Building a Basic Machine Learning Project with Scikit-Learn

In this project, we will use a dataset from GitHub to build a simple machine learning model using the Scikit-Learn library. We will load the data, prepare it for machine learning, train a classifier, and evaluate the performance of the model.

Step 1: Importing Required Libraries

The first step is to import the required libraries for our project. We will be using NumPy and Pandas for data manipulation, and Scikit-Learn for building our machine learning model.

```
import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy_score
```

Step 2: Loading the Dataset

Next, we need to load the dataset that we will be using for our project. For this example, we will be using the Iris Flower Dataset which contains information about different species of iris flowers. The dataset is available on GitHub in CSV format.

```
url = 'https://raw.githubusercontent.com/uiuc-cse/data-fa14/gh-pages/data/iris.csv'

iris_df = pd.read_csv(url)
```

Step 3: Preparing the Data

Before we can build our machine learning model, we need to prepare the data for it. This includes splitting the data into training and testing sets, and separating the target variable from the features.

```
# Separate target variable from features

X = iris_df.iloc[:, :-1].values

y = iris_df.iloc[:, -1].values



# Split data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Step 4: Building the Model

Now that we have prepared our data, we can build our machine learning model. For this example, we will be using a decision tree classifier.

```
# Create a decision tree classifier

clf = DecisionTreeClassifier()



# Fit the classifier to the training data

clf.fit(X_train, y_train)
```

Step 5: Evaluating Model Performance

Finally, we need to evaluate the performance of our machine learning model. For this example, we will be using accuracy as our performance metric.

```
# Make predictions on the test data

y_pred = clf.predict(X_test)



# Calculate accuracy

accuracy = accuracy_score(y_test, y_pred)

print('Accuracy:', accuracy)
```