

Detail Project Report  
Forest Cover  
Classification

Revision Number – 1.0

Last Date of Revision : 27– 07 -2022

Balamurali K

## Document Version Control

Date	Version	Description	Author
27 – 07 - 2022	1.0	Abstract Introduction General Description	Balamurali
27 – 07 - 2022	1.1	Technical Requirements Data Requirements Data Preprocessing Design Flow	Balamurali
27– 07 - 2022	1.2	Data from User and its validation Rendering the Results Deployment Conclusion	Balamurali

# Contents

Document Version Control

.....  
2 Abstract

.....  
.....

..... 6

- Introduction

.....  
.....

.... 7

- Why this DPR Document ?

..... 7

- General Description

.....  
..... 7

- Problem Perspective

..... 7

- Problem Statement

.....  
..... 7

- Proposed Solution

.....  
..... 7

- Technical Requirements

.....  
8

- Tools Used

.....  
.....

- Data Requirements

.....  
 ..... 8

- Data Preprocessing

.....  
 ..... 8

- Design Flow

.....  
 .....  
 ... 9

- UI Integration

.....  
 ..... 9

- Logging

.....  
 .....  
 ..... 9

- Data from User

.....  
 .....  
 9

- Data Validation

.....  
 .....

## 9 9 Rendering the Results

.....  
 ..... 9

- Deployment

.....  
 .....  
 . 10


- Conclusion

.....  
.....  
.... 10

- FAQs

.....  
.....  
..... 10

## Abstract

 The identification of type of forest cover usually requires the knowledge and expertise about the land and physical presence. This work discusses the implementation of forest cover prediction based on observations of various catagrophic features (No sensor data) .

# 1. Introduction

## 1.1 Why this DPR Document ?

The main purpose of this DPR documentation is to add the necessary details of the project and provide the description of the machine learning model and the written code. This also provides the detailed description on how the entire project has been designed end-to-end.

Key points :

- Describes the design flow
- Implementations
- ☐ ☐ Software requirements    Architecture of the project  
Non-functional attributes like:
- Reusability

- Portability
- Resource utilization

## 2. General Description

### 2.1 Problem Perspective

The Forest Cover Prediction is a machine learning-based default prediction model which will help us to know the type of forest cover present in the area based on various catagrophic features (No sensor data).

### 2.2 Problem Statement

Identifying the type of forest cover usually requires the expertise and knowledge of surrounding areas. This problem is solved by training a machine learning model based on observation of various categrorphic features

### 2.3 Proposed Solution

This project has been developed in semi-supervised learning way. Initaly, the data without the target column is passed through a clustering model which predicts the cluster. Then for each cluster, a ML supervised learning model has been used to predict the type of forest cover.

## 3. Technical Requirements

As technical requirements, we doesn't need any specialized hardware for virtualization of the application. The user should have the device that has the access to the web and the fundamental understanding of providing the input.

### 3.1 Tools Used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Kneed, Flask used to build the whole model.

- VScode is used as IDE.
- For visualization of the plots, Matplotlib, Seaborn are used

- Pycaret with the integration of MLflow is used for performing and logging the experiments respectively.
- DVC is used for tracking the data, code and model.
- Heroku is used for deployment of the model.
- Front end development is done using HTML/CSS.
- Python is used for backend development.
- GitHub is used as version control system.

## 4. Data Requirements

The Data requirements is totally supported the matter statement and also the dataset is accessible on the Kaggle within the file format of (.xlsx).

## 5. Data Preprocessing

- Checked for info of the Dataset, to verify the correct datatype of the Columns.
- Validated the filename whether in the correct format
- Checked for Null values, because the null values can affect the accuracy of the model.
- Checking the distribution of the columns to interpret its importance.
- Now, the info is prepared to train a Machine Learning Model.

## 6. Design Flow

### 6.1 UI Integration

Both CSS and HTML files are being created and are being integrated with the created machine learning model. All the required files are then integrated to the app.py file and tested locally.

### 6.2 Logging

In logging, at each if an error or an exception is occurred, the event is logged into the system log file with reason and timestamp. These helps the developer to debug the system bugs and rectifying the error.



## 7. Data from User

The data from the user is retrieved from the created HTML web page.

## 8. Data Validation

The data provided by the user is then being processed by app.py file and validated. The validated data is then sent to the prepared model for the prediction.

## 9. Rendering the Results

The data sent for the prediction is then rendered to the web page.

## 10. Deployment

The tested model is then deployed to Heroku. So, users can access the project from any internet devices.

## 11. Conclusion

The Forest Cover Prediction system will help the prediction of type of forest cover based on the observation of catagrophic data.

## 12. Frequently Asked Questions ( FAQs )

Q1) What's the source of data ?

The data for training is provided by the client in multiple batches and each batch contain multiple files.

Q2) What was the type of data ?

The data was the combination of numerical and Categorical

values.

Q3) What's the complete flow you followed in this Project ?

Refer Page no 6 for better Understanding.

Q4) After the File validation what you do with incompatible file or files which didn't pass the validation ?

Files like these are moved to the Achieve Folder and a list of these files has been shared with the client and we removed the bad data folder.

Q5) How logs are managed ?

We are using different logs as per the steps that we follow in validation and modeling like File validation log, Data Insertion, Model Training log, prediction log etc.

Q6) What techniques were you using for data pre-processing ?

- Removing unwanted attributes.
- Visualizing relation of independent variables with each other and output variables.
- Checking and changing Distribution of continuous values.
- Removing outliers
- Cleaning data and imputing if null values are present.
- Converting categorical data into numeric values.

Q7) How training was done or what models were used ?

- Before dividing the data in training and validation set, we performed preprocessing over the data set and made the final dataset.
- As per the dataset training and validation data were divided.
- Algorithms like Linear regression, SVM, Decision Tree, Random Forest, XGBoost were used based on the recall, final model was used on the dataset and we saved that model.

Q8) How Prediction was done ?

The testing files are shared by the client. We Performed the same life cycle on the provided dataset. Then, on the basis of dataset, model is loaded and prediction is performed. In the end we get the accumulated data of predictions.

Q9) What are the different stages of deployment?

- First, the scripts are stored on GitHub as a storage interface.
- The model is first tested in the local environment.
- After successful testing, it is deployed on Heroku.