Machine Learning with Python

Contents

1 Overview

Machine learning is a subset of AI that uses computer algorithms that require feature engineering. It teaches computers to learn from data and identify patterns and use them to make decisions without receiving explicit input from the user. There are three main types of machine learning models:

1. Supervised learning models are trained on a known set of features and a target variable, to identify relationships which are then used for inference or forecasting on new data. For example: linear regression.
2. Unsupervised learning models do not admit labelled feature-target style data, instead they are trained on a set of variables which are all considered features; the model finds relationships between these features. For example: Principal Component Analysis.
3. Reinforcement learning models simulate an AI agent interacting with its environment, they learn how to make decisions based on feedback from the environment. For example:

The two focuses of supervised learning are regression and classification One of the main types of unsupervised learning are clustering

1.1 Machine Learning Lifecycle

1. **Problem Definition:** Clearly state the objective and desired outcome.
2. **Data Collection:** Identify required data and its source.
3. **Data Preparation:** Clean the data, handle missing values, normalize if necessary, engineer features, and perform exploratory data analysis. Split into training and testing sets.
4. **Model Development:** Train the model, tune hyperparameters, and evaluate performance using appropriate metrics.
5. **Deployment:** Integrate the trained model into a production environment.

1.2 Difference between a Data Scientist and AI Engineer

Aspect	Data Science	AI Engineering
Primary Use Cases	Descriptive and predictive analytics (e.g., EDA, clustering, regression, classification)	Prescriptive and generative AI (e.g., optimisation, recommendation systems, intelligent assistants)
Data Type Focus	Primarily structured/tabular data, cleaned and preprocessed	Primarily unstructured data (text, images, audio, video), used at large scale
Model Characteristics	Narrow-scope ML models, smaller in size, domain-specific, faster to train	Foundation models, large-scale, general-purpose, high compute and data requirements
Development Process	Data-driven model development (feature engineering, training, validation)	Application of pre-trained models with prompt engineering, PEFT, and RAG frameworks

Table 1: Key Differences Between Data Science and AI Engineering

1.3 Tools for Machine Learning

Python has several modules that handle the different stages of the machine learning model development pipeline:

1. Data preprocessing: `pysql`, `pandas`
2. Exploratory data analysis: `pandas`, `numpy`, `matplotlib`
3. Optimisation: `scipy`
4. Implementation: `scikit-learn` (supervised and unsupervised methods), `keras`, `pytorch` (deep learing)

1.4 Scikit-learn

The basic syntax for using supervised learning models in `scikit-learn` follows a standard workflow:

1. **Split the data:**

```
x_train, X_test, y_train, y_test = train_test_split(X, y, test_size=...)
```

2. **Import and initialise the model:**

```
from sklearn import svm
model = svm.SVC(...)
```

3. **Train the model:**

```
model.fit(X_train, y_train)
```

4. **Make predictions:**

```
predictions = model.predict(X_test)
```

5. **Evaluate performance:** Use a confusion matrix to assess classification accuracy:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, predictions)
```

6. **Optional – Save the model:** You can serialise the trained model using `pickle`:

```
import pickle
with open("model.pkl", "wb") as f:
    pickle.dump(model, f)
```

Whether this step is necessary depends on the context and industry practice.

2 Linear Regression

Regression is a type of supervised learning model. It models a relationship between a continuous target variable and explanatory features.

2.1 Introduction to Regression

Definition

Linear regression models the relationship between a response variable Y and one or more features $X_1, \ldots, X_p$. In the case of simple linear regression with one predictor X, the model assumes:

$$Y \approx \beta_0 + \beta_1 X$$

where β_0 is the intercept and β_1 is the slope.

Coefficient Estimation

Given training data $(x_1, y_1), \ldots, (x_n, y_n)$, the goal is to estimate the coefficients β_0, β_1 such that the fitted values are as close as possible to the observed values. This is done by minimizing a norm of the residual vector:

$$\beta_{\min} = \arg\min_{\beta \in \mathbb{R}^2} \|\mathbf{Y} - \mathbf{X}\beta\|_p^p$$

For ordinary least squares (OLS), we use the 2-norm ($p = 2$), leading to the minimisation of the residual sum of squares (RSS). The OLS solution yields closed-form expressions for $\hat{\beta}_0$ and $\hat{\beta}_1$. For least absolute deviations (LAD), we take $p = 1$. For Chebyshev or minimax regression, we take $p = \infty$, minimising the maximum residual.

Population vs Sample Regression

In the real world we almost always do not have a maximal data set. This means that we do not have data for every single item in the population, all we can hope to obtain are samples. The fitted regression line from a sample *estimates* the *population regression line*:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

where ε is the irreducible error due to unobserved factors. Since we cannot observe the full population, we fit the model on a sample and obtain estimates $\hat{\beta}_0, \hat{\beta}_1$. However, since we only observe a sample, the estimates $\hat{\beta}$ vary from sample to sample[1]. The sample regression is thus an estimate of the population regression. If we had access to all population data, the OLS minimisation would yield the true β_0, β_1.

Sampling and the Central Limit Theorem

Across repeated samples of size n, we obtain different estimates of β, say $\hat{\beta}^{(1)}, \ldots, \hat{\beta}^{(n)}$. Their average converges to the true coefficient $\bar{\beta}$ as $n \to \infty$, due to the central limit theorem:

$$\hat{\beta} - \bar{\beta} \xrightarrow{d} \mathcal{N}(0, \sigma_*)$$

Standard Error and Inference

The variance of the coefficient estimates across samples is the *standard error*:

$$SE(\hat{\beta}_i) = f(\sigma^2, x_j, \bar{x})$$

where σ^2 is estimated from the data via:

$$\hat{\sigma}^2 = \frac{RSS}{n-2}$$

The standard error quantifies the variability of the estimated coefficient under repeated sampling. If we observe a high standard error then it means that we do not see replicability of the relationship as we vary our sample, which may suggest that the relationship between the features and target is ephemeral and noisy, so any forecast or inference made using it will be unreliable.

[1] The goal of statistical inference is to understand how close $\hat{\beta}$ is to the true β and quantify this uncertainty.