# Machine Learning with Python [1]

## Vocabulary

**Machine Learning**: Field of study that gives computers the ability to learn without being explicitly programmed [1959, Arthur Samuel]

**Major machine learning techniques**

- **Regression/Estimation:** Predicting continuous values
- **Classification:** Predicting the item class/category of a case
- **Clustering:** Finding the structure of data; summarization
- **Associations:** Associating frequent co-occurring items/events.
- **Anomaly detection:** Discovering abnormal and unusual cases
- **Sequence mining:** Predicting next events; click-stream (Markov Model, HMM)
- **Dimension Reduction:** Reducing the size of data (PCA)
- **Recommendation systems:**

## Python for Machine Learning

NumPy - SciPy - Scikit-learn - Pandas - Matplotlib.

## Comparaison

| Supervised | Unsupervised |
|---|---|
| Classification: Classifies labeled data | Clustering: Finds patterns and groupings from unlabeled data |
| Regression: Predicts trends using previous labeled data | Has fewer evaluation methods than supervised learning |
| Has more evaluation methods than unsupervised learning | Less controlled environment |
| Controlled environment | |

## Regression algorithms

Ordinal regression
Poisson regression
Fast forest quantile regression
Linear, Polynomial, Lasso, Stepwise, Ridge regression
Bayesian linear regression
Neural network regression
Decision forest regression
Boosted decision tree regression
KNN (K-nearest neighbors)

## Simple Linear Regression

$$\hat{y} = \theta_0 + \theta_1 \mathrm{x}_1$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Estimating the parameters

$$\boldsymbol{\theta}_1 = \frac{\sum_{i=1}^{s}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{s}(x_i - \bar{x})^2}$$

$$\boldsymbol{\theta}_0 = \overline{\boldsymbol{y}} - \boldsymbol{\theta}_1 \overline{\boldsymbol{x}}$$

**Model Evaluation**

- Train and Test on the Same Dataset
- Train/Test Split

$$\text{Error } = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

**Training accuracy** is the percentage of correct predictions that the model makes when using the test dataset
High training accuracy isn't necessarily a good thing
Result of over-fitting

- Over-fit: the model is overly trained to the dataset, which may capture noise and produce a non-generalized model

**Out-of-sample accuracy** is the percentage of correct predictions that the model makes on data that the model has not been trained on.
**K-fold cross-validation**

## Metrics in Regression Models

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$
$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$
$$RAE = \frac{\sum_{j=1}^{n}|y_j - \hat{y}_j|}{\sum_{j=1}^{n}|y_j - \bar{y}|}$$
$$RSE = \frac{\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}{\sum_{j=1}^{n}(y_j - \bar{y})^2}$$
$$R^2 = 1 - RSE$$

## Multiple Linear Regression

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n$$
$$\hat{y} = \theta^T X$$
$$\theta^T = [\theta_0, \theta_1, \theta_2, \ldots] \quad X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \ldots \end{bmatrix}$$

## Classification algorithms

Decision Trees (ID3, C4.5, C5.0)
Naive Bayes
Linear Discriminant Analysis
k-Nearest Neighbor
Logistic Regression
Neural Networks
Support Vector Machines (SVM)

## K-Nearest Neighbours

Based on: similar cases with same class labels are near each other.
**K-nearest neighbor algorithm**
1. Pick a value for K.
2. Calculate the distance of unknown case from all cases.
3. Select the K-observations in the training data that are "nearest" to the unknown data point.
4. Predict the response of the unknown data point using the most popular response value from the K-nearest neighbors.

## Evaluation Metrics in Classification

**Jaccard index:**

$y$: Actual labels, $\hat{y}$ : Predicted labels

$J(y, \hat{y}) = \frac{|y \cap \hat{y}|}{|y \cup \hat{y}|} = \frac{|y \cap \hat{y}|}{|y| + |\hat{y}| - |y \cap \hat{y}|}$

**confusion matrix:**

TP: True Positive

FP: False Positive

$$\begin{aligned} \text{Precision} &= \text{TP}/(\text{TP} + \text{FP}) \\ \text{Recall} &= \text{TP}/(\text{TP} + \text{FN}) \end{aligned}$$

$$\text{F1-score} = 2(\text{ prc } * \text{ rec })/(\text{ prc+rec })$$

F1-score high accuracy $\rightarrow 1$

**Log Loss:**

Logarithmic loss measures the performance of a classifier where the predicted output is a probability value between 0 and 1.

$$\text{LogLoss} = -\frac{1}{n} \sum (y \times \log(\hat{y}) + (1 - y) \times \log(1 - \hat{y}))$$

Log Loss high accuracy $\rightarrow 0$

## Decision Trees

Each internal node corresponds to a test.

Each branch corresponds to a result of the test.

Each leaf node assians a classification

**Decision Trees algorithm**

1. Choose an attribute from your dataset.
2. Calculate the significance of attribute in splitting of data.
3. Split data based on the value of the best attribute.
4. Go to step 1.

**Entropy**:Measure of randomness or uncertainty.

The lower the Entropy, the less uniform the distribution, the purer the node.

$$\text{Entropy} = -p(A)\log(p(A)) - p(B)\log(p(B))$$

**Which attribute is the best?**

The tree with the higher Information Gain after splitting.

**Information Gain**: is the information that can increase the level of certainty after splitting.

Information Gain = (Entropy before split) − (weighted entropy after split)

## Logistic Regression

**When is logistic regression suitable?**

If your data is binary.

If you need probabilistic results.

When you need a linear decision boundary.

If you need to understand the impact of a feature.

$$\sigma\left(\theta^T X\right) = \frac{1}{1 + e^{-\theta^T X}}$$

**General cost function**

$$\sigma\left(\theta^T X\right) \longrightarrow P(y = 1 \mid x)$$

Change the weight $->$ Reduce the cost

**Cost function**

$$\text{Cost}(\hat{y}, y) = \begin{cases} -\log(\hat{y}) & \text{if } y = 1 \\ -\log(1 - \hat{y}) & \text{if } y = 0 \end{cases}$$

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} y^i \log\left(\hat{y}^i\right) + \left(1 - y^i\right) \log\left(1 - \hat{y}^i\right)$$

$$\nabla J = \left[\frac{\partial J}{\partial \theta_1}, \frac{\partial J}{\partial \theta_2}, \frac{\partial J}{\partial \theta_3}, \ldots, \frac{\partial J}{\partial \theta_k}\right]$$

$$\theta_{new} = \theta_{prv} - \eta \nabla J$$

**Logistic Regression algorithm**

1. initialize the parameters randomly.
2. Feed the cost function with training set, and calculate the error.
3. Calculate the gradient of cost function.
4. Update weights with new values.
5. Go to step 2 until cost is small enough.

## Support Vector Machine

SVM is a supervised algorithm that classifies cases by finding a separator.

1. Mapping data to a high-dimensional feature space
2. Finding a separator

**Kernelling** in SVM is Mapping data into a higher dimensional space, in such a way that can change a linearly inseparable dataset into a linearly separable dataset. (Linear, Polynomial, RBF, Sigmoid)

find hyperplane:

Find $\mathbf{w}$ and $b$ such that

$\mathbf{\Phi}(\mathbf{w}) = 1/2\mathbf{w}^{\mathrm{T}}\mathbf{w}$ is minimized;

and for all $\{(\mathbf{x_i}, y_i)\} : y_i \left(\mathbf{w}^{\mathrm{T}}\mathbf{x_i} + b\right) \geq 1$

## Support Vector Machine

**Advantages:**

- Accurate in high-dimensional spaces
- Memory efficient

**Disadvantages:**

- Prone to over-fitting
- No probability estimation
- Small datasets

**SVM applications**

- Image recognition
- Text category assignment
- Detecting spam
- Sentiment analysis
- Gene Expression Classification
- Regression, outlier detection and clustering

## Clustering

**Cluster:** A group of objects that are similar to other objects in the cluster, and dissimilar to data points in other clusters.

**Clustering applications**

- RETAIL/MARKETING:
  Identifying buying patterns of customers
  Recommending new books or movies to new customers
- BANKING:
  Fraud detection in credit card use
  Identifying clusters of customers (e.g., loyal)
- INSURANCE:
  Fraud detection in claims analysis
  Insurance risk of customers
- PUBLICATION:
  Auto-categorizing news based on their content
  Recommending similar news articles
- MEDICINE:
  Characterizing patient behavior
- BIOLOGY:
  Clustering genetic markers to identify family ties

## Clustering

**why Clustering?**
- Exploratory data analysis
- Summary generation
- Outlier detection
- Finding duplicates
- Pre-processing step

**Clustering algorithm**

- Partitioned-based Clustering
  Relatively efficient
  E.g. k-Means, k-Median, Fuzzy c-Means
- Hierarchical Clustering
  Produces trees of clusters
  E.g. Agglomerative, Divisive
- Density-based Clustering
  Produces arbitrary shaped clusters
  - E.g. DBSCAN

## k-Means

K-Means is a type of partitioning clustering, that is, it divides the data into K non-overlapping subsets or clusters without any cluster internal structure or labels. This means, it's an unsupervised algorithm.

**objective of k-means**

- To form clusters in such a way that similar samples go into a cluster, and dissimilar samples fall into different clusters.
- To minimize the "intra cluster" distances and maximize the "inter-cluster" distances.
- To divide the data into non-overlapping clusters without any cluster-internal structure

This algorithm is guaranteed to converge to a result, but the result may be a **local optimum** i.e. not necessarily the best possible outcome. To solve this problem, it is common to run the whole process multiple times with different starting conditions. This means with randomized starting centroids, it may give a better outcome.

## k-Means

**objective of k-means**

1. Randomly placing $k$ centroids, one for each cluster.
2. Calculate the distance of each point from each centroid.
3. Assign each data point (object) to its closest centroid, creating a cluster.
4. Recalculate the position of the $k$ centroids.
5. Repeat the steps $2-4$, until the centroids no longer move.

**Elbow method** In cluster analysis, the elbow method is a heuristic used in determining the number of clusters in a data set. The value of the metric as a function of K is plotted and the elbow point is determined where the rate of decrease sharply shifts. It is the right K for clustering.

## Hierarchical Clustering

Hierarchical clustering algorithms build a hierarchy of clusters where each node is a cluster consisting of the clusters of its daughter nodes.
Strategies for hierarchical clustering generally fall into two types:

- **Agglomerative**: This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
- **Divisive**: This is a "top-down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

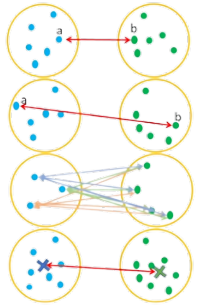
Hierarchical clustering is typically visualized as a dendrogram

## Hierarchical Clustering

**agglomerative algorithm**

1. Create $n$ clusters, one for each data point
2. Compute the Proximity Matrix
3. Repeat
   i. Merge the two closest clusters
   ii. Update the proximity matrix
4. Until only a single cluster remains

- Single Linkage Clustering
  - Minimum distance between clusters
- Complete-Linkage Clustering
  - Maximum distance between clusters
- Average Linkage Clustering
  - Average distance between clusters
- Centroid Linkage Clustering
  - Distance between cluster centroids




| $K$ -means | Hierarchical Clust |
|---|---|
| Much more efficient | Can be slow for large data sets |
| Requires the number of clusters to be specified | Does not require the number of clusters to run |
| Gives only one partitioning of the data based on the predefined number of clusters | Gives more than one partitioning depending on the resolution |
| Potentially returns different clusters each time it is run due to random initialization of centroids | Always generates the same clusters |

## DBSCAN

Density-based clustering locates regions of high density that are separated from one another by regions of low density.

- **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise).
  - Is one of the most common clustering algorithms.
  - Works based on density of objects.
- R (Radius of neighborhood): Radius (R) that if includes enough number of points within, we call it a dense area.
- M (Min number of neighbors):The minimum number of data points we want in a neighborhood to define a cluster

The points of the dataset are separated into 3 types :

- Core points
- border points
- noise points

## Recommender Systems

Two types of recommender systems Content-based: "Show me more of the same of what I've liked before."
Collaborative Filtering: "Tell me what's popular among my neighbors because I might like it too."

- Memory-based
  - Uses the entire user-item dataset to generate a recommendation
  - Uses statistical techniques to approximate users or items e.g., Pearson Correlation, Cosine Similarity, Euclidean Distance, etc.
- Model-based Develops a model of users in an attempt to learn their preferences

**Collaborative filtering**
- User-based collaborative filtering
- Item-based collaborative filtering

## Recommender Systems

Challabges of Collaborative Filtering
**Data Sparsity**: Users in general rate only a limited number of items.
**Cold start**: Difficulty in recommendation to new users or new items.
**Scalability**: Increase in number of users or items.

## Reference

# References

[1] Machine learning with python ibm. `https://www.coursera.org/learn/machine-learning-with-python`.