# 1. Data Preparation and Feature Engineering

**Overview**
The data preparation and feature engineering phase is essential for transforming raw data into a form suitable for machine learning models. This step ensures that the data is clean, consistent, and that the features derived from it help improve the predictive performance of the model. Key tasks in this phase include cleaning the data, filling missing values, removing duplicates, and creating new features based on the relationships between existing ones.

### *1.1 Data Collection*

The dataset for this project was sourced from the following platforms:

- **Ministry of Energy and Water (MEW)**: Provides environmental data related to rainfall, river flow, soil moisture, and snowmelt.
- **NASA Earth Data**: Contributes satellite imagery and real-time climate data.
- **United States Geological Survey (USGS)**: Provides river flow and water level records.
- **Meteo Website**: Offers real-time weather data such as temperature and humidity.

These datasets were merged into a unified table for further analysis and processing.

### *1.2 Data Cleaning*

Cleaning the data is a critical step, as raw datasets often contain errors, missing values, or inconsistencies. The following methods were used for data cleaning:

- **Handling Missing Values**: For columns with missing entries, mode imputation was applied for categorical variables like precipitation and soil moisture.
- **Outlier Detection**: Statistical techniques like Z-score and IQR were used to identify and handle outliers.
- **Normalization and Standardization**: The data was normalized to ensure consistency in scale, particularly important for machine learning models that are sensitive to the scale of input features.

### *1.3 Exploratory Data Analysis (EDA)*

EDA was performed to understand the structure and relationships within the dataset:

- **Histograms and Boxplots**: Analyzed the distribution of key variables and detected any potential outliers.
- **Correlation Matrix**: Showed the relationships between environmental factors, highlighting features like precipitation, river discharge, and soil moisture, which are critical for flood Key Insights:

- Strong correlation between precipitation and river discharge.
- Soil moisture and snowmelt also influenced river discharge significantly.
- Specific seasonal trends in rainfall were noted, indicating periods of higher flood risk.

### *1.4 Feature Engineering*

Feature engineering created new variables to improve the model's predictive capabilities:

- **Lagged Features**: Introduced lag variables for precipitation and river discharge to capture temporal dependencies.
- **Rolling Mean Features**: Calculated rolling averages for precipitation and river flow to smooth out daily fluctuations and capture trends.
- **Categorical Features**: Created regional grouping to identify flood risks specific to different geographic areas.
- **Interaction Features**: Developed a "flood risk index" by combining precipitation and soil moisture.

### *1.5 Data Transformation*

To prepare the features for machine learning models, several transformation steps were applied:

- **Normalization**: Features such as precipitation and river discharge were standardized using **Min-Max Scaling** to bring values within a 0-1 range.
- **Encoding**: Categorical variables like region were one-hot encoded to convert them into numerical form.
- **Feature Selection**: Irrelevant or highly correlated features were eliminated using Recursive Feature Elimination (RFE).

---

## 2. Model Exploration

### *2.1 Model Selection*

Two machine learning models were selected:

- **Random Forest**: Chosen for its ability to handle large datasets, model complex relationships, and deal with missing values.
- **Gradient Boosting**: Selected for its effectiveness in imbalanced datasets and potential to improve predictive accuracy by focusing on the hardest-to-predict instances.

### *2.2 Model Training*

The models were trained using the preprocessed dataset. Hyperparameters were tuned through grid search and cross-validation to optimize the model performance:

- **Random Forest**: Hyperparameters like the number of trees (`n_estimators`), tree depth (`max_depth`), and minimum samples per leaf (`min_samples_leaf`) were tuned.
- **Gradient Boosting**: Parameters like the learning rate, maximum depth, and number of estimators were adjusted.

Example code for training the Random Forest model:

```
python
CopyEdit
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV

# Random Forest model initialization
rf_model = RandomForestClassifier()

# Hyperparameter grid for tuning
param_grid = {'n_estimators': [100, 200], 'max_depth': [5, 10],
'min_samples_leaf': [1, 2]}

# Grid search with cross-validation
grid_search = GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=5)
grid_search.fit(X_train, y_train)

# Best parameters and model evaluation
print(grid_search.best_params_)
```

### *2.3 Model Evaluation*

The model's performance was evaluated using the following metrics:

- **Confusion Matrix**: Used to understand the classification performance.
- **ROC Curve**: Analyzed the sensitivity and specificity of the model.
- **Accuracy, Precision, Recall, and F1 Score**: Standard metrics for evaluating classification performance.

Example code for evaluating the model:

```
python
CopyEdit
from sklearn.metrics import confusion_matrix, roc_curve, auc,
classification_report
import matplotlib.pyplot as plt

# Predict on the test set
y_pred = grid_search.best_estimator_.predict(X_test)

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix: \n", cm)

# ROC Curve
fpr, tpr, _ = roc_curve(y_test, y_pred)
roc_auc = auc(fpr, tpr)
plt.plot(fpr, tpr, color='blue', lw=2, label='ROC curve (area = %0.2f)' %
roc_auc)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic')
plt.legend(loc='lower right')
plt.show()

# Classification Report
print(classification_report(y_test, y_pred))
```

---

## 3. Prediction with New Data

Once the model is trained, it can be used to predict flood categories based on new data entries. The key steps for making predictions are:

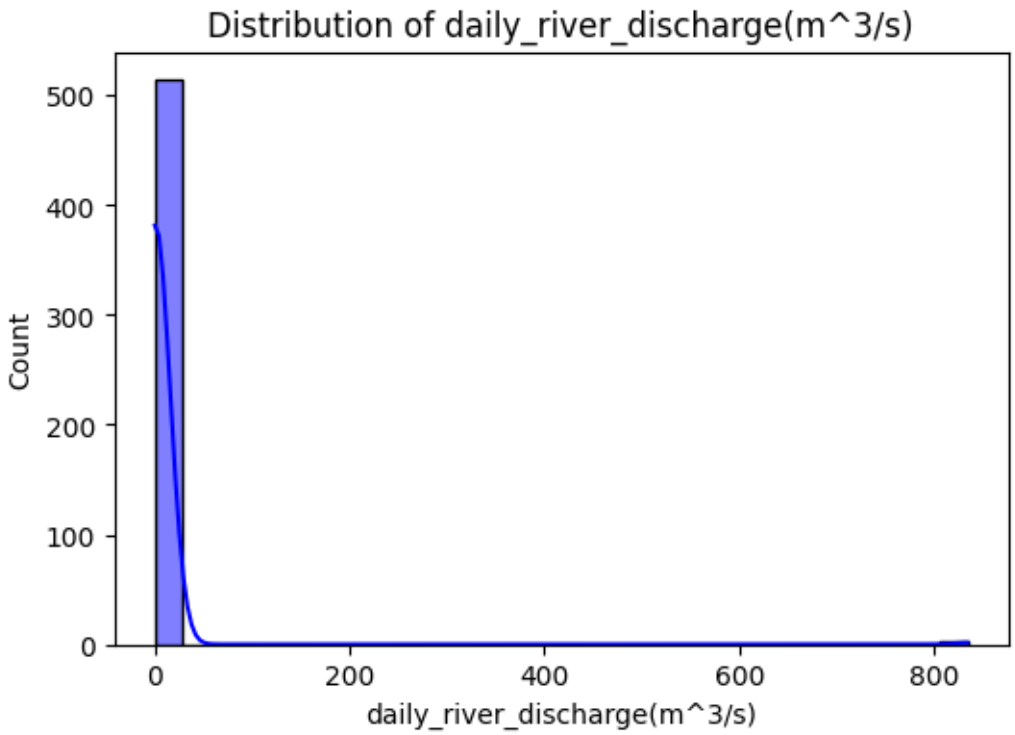
1. **Scale the New Data**: Ensure that any new data is standardized using the same scaler fitted on the training data.
2. **Predict Using the Trained Model**: Use the trained RandomForest model to make predictions on the scaled data.

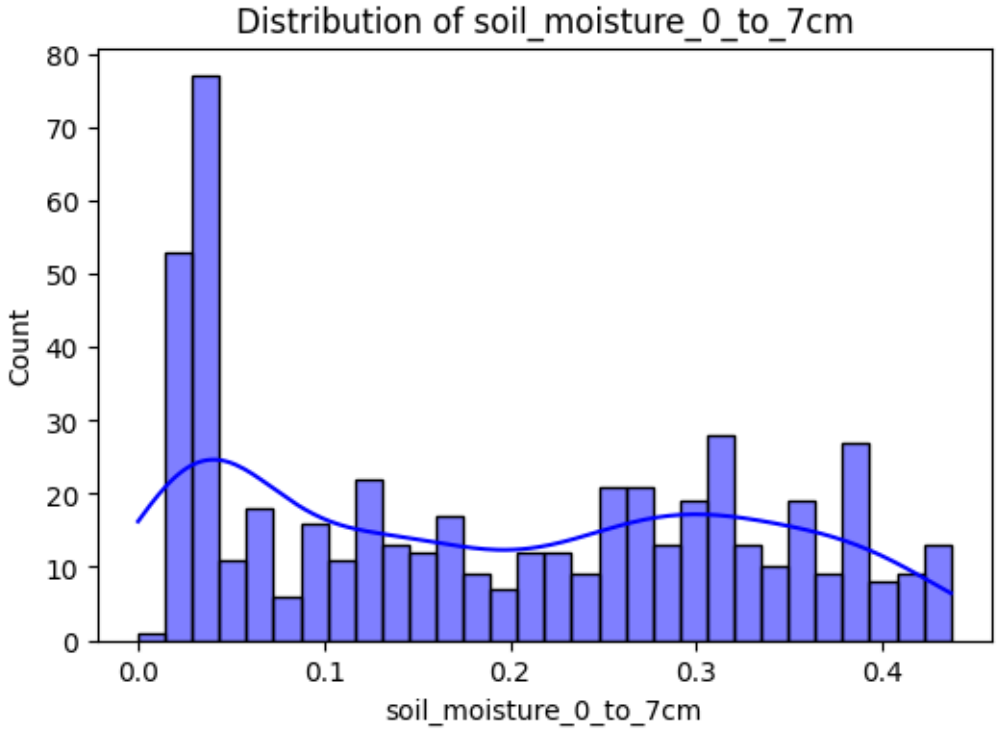
Here's an example for predicting flood categories based on new entries:

```
python
CopyEdit
from sklearn.preprocessing import StandardScaler
import numpy as np

# Example new entry
new_entry = {
    'precipitation': 15.100000,
    'soil_moisture_0_to_7cm': 0.209,
    'daily_river_discharge(m^3/s)': 0.047129,
    'Height in meters': 2.08
}

# Convert to numpy array
new_entry_array = np.array([
    new_entry['precipitation'],
    new_entry['soil_moisture_0_to_7cm'],
    new_entry['daily_river_discharge(m^3/s)'],
    new_entry['Height in meters']
]).reshape(1, -1)

# Standardize the new entry using the same scaler from training
scaler = StandardScaler()
new_entry_scaled = scaler.fit_transform(new_entry_array)

# Predict the flood category using the trained Random Forest model
new_pred = rf_classifier.predict(new_entry_scaled)

# Output the predicted flood category
print(f"The predicted flood category is: {new_pred[0]}")
```

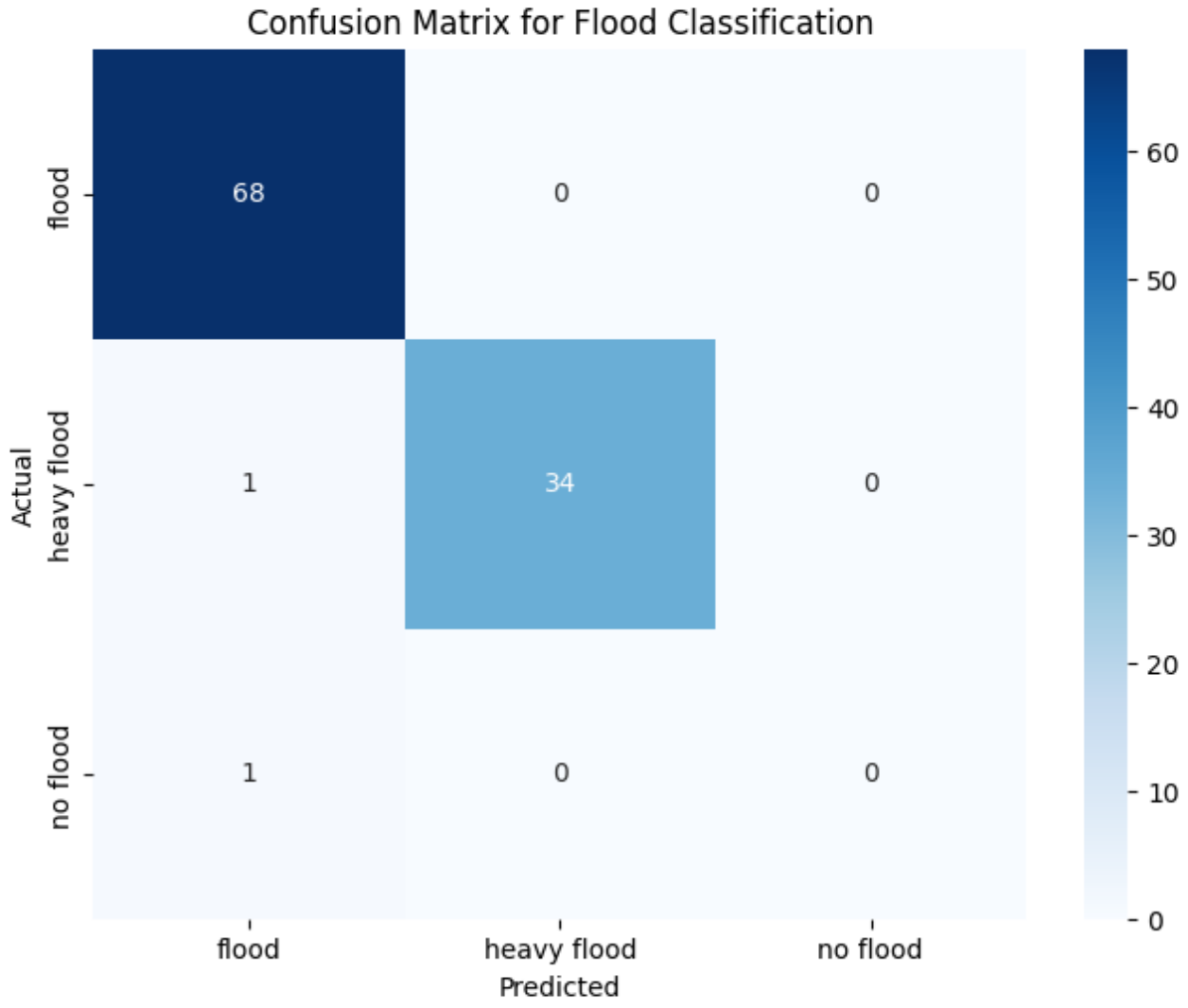
---

## Visualizations


Distribution of daily_river_discharge(m^3/s)
Count
daily_river_discharge(m^3/s)
0
100
200
300
400
500
0
200
400
600
800


Distribution of soil_moisture_0_to_7cm
Count
soil_moisture_0_to_7cm
0
10
20
30
40
50
60
70
80
0.0
0.1
0.2
0.3
0.4


Correlation Heatmap
latitude
longitude
precipitation
soil_moisture_0_to_7cm
daily_river_discharge(m^3/s)
Height in meters
1.00 -0.53 -0.07 0.07 0.06 0.16
-0.53 1.00 0.07 0.06 0.67 0.04
-0.07 0.07 1.00 0.43 -0.05 -0.01
0.07 0.06 0.43 1.00 0.00 -0.02
0.06 0.67 -0.05 0.00 1.00 -0.01
0.16 0.04 -0.01 -0.02 -0.01 1.00
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4


Confusion Matrix for Flood Classification
flood
heavy flood
no flood
Actual
68
0
0
1
34
0
1
0
0
flood
heavy flood
no flood
Predicted
60
50
40
30
20
10
0