Data Analytics Course - Lesson 06

Ths. Vu Duy Khuong

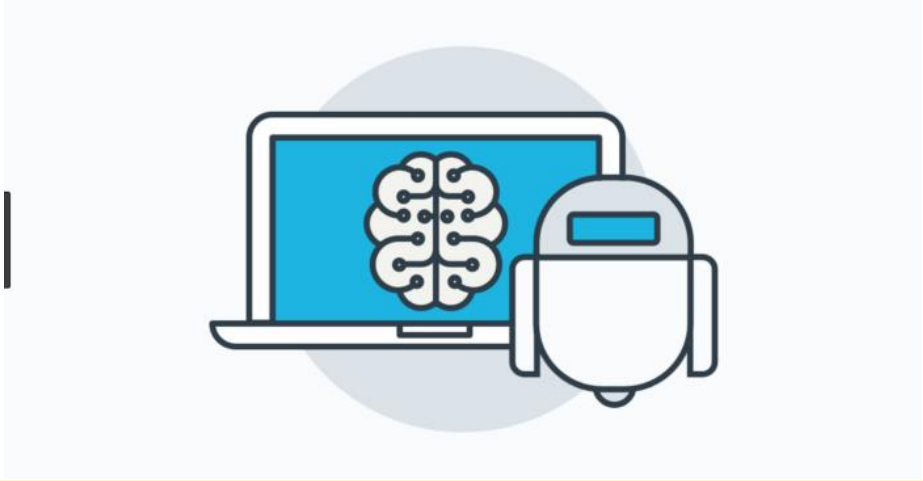
Agenda

- I. Introduction to Machine Learning (ML)
- II. Build ML Models with Scikit-learn Library

I. Introduction to Machine Learning

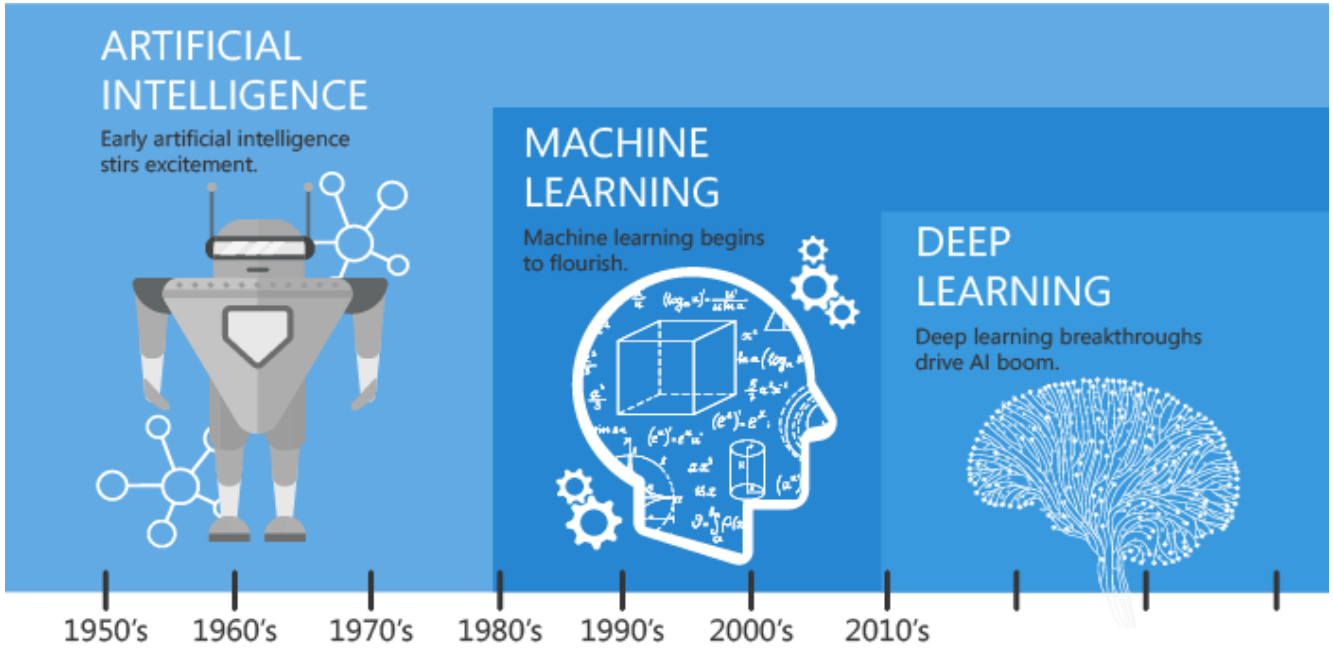
1. What is ML?

Machine learning (ML) is a category of an algorithm that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available.



I. Introduction to Machine Learning

2. A Quick History of Machine Learning



Since an early flush of optimism in the 1950's, smaller subsets of artificial intelligence - first machine learning, then deep learning, a subset of machine learning - have created ever larger disruptions.

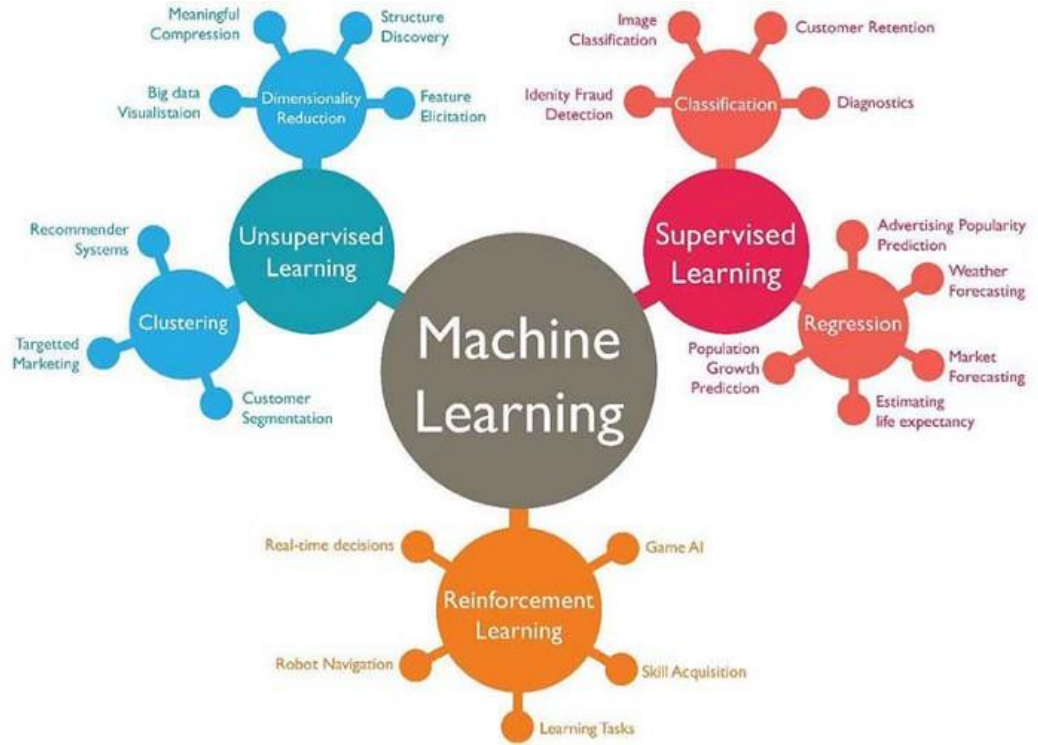
I. Introduction to Machine Learning

3. Types of Machine Learning?

a, Supervised Learning

b, Unsupervised Learning

c, Reinforcement Learning



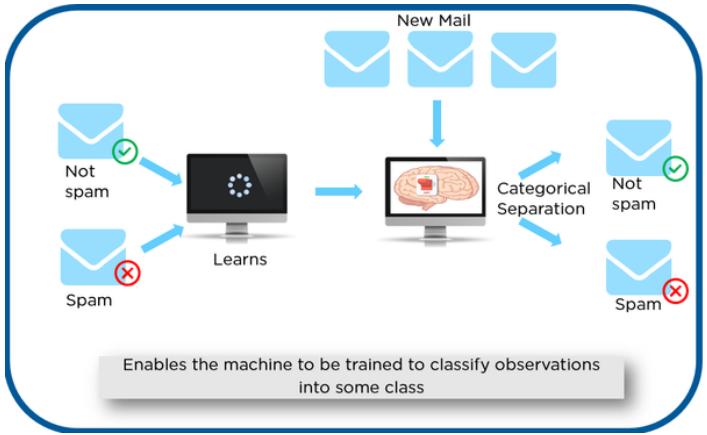
I. Introduction to Machine Learning

3. Types of Machine Learning?

a, Supervised Learning

- In Supervised learning, an AI system is presented with data which is labeled, which means that each data tagged with the correct label.

- The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.



I. Introduction to Machine Learning

3. Types of Machine Learning?

a, Supervised Learning

Types of Supervised learning:

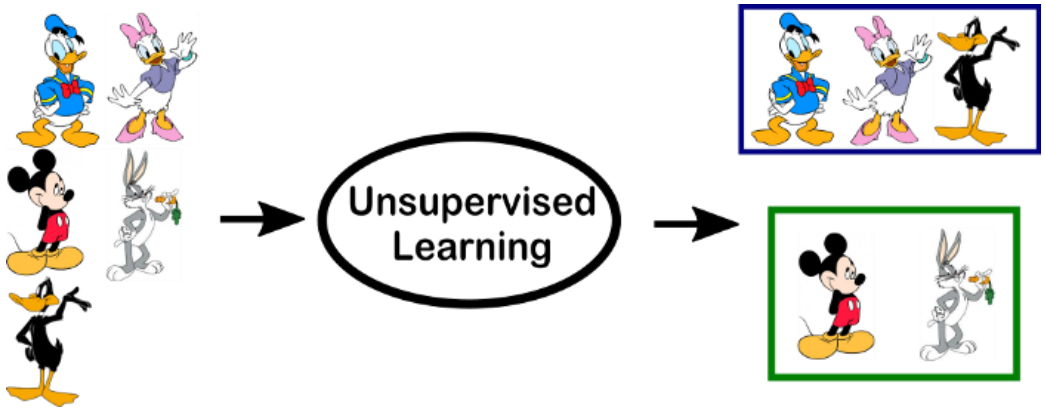
- **Classification**: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”.
- **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

I. Introduction to Machine Learning

3. Types of Machine Learning?

b, Unsupervised Learning

- In unsupervised learning, an AI system is presented with unlabeled, uncategorized data and the system's algorithms act on the data without prior training.
- The output is dependent upon the coded algorithms. Subjecting a system to unsupervised learning is one way of testing AI.



I. Introduction to Machine Learning

3. Types of Machine Learning?

b, Unsupervised Learning

Types of Unsupervised learning:

- **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.
- **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

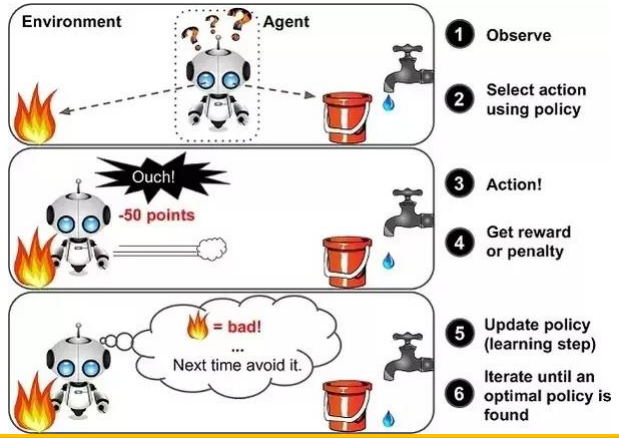
I. Introduction to Machine Learning

3. Types of Machine Learning?

c, Reinforcement Learning

- A reinforcement learning algorithm, or agent, learns by interacting with its environment. The agent receives rewards by performing correctly and penalties for performing incorrectly.

- The agent learns without intervention from a human by maximizing its reward and minimizing its penalty. It is a type of dynamic programming that trains algorithms using a system of reward and punishment.



II. Build ML Models with Scikit-learn

1. Scikit-learn Library

- Scikit-learn is an open source Python library that implements a range of machine learning, preprocessing, cross-validation and visualization algorithms using a unified interface.

II. Build ML Models with Scikit-learn

2. Basic Example

```
>>> from sklearn import neighbors, datasets, preprocessing
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.metrics import accuracy_score

>>> iris = datasets.load_iris()
>>> X, y = iris.data[:, :2], iris.target
>>> X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=33)
>>> scaler = preprocessing.StandardScaler().fit(X_train)
>>> X_train = scaler.transform(X_train)
>>> X_test = scaler.transform(X_test)
>>> knn = neighbors.KNeighborsClassifier(n_neighbors=5)
>>> knn.fit(X_train, y_train) >>> y_pred = knn.predict(X_test)
>>> accuracy_score(y_test, y_pred)
```

II. Build ML Models with Scikit-learn

3. Loading The Data

Your data needs to be numeric and stored as NumPy arrays or SciPy sparse matrices. Other types that are convertible to numeric arrays, such as Pandas DataFrame, are also acceptable.

```
>>> import numpy as np
>>> X = np.random.random((10,5))
>>> y = np.array(['M','M','F','F','M','F','M','M','F','F','F'])
>>> X[X < 0.7] = 0
```

II. Build ML Models with Scikit-learn

4. Training And Test Data

```
>>> from sklearn.model_selection import train_test_split
>>> X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

II. Build ML Models with Scikit-learn

5. Preprocessing The Data

a. Standardization

```
>>> from sklearn.preprocessing import StandardScaler
>>> scaler = StandardScaler().fit(X_train)
>>> standardized_X = scaler.transform(X_train)
>>> standardized_X_test = scaler.transform(X_test)
```

II. Build ML Models with Scikit-learn

5. Preprocessing The Data

b. Normalization

```
>>> from sklearn.preprocessing import Normalizer
>>> scaler = Normalizer().fit(X_train)
>>> normalized_X = scaler.transform(X_train)
>>> normalized_X_test = scaler.transform(X_test)
```

II. Build ML Models with Scikit-learn

5. Preprocessing The Data

c. Encoding Categorical Features

```
>>> from sklearn.preprocessing import LabelEncoder
>>> enc = LabelEncoder()
>>> y = enc.fit_transform(y)
```

II. Build ML Models with Scikit-learn

5. Preprocessing The Data

d. Imputing Missing Values

```
>>> from sklearn.preprocessing import Imputer
>>> imp = Imputer(missing_values=0, strategy='mean', axis=0)
>>> imp.fit_transform(X_train)
```

II. Build ML Models with Scikit-learn

6. Create ML Model

a. Supervised Learning

Linear Regression

```
>>> from sklearn.linear_model import LinearRegression
>>> lr = LinearRegression(normalize=True)
```

Support Vector Machines (SVM)

```
>>> from sklearn.svm import SVC
>>> svc = SVC(kernel='linear')
```

II. Build ML Models with Scikit-learn

6. Create ML Model

a. Supervised Learning

Naive Bayes

```
>>> from sklearn.naive_bayes import GaussianNB
>>> gnb = GaussianNB()
```

KNN

```
>>> from sklearn import neighbors
>>> knn = neighbors.KNeighborsClassifier(n_neighbors=5)
```

II. Build ML Models with Scikit-learn

6. Create ML Model

b. Unsupervised Learning

```
 Principal Component Analysis (PCA)
>>> from sklearn.decomposition import PCA
>>> pca = PCA(n_components=0.95)

K Means
>>> from sklearn.cluster import KMeans
>>> k_means = KMeans(n_clusters=3, random_state=0)
```

II. Build ML Models with Scikit-learn

7. Model Fitting

a. Supervised Learning

```
>>> lr.fit(X, y)
>>> knn.fit(X_train, y_train)
>>> svc.fit(X_train, y_train)
```

II. Build ML Models with Scikit-learn

7. Model Fitting

b. Unsupervised Learning

```
>>> k_means.fit(X_train)
>>> pca_model = pca.fit_transform(X_train)
```

II. Build ML Models with Scikit-learn

8. Prediction

a. Supervised Estimators

```
>>> y_pred = svc.predict(np.random.random((2,5)))
>>> y_pred = lr.predict(X_test)
>>> y_pred = knn.predict_proba(X_test)
```

b. Unsupervised Estimators

```
>>> y_pred = k_means.predict(X_test)
```

II. Build ML Models with Scikit-learn

9. Evaluate Your Model's Performance

a. Classification Metrics

Accuracy Score

```
>>> knn.score(X_test, y_test)
>>> from sklearn.metrics import accuracy_score
>>> accuracy_score(y_test, y_pred)
```

Classification Report

```
>>> from sklearn.metrics import classification_report
>>> print(classification_report(y_test, y_pred))
```

Confusion Matrix

```
>>> from sklearn.metrics import confusion_matrix
>>> print(confusion_matrix(y_test, y_pred))
```

9. Evaluate Your Model's Performance

b. Regression Metrics

Mean Absolute Error

```
>>> from sklearn.metrics import mean_absolute_error
>>> y_true = [3, -0.5, 2]
>>> mean_absolute_error(y_true, y_pred)
```

Mean Squared Error

```
>>> from sklearn.metrics import mean_squared_error
>>> mean_squared_error(y_test, y_pred)
```

R^2 Score

```
>>> from sklearn.metrics import r2_score
>>> r2_score(y_true, y_pred)
```

II. Build ML Models with Scikit-learn

9. Evaluate Your Model's Performance

c. Clustering Metrics

Adjusted Rand Index

```
>>> from sklearn.metrics import adjusted_rand_score
>>> adjusted_rand_score(y_true, y_pred)
```

Homogeneity

```
>>> from sklearn.metrics import homogeneity_score
>>> homogeneity_score(y_true, y_pred)
```

V-measure

```
>>> from sklearn.metrics import v_measure_score
>>> metrics.v_measure_score(y_true, y_pred)
```

III. Reference

Book:

Data Science from Scratch, from chapter 11-17.

Q & A