Project Title	**EcoType: Forest Cover Type Prediction Using Machine Learning**
📚 **Skills Take Away**	● Exploratory Data Analysis (EDA) ● Data Cleaning and Preprocessing ● Feature Engineering ● Classification Model Building ● Model Evaluation ● Hyperparameter Tuning ● Streamlit App Development
🌿 **Domain**	Environmental Data & Geospatial Predictive Modeling

📌 Project Statement:

To develop a machine learning classification model that predicts the **type of forest cover** in a given geographical area based on cartographic variables like elevation, soil type, slope, and wilderness area information. The goal is to assist in environmental monitoring, forestry management, and land use planning by providing automated, reliable forest cover type identification.

📌 Real-World Use Cases:

- **Forest Resource Management:** Assist forestry departments in identifying and classifying forest areas for planning, conservation, and logging.

- **Wildfire Risk Assessment:** Combine vegetation type with fire risk models to predict high-risk zones.

- **Land Cover Mapping:** Used by environmental scientists and geospatial analysts for mapping and monitoring land usage patterns.

- **Ecological Research:** Support studies in biodiversity, soil conservation, and habitat analysis.

📊 Dataset:

- **Source:** Dataset Link

- **Size:** 1,45,891 rows × 55 columns

- **Target Variable:** `Cover_Type` (7 classes)

📊 Columns Description

S.No	Column Name	Description	Type
1	**Elevation**	Elevation in meters	Numerical
2	**Aspect**	Aspect in degrees (azimuth)	Numerical
3	**Slope**	Slope in degrees	Numerical
4	**Horizontal_Distance_To_Hydrology**	Horizontal distance to nearest surface water (meters)	Numerical
5	**Vertical_Distance_To_Hydrology**	Vertical distance to nearest surface water (meters)	Numerical
6	**Horizontal_Distance_To_Roadways**	Horizontal distance to nearest roadway (meters)	Numerical
7	**Hillshade_9am**	Hillshade index at 9:00 AM	Numerical
8	**Hillshade_Noon**	Hillshade index at noon	Numerical

9	**Hillshade_3pm**	Hillshade index at 3:00 PM	Numerical
10	**Horizontal_Distance_To_Fire_Points**	Horizontal distance to nearest wildfire ignition point (meters)	Numerical
11–14	**Wilderness_Area_1 to Wilderness_Area_4**	Binary flag for 4 wilderness area categories (one-hot encoded)	Categorical (0/1)
15–54	**Soil_Type_1 to Soil_Type_40**	Binary flag for 40 soil types (one-hot encoded)	Categorical (0/1)
55	**Cover_Type**	Forest cover type (7 classes)(target)	Categorical (To be label encoded)

🔧 Tasks & Workflow:

1️⃣ Data Collection

- Download and load the complete **Forest Cover Type** dataset using Pandas.
- Understand the dataset's structure, column descriptions, and target classes.

2️⃣ Data Understanding

- Explore and understand the dataset using Pandas functions like `.shape`, `.info()`, `.describe()`, and `.value_counts()`.
- Check for duplicate records, missing values, and **class distribution imbalance**.

3️⃣ Data Cleaning & Transformation

- Handle missing values by imputing with appropriate techniques (like median,mean,mode). (simple imputer)
- Detect and handle outliers using Z-score or IQR method for numeric features.
- Fix skewness in continuous variables using suitable transformations (e.g., log1p).

4️⃣Feature Engineering

- Analyze which existing features might require transformation or interaction terms.
- Optionally create derived columns if required to improve model interpretability (like distance ratios or shade indices differences).
- Ensure that all categorical columns are **label encoded** before training the model.
- The same encoder must be saved (using `pickle` or `joblib`) and reused during inference to maintain consistency between training and prediction.

5️⃣Exploratory Data Analysis (EDA)

- Perform univariate and bivariate analysis to understand feature distributions.
- Visualize class imbalance and important feature relationships using histograms, boxplots, and heatmaps.
- Plot feature importances after baseline model fitting for interpretability.

6️⃣Class Imbalance Handling

- Use **RandomOverSampler** or **SMOTE** or other resampling techniques to balance class distribution in the training data

---

7️⃣Feature Selection

- Apply feature importance analysis using Random Forest or correlation-based methods.
- Drop less relevant or low-variance features to simplify the model and improve training efficiency.

8️⃣ Model Building

- Build and evaluate at least **5 different classification models**:
    - Random Forest
    - Decision Tree
    - Logistic Regression
    - K-Nearest Neighbors (KNN)
    - XGBoost
- Compare models using Accuracy, Confusion Matrix, and Classification Report metrics.

🛠️ Hyperparameter Tuning

- Perform hyperparameter tuning (GridSearchCV / **RandomizedSearchCV**) on the best-performing model to optimize accuracy and generalization.

9️⃣ Finalize and Save Best Model

- Choose the best-performing model based on evaluation metrics.
- Save this trained model as a `.pkl` file for deployment. (pickle /joblib)

🔟 Final Task: Build a Streamlit UI

- Design a clean, interactive Streamlit application.
- Allow users to manually input values for all model features through numeric input fields and dropdowns.
- Use the saved model to predict the forest cover type and display the result clearly.
- Inverse transform the target variables while displaying the output.

Technical Tags:

machine learning, classification, random forest, decision tree, logistic regression, xgboost, knn, model evaluation, feature engineering, feature importance, outlier detection, skewness treatment, data imbalance handling, random oversampling, streamlit, model deployment, forest cover type prediction, scikit-learn, imbalanced-learn, exploratory data analysis, eda, matplotlib, seaborn, pkl model saving, numeric input handling, web app development

📦 Project Deliverables

- Well-documented Python scripts or Jupyter notebooks for the complete ML workflow.
- A summary covering data analysis, model selection, evaluation, and insights.
- Visualizations highlighting data trends, feature importance, and model performance.

- The best-performing trained model saved as a `.pkl` file.
- A Streamlit app with user input fields and real-time prediction display.
- Comparison of multiple ML models with accuracy and evaluation metrics.

📑 Project Guidelines

- **Coding Standards:**
  - Use consistent variable naming conventions, clear function names, and proper code comments for readability.
  - Follow Python best practices for structuring scripts and notebooks.
- **Version Control:**
  - Use **Git** for version tracking and collaborative development.
  - Maintain a clean, organized, and well-structured project repository.
- **Testing and Validation:**
  - Validate model performance using appropriate techniques like cross-validation and hold-out sets.
  - Ensure experiment reproducibility by setting random seeds for model runs and data splits.

🗓️ Project Timeline

The complete project must be finished and submitted within **10 days** from the assigned date.

References

Streamlit recording (English)	Special session for STREAMLIT(11/08/2024)
Streamlit Reference doc	Streamlit API reference
Project Live Evaluation	Project Live Evaluation
Capstone Explanation Guideline	Capstone Explanation Guideline
GitHub Reference	How to Use GitHub.pptx
Machine Learning(Eng) Classification and Regression	Project Excellence Series: Guided Lear…
Machine Learning(Tam) Classification and Regression	Project Excellence Series: Guided Lear…
Machine Learning study material	Mastering Supervised Learning Placem…
Project Orientation (Tam)	Project Orientation Session : EcoType: …
Project Orientation (Eng)	Project Orientation Session : EcoType: … Project Orientation Session : EcoType: … Project Orientation Session : EcoType: …

Created By	Verified By	Approved By
Nilofer Mubeen	Shadiya	Nehlath Harmain

PROJECT DOUBT CLARIFICATION SESSION (PROJECT AND CLASS DOUBTS)

About Session: The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.
Note: Book the slot at least before 12:00 Pm on the same day

Timing: Monday-Saturday (3:30PM to 4:30PM)

Booking link :https://forms.gle/XC553oSbMJ2Gcfug9

LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

About Session: The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.

Note: This form will Open only on Saturday (after 2 PM) and Sunday on Every Week

Timing: Monday-Saturday (05:30PM to 07:00PM)

Booking link : https://forms.gle/1m2Gsro41fLtZurRA
