Project Title: Forest Cover Type Prediction Using Machine Learning
Name: Muhammed Saneen Ak
Internship: Machine Learning Internship
Institution: Unified Mentor Pvt. Ltd
Date: 20/06/2025

Acknowledgment

I express my heartfelt gratitude to my mentors and peers who supported me in completing this project. I would also like to thank the internship program for providing access to real-world datasets and encouraging impactful problem-solving using machine learning.

Abstract

This project focuses on predicting forest cover types based on cartographic variables such as elevation, soil type, slope, aspect, and more. Using supervised machine learning algorithms, we train a model to classify seven different cover types found in the Roosevelt National Forest. The aim is to create a reliable and efficient predictive model to support forest management and conservation efforts.

Table of Contents

1. Introduction

Forest cover classification is crucial for resource management, wildfire prevention, and conservation. This project uses a UCI dataset to classify the type of forest cover based on 54 numerical and categorical features representing terrain, soil, and geographical attributes.

2. Problem Statement

Manual classification of forest cover is inefficient and error-prone. A predictive model can assist forest officials in identifying forest types automatically and accurately using geospatial data.

3. Objective

- To develop a machine learning model that predicts forest cover types.
- To improve classification accuracy using feature engineering and algorithm tuning.
- To evaluate the model using appropriate performance metrics.

4. Tools and Technologies Used

- **Language**: Python
- **Libraries**: NumPy, Pandas, Scikit-learn, XGBoost, Matplotlib, Seaborn
- **Environment**: Jupyter Notebook
- **Visualization**: Seaborn, Matplotlib
- **Algorithm**: Decision Tree, Random Forest, XGBoost, etc.

5. Dataset Description

- Source: UCI Machine Learning Repository
- Target: Forest Cover Type (1–7)
- Features: 54 columns including Elevation, Aspect, Soil_Type, Wilderness_Area etc.
- Preprocessing involved handling imbalanced classes, standardizing continuous variables, and encoding categorical features.

6. Methodology

- Data preprocessing: null check, normalization, label encoding
- EDA: Distribution analysis, correlation heatmaps
- Model selection and training
- Hyperparameter tuning using GridSearchCV
- Evaluation using accuracy, confusion matrix, and F1-score

7. Implementation

- Tried multiple classifiers: Random Forest, XGBoost
- XGBoost gave the best performance with tuned parameters

```
xgb_model = XGBClassifier(n_estimators=100, max_depth=8, learning_rate=0.1, use_label_encoder=False, eval_metric='mlogloss', random_state=42)
xgb_model.fit(xtrain, ytrain)
```

```
C:\Users\muham\AppData\Local\Programs\Python\Python312\Lib\site-packages\xgboost\training.py:183: UserWarning: [20:38:46] WARNING: C:\actions-runner\_wor
k\xgboost\xgboost\src\learner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
```

```
XGBClassifier
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric='mlogloss',
              feature_types=None, feature_weights=None, gamma=None,
              grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=0.1, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=8, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              multi_strategy=None, n_estimators=100, n_jobs=None,
```

```
pred_xgb = xgb_model.predict(xtest)
acc_xgb = accuracy_score(ytest, pred_xgb)
print("XGBoost Test Accuracy:", acc_xgb)
```

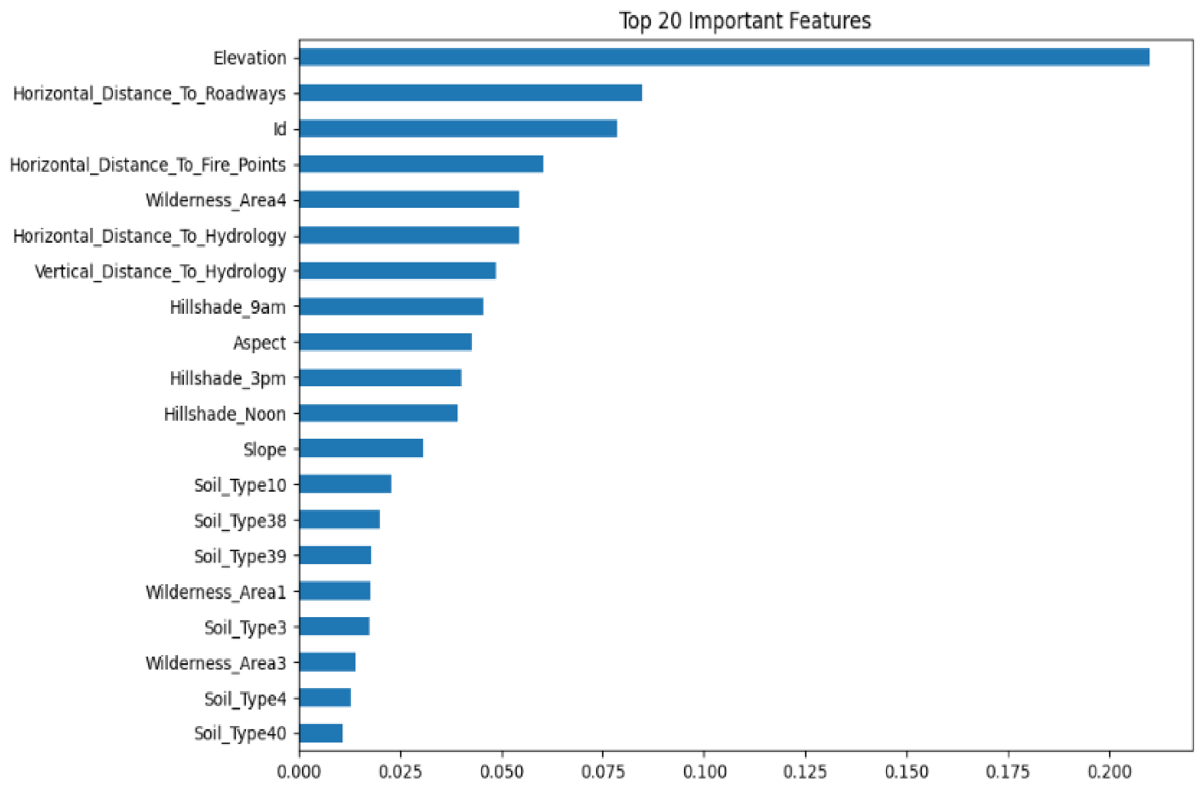
```
XGBoost Test Accuracy: 0.8713624338624338
```

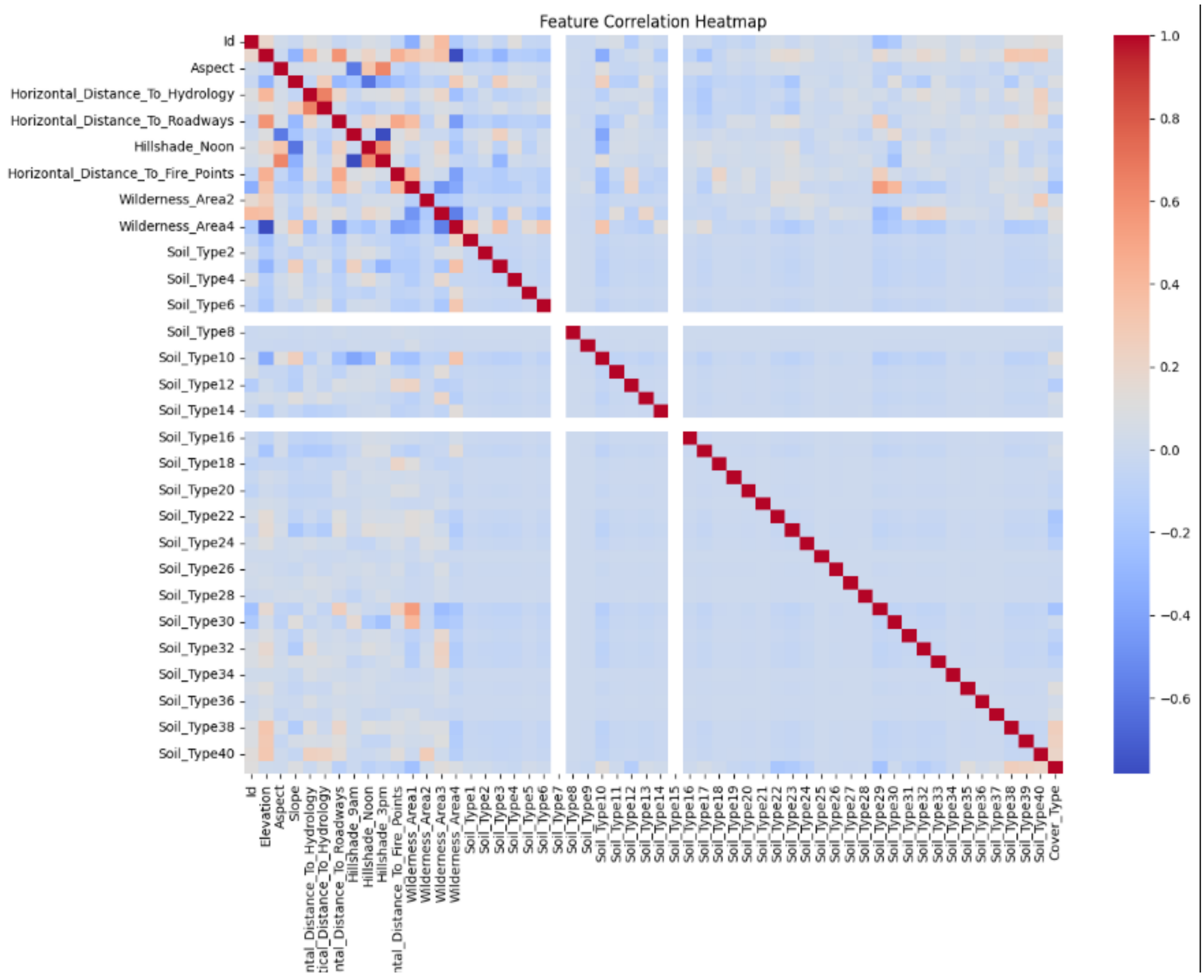
8. Challenges Faced

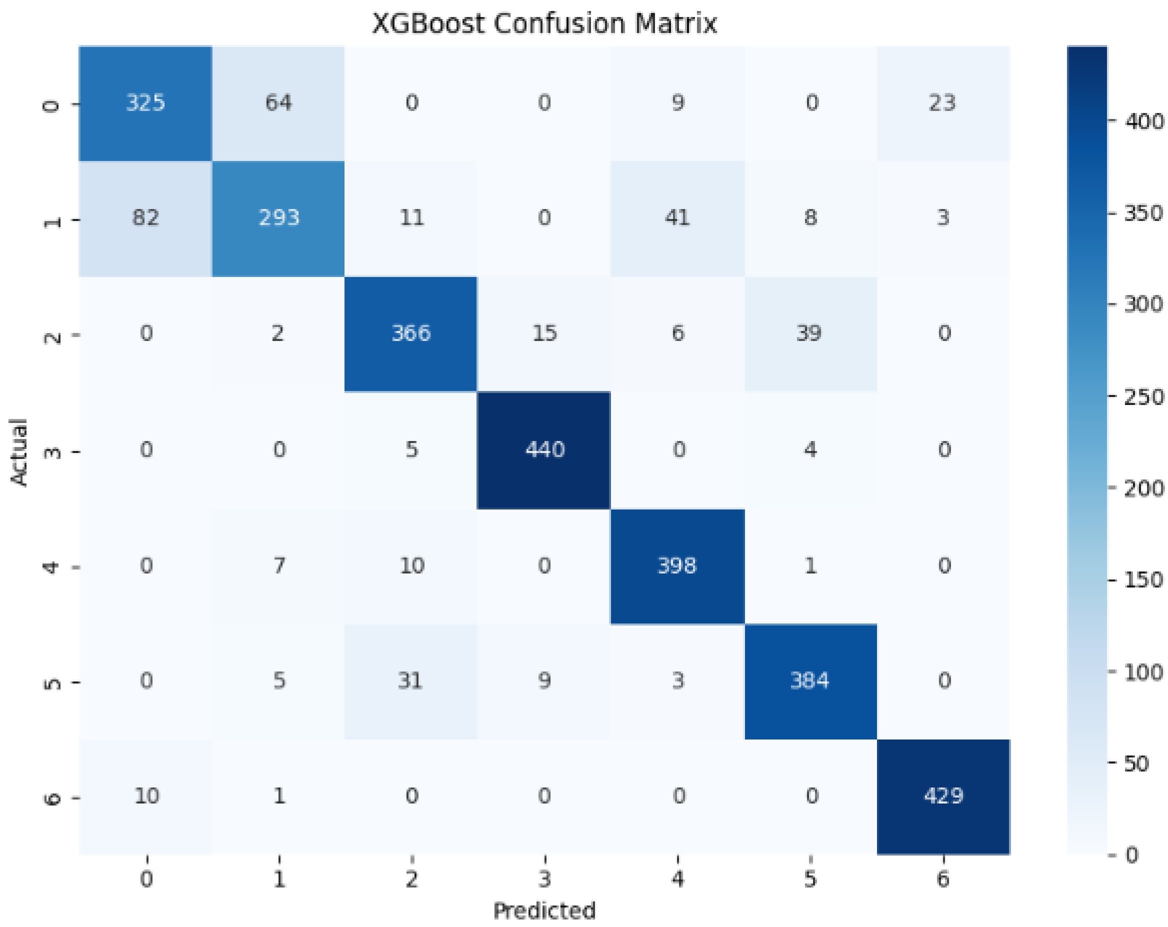
- High feature dimensionality required careful preprocessing
- Some classes were underrepresented (class imbalance)
- Model overfitting during early experimentation

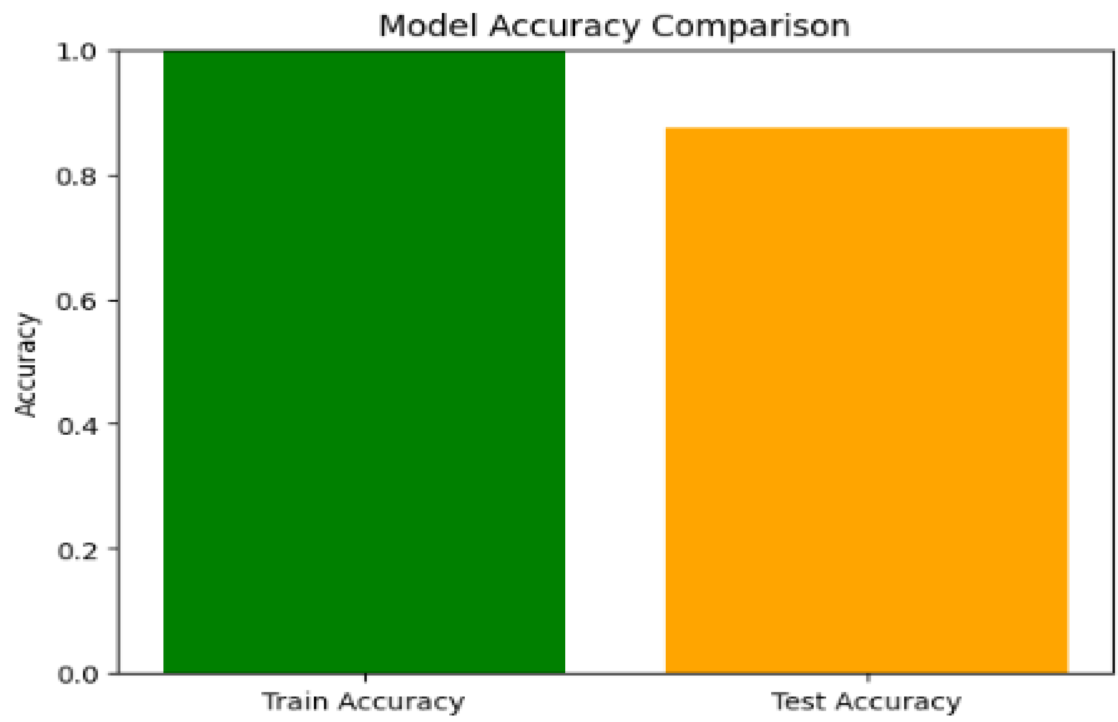
9. Results and Evaluation

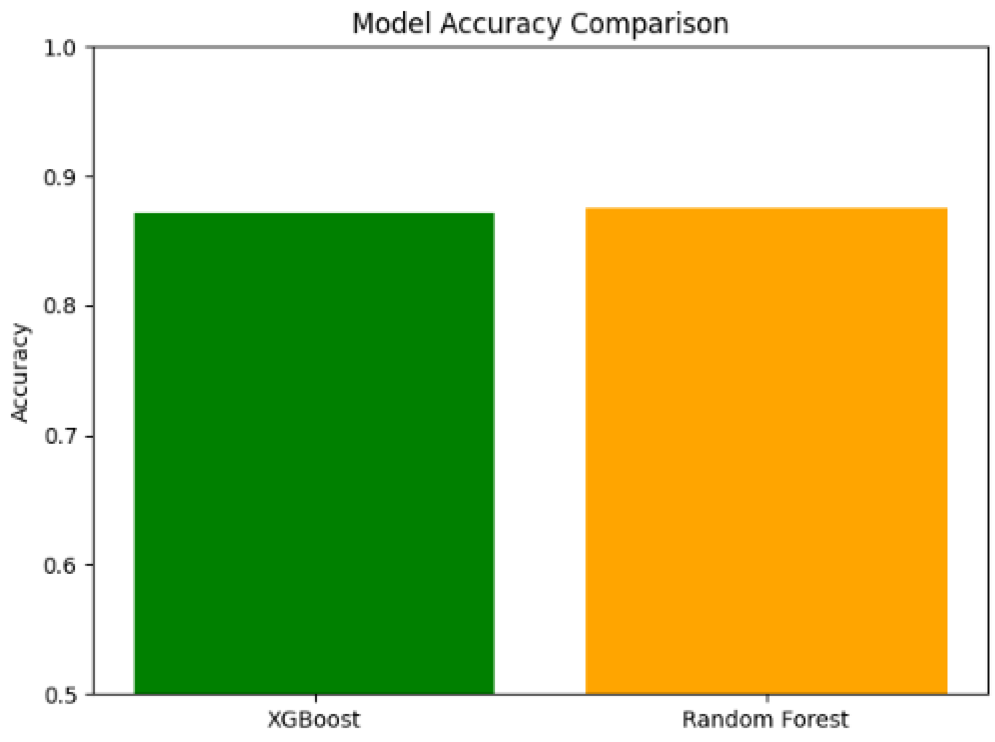
- **Best Accuracy**: ~87% on test data
- **Model Used**: XGBoost
- **Confusion Matrix**: Clear separation of most classes with minor confusion in neighboring forest types
- **MAE**: ~29
- **MSE**: ~99
- Visualizations included feature importance, prediction vs actual plots


Top 20 Important Features
Elevation
Horizontal_Distance_To_Roadways
Id
Horizontal_Distance_To_Fire_Points
Wilderness_Area4
Horizontal_Distance_To_Hydrology
Vertical_Distance_To_Hydrology
Hillshade_9am
Aspect
Hillshade_3pm
Hillshade_Noon
Slope
Soil_Type10
Soil_Type38
Soil_Type39
Wilderness_Area1
Soil_Type3
Wilderness_Area3
Soil_Type4
Soil_Type40
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200


Feature Correlation Heatmap
Id
Aspect
Horizontal_Distance_To_Hydrology
Horizontal_Distance_To_Roadways
Hillshade_Noon
Horizontal_Distance_To_Fire_Points
Wilderness_Area2
Wilderness_Area4
Soil_Type2
Soil_Type4
Soil_Type6
Soil_Type8
Soil_Type10
Soil_Type12
Soil_Type14
Soil_Type16
Soil_Type18
Soil_Type20
Soil_Type22
Soil_Type24
Soil_Type26
Soil_Type28
Soil_Type30
Soil_Type32
Soil_Type34
Soil_Type36
Soil_Type38
Soil_Type40
Elevation
Slope
Hillshade_9am
Hillshade_3pm
Wilderness_Area1
Wilderness_Area3
Soil_Type1
Soil_Type3
Soil_Type5
Soil_Type7
Soil_Type9
Soil_Type11
Soil_Type13
Soil_Type15
Soil_Type17
Soil_Type19
Soil_Type21
Soil_Type23
Soil_Type25
Soil_Type27
Soil_Type29
Soil_Type31
Soil_Type33
Soil_Type35
Soil_Type37
Soil_Type39
Cover_Type
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6


XGBoost Confusion Matrix
Actual
Predicted
325
64
0
0
9
0
23
82
293
11
0
41
8
3
0
2
366
15
6
39
0
0
0
5
440
0
4
0
0
7
10
0
398
1
0
0
5
31
9
3
384
0
10
1
0
0
0
0
429
0
1
2
3
4
5
6
400
350
300
250
200
150
100
50
0


Model Accuracy Comparison
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Train Accuracy
Test Accuracy



10. Project Scope and Future Work

- Use satellite imagery and integrate remote sensing data
- Apply deep learning (MLP or CNN on terrain maps)
- Deploy as a web service for forest department usage
- Incorporate real-time geolocation-based predictions

11. Conclusion

This project achieved high accuracy in predicting forest cover types using structured terrain data. With proper preprocessing and model tuning, machine learning can be a reliable tool for supporting environmental sustainability and forest management.

12. References

- UCI ML Repository: Forest Cover Type Dataset
- XGBoost Documentation
- Scikit-learn Documentation
- Research articles on environmental ML models