction

## 1.1 Objective of project

**Designing an expert system that can detect the diseases of wheat plants using the images.**
The expert system using the images of the healthy plants should be able to detect the possibility of the existence disease of the wheat in the image provided to it. The expert system processes the image provided to it, and analyses with the data that it has been trained earlier and make the predictions based on system's training.

## 1.2 Description of project

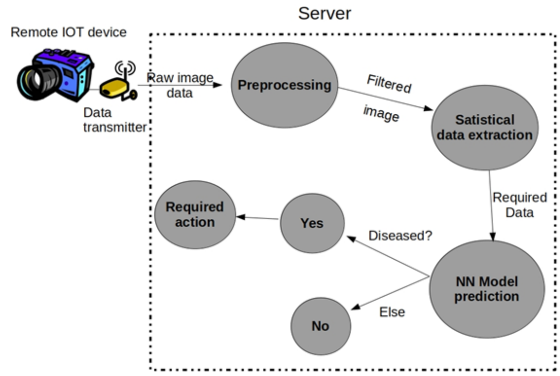
The project is the best of both Machine Learning and IoT working for the detection of the diseases in wheat. Image recognition is the preferred choice for the recognition of the diseases as diseases in wheat change the physical structure of the leaves. IoT and Machine learning play an important role in the working of the system.

IoT's role in the project has to do everything that related to capturing and sending images. Devices can used for collecting the images in a periodic interval of time in various locations of the field. The collection of images is done using digital camera which is then sent to the server/base via IoT devices like Wireless Sensing Network(WSN) which can be used to send the data with multiple devices wireless.

Machine learning comes into picture after the image is sent to the server. Advanced tasks like preprocessing, data extraction, splitting the dataset into training and testing datasets are done to support and enhance the performance and accuracy of the core machine learning model (the predictor). Neural Network is an apt model for image recognition and training of the model. Neural Network is also helpful to incorporate changes to the output of the model, for future developments.

The entire machine learning system is divided into 4 phases:

- processing
- Feature extraction
- Model system
- Required action / Additional requirements.




### 1.2.1. PRE-PROCESSING

Pre-processing is the stage in which the raw data reaching the server goes through a filtration process in which noise is reduced/removed to get required information.
Pre-processing in this project is where Image Processing comes in, as the input data is purely the image of wheat crop. Image processing will help the image to separate the non-necessary parts of the

image. Image processing for each disease is different and image processing is different for each type of crop.
Removal of noise like sunlight can be extremely crucial as the plants tend to reflect yellow color on exposure to sunlight and the diseases like Yellow Rust is detected by taking the yellow pores in the wheat leaf blades.

### 1.2.2 FEATURE EXTRACTION

In this stage, the processed information, which is in the form of table, is taken and the required feature columns are taken out, for the model to detect. Feature extraction can also be called the second level of processing since the many features that are not required are also filtered out.
The pre-processed data that is in the form tables contains features sets that are not required by the model are not considered and the required features are taken out in another table. Feature selection can vary for different species of crops and for different diseases.

### 1.2.3 MODEL SYSTEM

Model system essentially is the core evaluator that predicts the existence of disease using the feature set. Neural Networks is the preferred model for the recognition of the patterns and does a good job at recognizing and adapting to the changes in the input over a period of time. There outcomes of the model can be represented as class. In this project, there are two classes-

- Plant is affected with disease
- Plant is not affected with disease

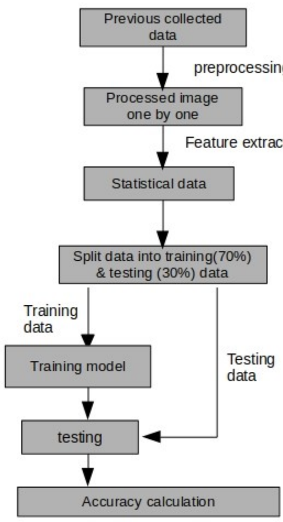
Model should be fed some amount of data for the training to show accurate results. Then particular amount of data is fed into the model for testing the images and the results are compared with the actual results. This is Supervised learning. Supervised learning is the learning that can best adapt to pattern recognition as the images are abundantly available for the model to get trained. Outputs of the Neural Network model is Yes or No/ Diseased or Healthy.

The building stages of the core model has different tasks during different phases of development. There are two phases for the model

1. Training phase
2. Prediction phase

#### *1.2.3.1 Training phase*

Training phase comes before the prediction phase. Training phase has few extra steps, even though computationally intensive, is crucial for the improvement of accuracy of the prediction. Our model uses supervised learning technique. The images of diseased and the healthy wheat crop, along with the output of the image (diseased or not), are used to train the model. Various stages include accuracy calculation, dataset split.



*1.2.3.1.1 Training and testing data split*

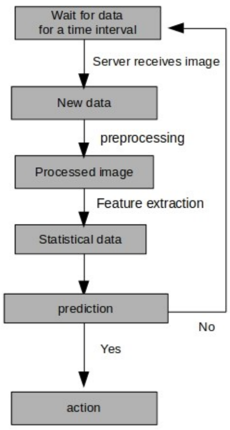
Training and testing data split is used to divide the given dataset of the user into testing and training. Training dataset is used to train the model by adjusting the weight of the Neural Network. Testing dataset is used to calculate the accuracy of the model. In order to tackle the issues like Overfitting, techniques like K-Fold techniques can be used for training.

*1.2.3.1.2 Accuracy prediction*

Accuracy prediction is used to calculate the correctness of the model. The adjusted weights during the training is then checked with the testing data. The results of the testing data produced by the model is compared with the actual outcomes of the testing data. The accuracy is calculated based on the testing data.

*1.2.3.2 Prediction phase*

Prediction phase can also be called as the production version of the Neural Network model and is the final version of the model before any further changes are done to the model. Prediction phase is used to give the prediction/judgment, when a new dataset comes to the model. Prediction phase, unlike training phase does not include steps like training and testing. Prediction phase, combined with the capabilities of IoT, automate the task of prediction of the diseases without the need humans. Final output of the model is to decide if the image of the wheat crop is diseased or not.




**1.2.4 REQUIRED ACTIONS**

Required action is the part done after the classes of the data has been predicted. This is the stage where additional operations are done and results based on the current input and its predicted class is given.

## 1.3 Scope of the project

The project is designed to automate the entire process of detection of disease in the wheat plant using the picture of the leaf, as most of the symptoms of the diseases are visible in the leaf. The goal of the project was to remove the need of humans to solve the problem of detection upto an extent, especially in the phase of disease detection. Using humans to detect the diseases in the wheat plants is a naive approach to solve the problem. This is project is also used as a start to explore the capabilities of Machine learning in various fields in the industry.

The project, at the initial stage, is designed to detect the possibility of one disease in a single plant. The training of the model is also based on a single disease. As several diseases, each of it's own kind, are subjectably a challenge for the model to detect the diseases. Each disease has a different filters applied during the phase of preprocessing, as the image of wheat captured can have "Yellow Rust" (Yellow patches) or "Powdery Mildew" (White spots). Since the model is trained to detect the diseases as patches in the leaf, only one model is sufficient to detect the diseases. This opens up door for the future development in which there is a separate model for choosing the filtering option during image processing. This helps improve the performance but also happens to be very complex and bugs are very complex to fix. Hence in future, a single picture of the plant can be subjected to detection of several diseases in the phase of preprocessing.

Initial model during it's detection phase, evaluates the diseases from pictures of a single leaf, instead of pictures of the entire field. Entire field is the future goal of the project where less number of cameras are required in order to setup a production level of the implementation of the project.

The entire project can be said to be a unique version of Client-Server model in networking. Client in this project are the cameras with IoT devices placed in the fields. IoT devices transmit the images using technologies like WSN (Wireless Sensor Network) or data services. The server is the house for the brain of the project. Server is the place where the entire model exists, which detects and does the required action.

The required action can be a future part of the end result of the project where certain tasks are done by the server after the disease has been detected in the wheat plant. These events can also be used to analyse the trend in the data and store the results for future use.
The actions and calculations include:

- Notifying the farmer when the diseased class is predicted by the model
- Calculating the degree of severity of the disease.
- Calculating the type of disease affected.
- Recommendations of pesticides for the disease.
- Data analysis.

# 2. System Description

## 2.1. Customer / User profiles

The project is expected to be widely used by the following classes:

- Farmers.
- Setup manager.
- Maintainance persons.
- Machine learning scientists.

### 2.1.1 Farmers

- Farmers use it to get the information about the diseases in their fields.

### 2.1.2 Setup manager

- Setup manager uses the project to setup the model in the respected fields
- Training and feeding part is done by the setup manager.
- Placing IoT devices are done by setup manager.
- connectivity to the server is also done by setup manager.

### 2.1.3 Maintainance person

- Maintainance person makes sure the devices are intact and working.
- Fixes the non working devices.
- Makes sure to fix the bugs in the project.

### 2.1.4 Machine Learning scientist

- Understand the working of the project.
- Explore more projects in these areas.

- Further develop the project to increase possibilities.
- Create more projects in this field.

## 2.2. Assumptions and Dependencies:

**Assumptions:**

- Image is being captured and sent using IoT device.
- High quality images are used.
- Images sent are of wheat plant only.
- Farmer has knowledge about phones.

**Dependencies:**

- Model is present in server only.
- Network connectivity for transfer of data.
- Pre existing data of diseases is available for training.

- The first assumtion deals with the image. The project is mainly focused on the creation of the machine learning model and the core compnents related to the project like image processing and feature extraction. Hence we are assuming that the image is being fed to the model.
- High quality images are being assumed as the images produced by the camera is enough for the model to evaluate the data. Low quality images can produce false results resulting in loss of human effort and time.
- As the inital disease detection is based on wheat only, pictures of wheat plants are expected as the training of the model is designed for wheat as the diseases of the wheat are visible physically.
- If farmer has knwoledge about the phone, farmer can be notified to his/her mobile once the model detects disease in a particular image.
- The model residing in the server is the first dependency for the evaluation of the model. As the image training and the model performance can be intensive, server is apt for dealing with large computation power and also helping with multiple request and high resolution images. It is also helpful to place a single remote server in a particular area serving the needs for several farmers.
- Network connectivity is crucial for the model to work as the images that are captured are sent to the server by the IoT devices placed in the fields.
- Pre-existing data is really important for the evaluation and training of the model. As this is supervised model, pre-existing data of the diseases help the model detect when a new picture is received.

## 2.3 Functional Requirements

### 2.3.1. Image processing:

**R1: Sunlight removal**

- **Description:** Sunlight needs to be removed from the image as images taken in the day light contain the yellow pigment.
- **Input:** Raw image
- **Processing:** Removing yellowish green or light green pigments.
- **Output:** Processed image without sunlight effects and black values in the areas of light green color.

**R2: Green pigments removal**

- **Description:** Green pigments need to removed as the leaf color of healthy wheat plants is green.
- **Input:** Image
- **Processing:** Removing green pigments.
- **Output:** Processed image without green color and black values in the areas of green color.

**R3: Background removal**

- **Description:** Background regions need to removed to focus on the leaf
- **Input:** Image
- **Processing:** Removing blurred areas.
- **Output:** Image without background areas.

**R4: Patches as binary input**

- **Description:** Rest of the areas of the
- **Input:** Image
- **Processing:** changing non zeros values to one.
- **Output:** Image with patches in white color.

**R5: convert to tabular form**

- **Description:** Final image needs to be converted to tabular form for the feature extraction phase.
- **Input:** Image
- **Processing:** Converting pigment values to tabular form.
- **Output:** Table of rows and column of pixel values one or zero.

### 2.3.2 Feature extraction

**R1: One hot encoding**

- **Description:** Conversion of output/input values into one and zero representing diseased or not. Preferrable choice when model is a Neural Network.
- **Input:** Table of processed image
- **Processing:** Converting particular column into zero and one.
- **Output:** Processed table.

**R2: Dimensionality reduction**

- **Description:** As the table can be large, multiple columns in the table are reduced to certain number of columns.
- **Input:** Table of processed image
- **Processing:** Converting multiple columns to reduced columns.
- **Output:** Processed table.

**R3: Normalization**

- **Description:** Dimensions of each column can be very different, which might lead to performance downfall and increasing learning error.
- **Input:** Table of processed image
- **Processing:** Column values changed to desirable range
- **Output:** Processed table.

**R4: Regularization**

- **Description:** The problem of overfitting is common when it comes to Neural Network. Regularization is used to remove certain features by altering the values of certain features and to improve accuracy as well.
- **Input:** Table of processed image.
- **Processing:** Increasing certain feature column values.
- **Output:** Processed table.

### 2.3.3 Model:

**R1: Train test split**

- **Description:** Feature table needs to be divided into two tables, for training the model.
- **Input:** Feature table and output table.
- **Processing:** separating certain rows into test table.

- **Output:** Train and test tables of feature and output tables.

**R2: Fitting**

- **Description:** The train and test tables obtained must to used to train and test the model. Model adjusts the weights inorder to produce accurate results.
- **Input:** Train and test tables
- **Processing:** Adjust the weights in the model.
- **Output:** Trained model

**R3: Preditiction**

- **Description:** The trained model should predict the values of the new dataset that is presented to it. The inputs are multiplied with the weights along with bias and produce output.
- **Input:** New dataset (table).
- **Processing:** Passing each row as input dataset and computing the result.
- **Output:** Binary value determining if the given dataset is diseased or not.

**R4: Accuracy prediction**

- **Description:** After the training, the model must use the test table to determine the accuracy of the model.
- **Input:** Test table.
- **Processing:** calculate the percentage of the correctly predicted outputs.
- **Output:** Percentage score.

**R5: Confusion matrix plot**

- **Description:** The confusion matrix is used for analysing the predictions of the model. Confusion matrix for binary output is 2x2 matrix where rows are actual output and columns represent predicted values.
- **Input:** Test table.
- **Processing:** Calculate the count of the correctly predicted outputs.
- **Output:** Confusion matrix graph.

## 2.4. Non Functional Requirements

### 2.4.1 Security

- Security of the network is crucial as the amount of data sent in an interval of time needs to secure.
- Privacy of the farmer’s farm must be safe from the third party access.
- Encryption algorithms must used at the IoT device to securely send the data.
- The server must be protected from unautorized entry to prevent the altering the model.

### 2.4.2 Performance

- The model must perform efficiently to reduce the computational costs.
- The model must produce fast results.
- The model must have accuracy between 90% – 99%.

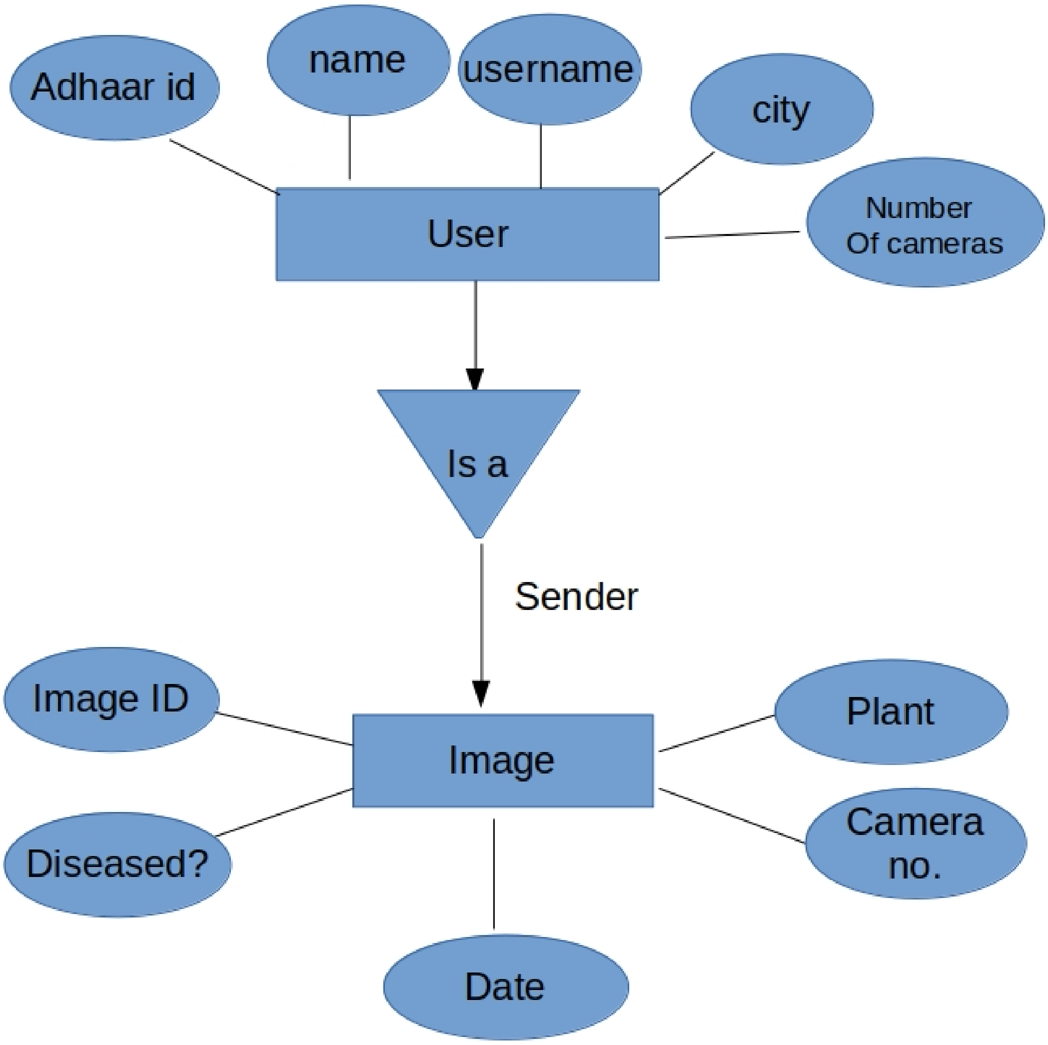
### 2.4.3 Database

- The database must store the details and images where the disease is detected by the model.
- Storing helps for future analysis and learning.
- Databases also acts like a reference to past data.

# 3 Design

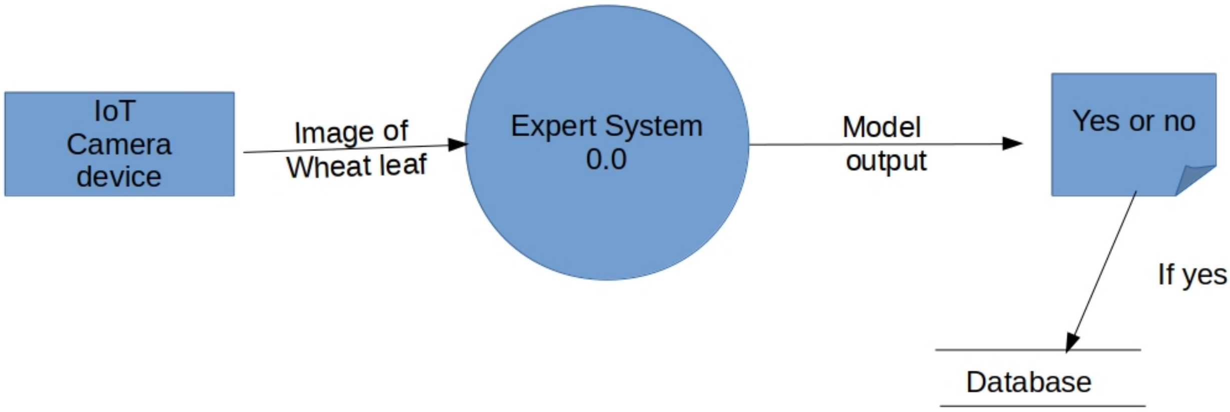
## 3.1 System design

### 3.1.1 ER Diagram

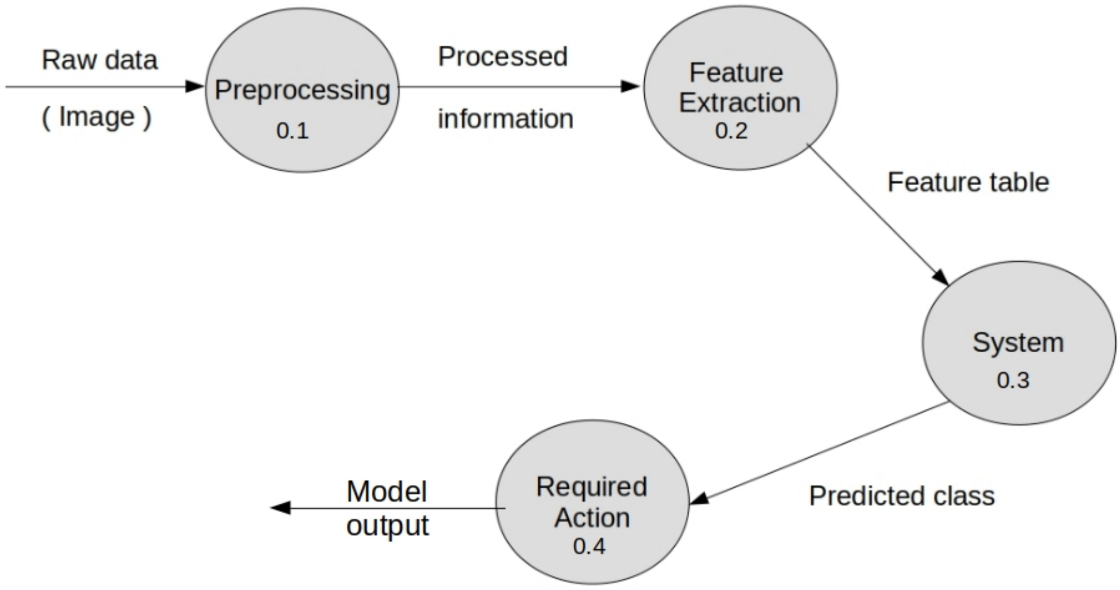


### 3.1.2 DFD

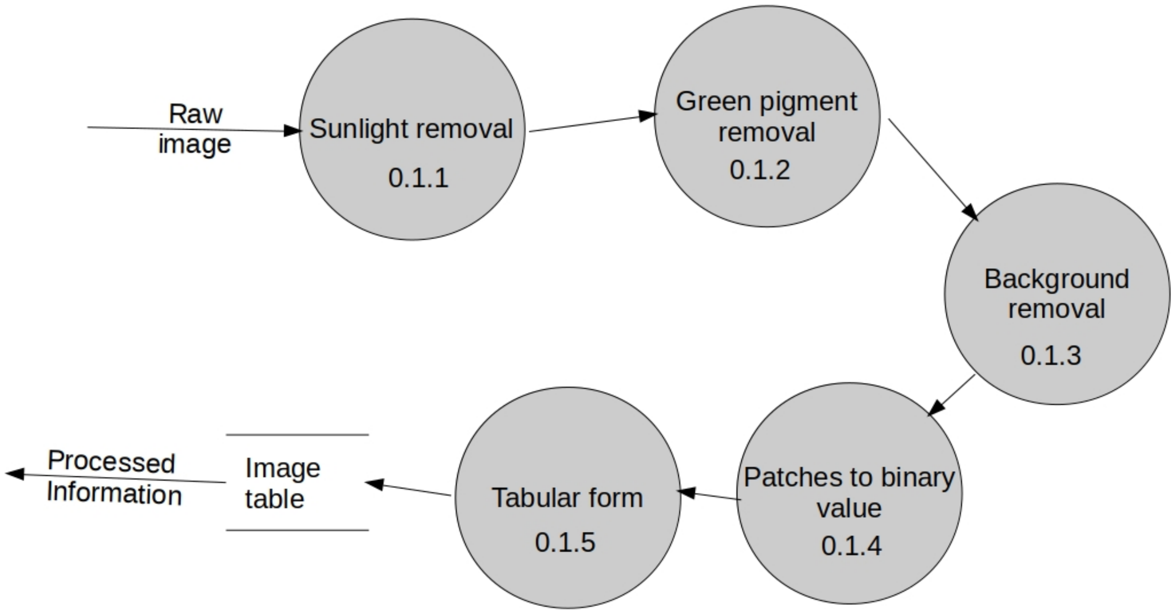
Level 0



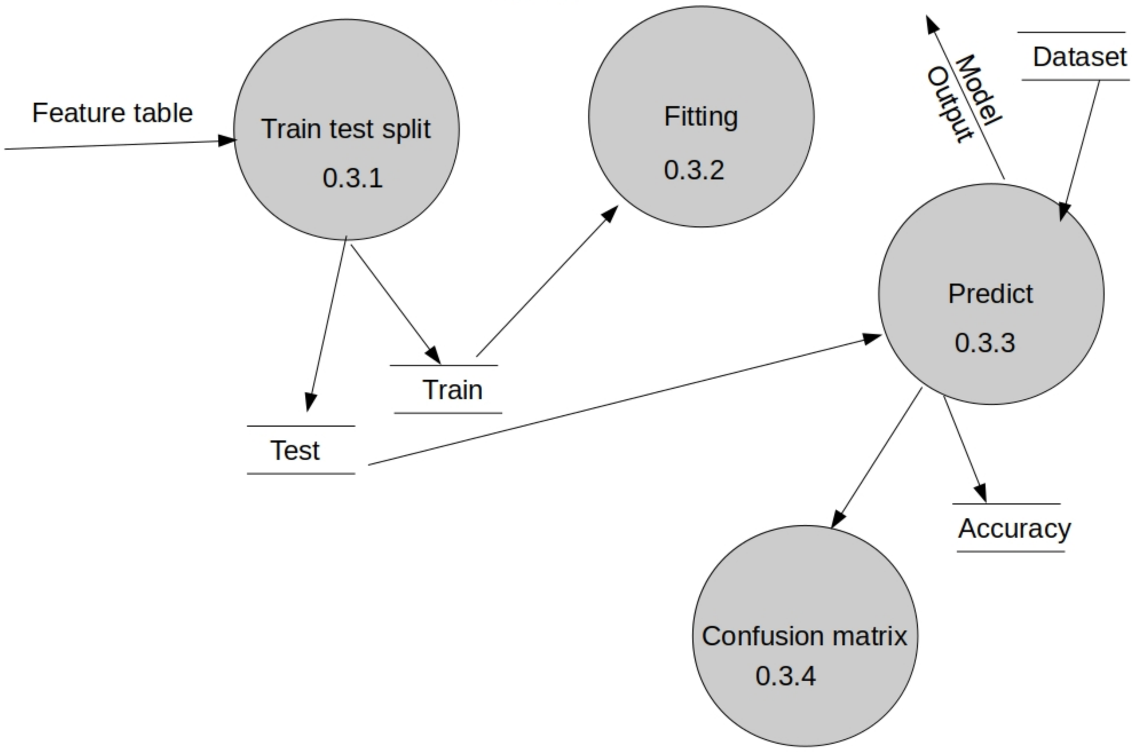

Level 1

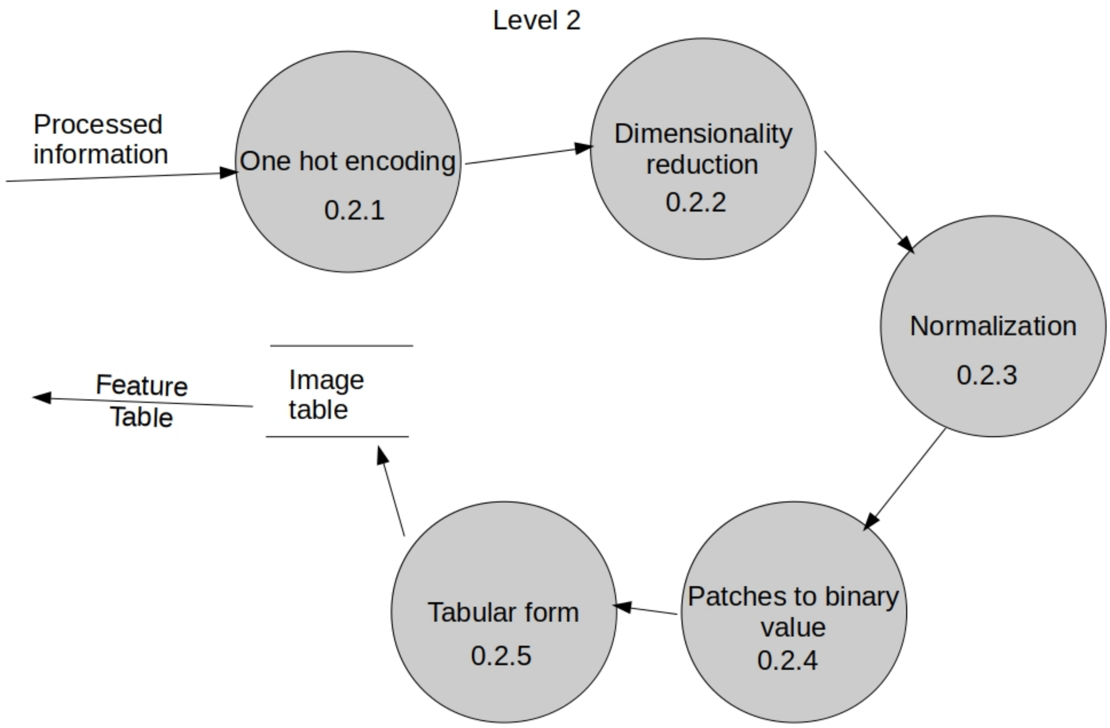


Level 2



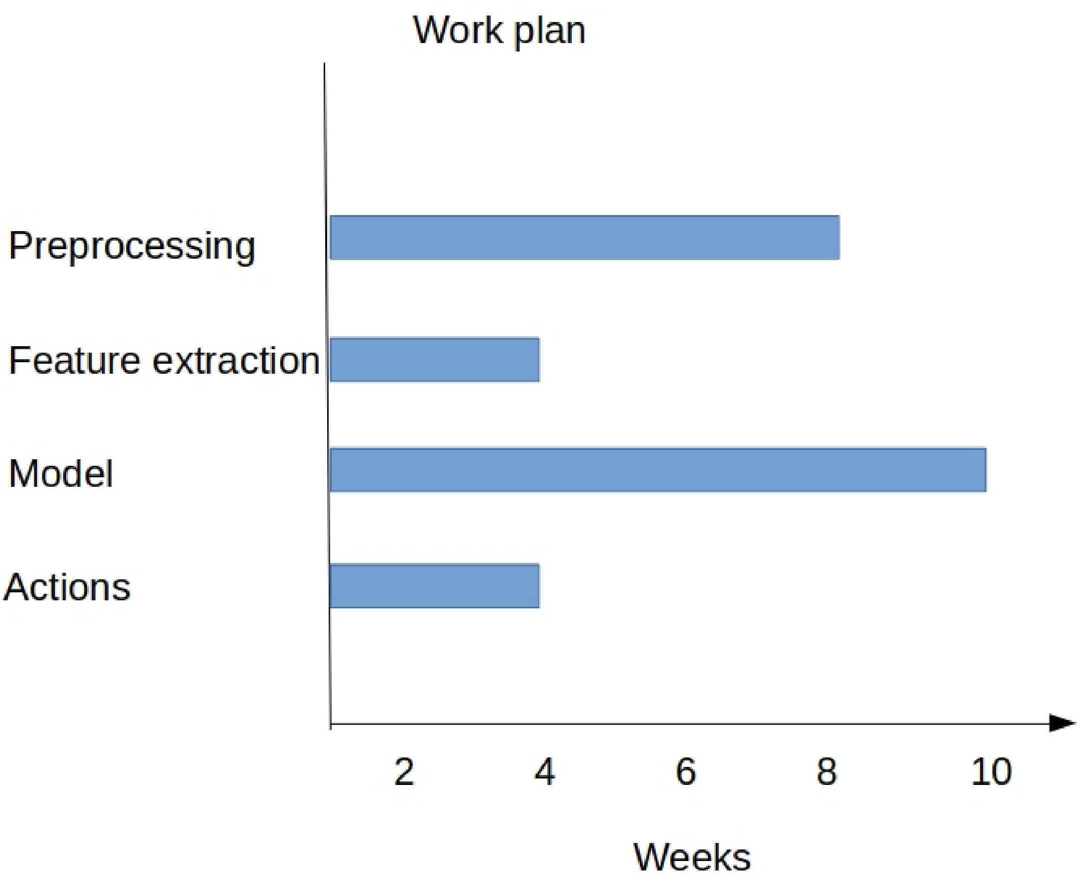

Level 2






# 4. Scheduling and Estimation

## 4.1 Ghant chart




**Estimated Time:** 4 Months