Classification

Agenda

Key Takeaways-

- Classification, its types and algorithms
- k Nearest Neighbors (kNN)
- Types of distance measures
- Hyperparameter Tuning, GridSearchCV
- Cross validation and its types
- K-fold Cross Validation
- Evaluation measures for Classification

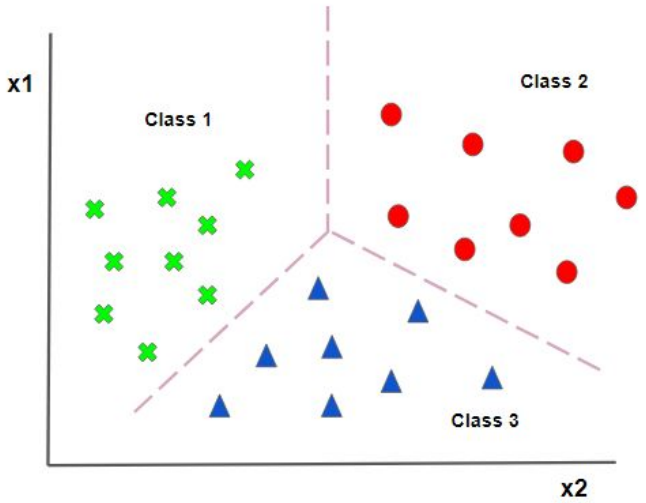
Classification

Classification is a Supervised ML technique that helps in identifying/predicting the class/category for (new) instances.

If the target variable is categorical/qualitative in nature, then we use classification algorithms.

For example, Identifying

- Whether an email is spam or ham.
- Whether a transaction is fraudulent or not
- The handwritten digits 0-9
- Types of debit cards
- Whether a patient is cancerous or non-cancerous



Types of Classification Tasks

Based upon the number of unique classes in the target variable, there are 3 types of classification tasks.

1. Binary Classification : The target variable has only two unique classes and the classification task is to identify/predict one of two classes for each instance.
   E.g. Identifying whether a transaction is fraudulent or not.

2. Multi-class Classification : The target variable has more than two unique classes and the classification task is to identify/predict one of more than two classes for each instance.
   E.g. Handwritten digit recognition

3. Multi-label Classification : Each instance of the labelled data belongs to one or more classes and the classification task is to identify/predict one or more classes for each instance.
   E.g. Predicting tags for Quora questions

Types of Classification Algorithms

The most commonly used classification algorithms are -

1. k Nearest Neighbors (kNN)
2. Decision Tree
3. Random Forest
4. Logistic Regression
5. Support Vector Machine (SVM)

k Nearest Neighbors (kNN)

k-NN algorithm assumes the similarity/distance between the new data point and available data points and put the new data point into a category having maximum votes in the neighborhood.

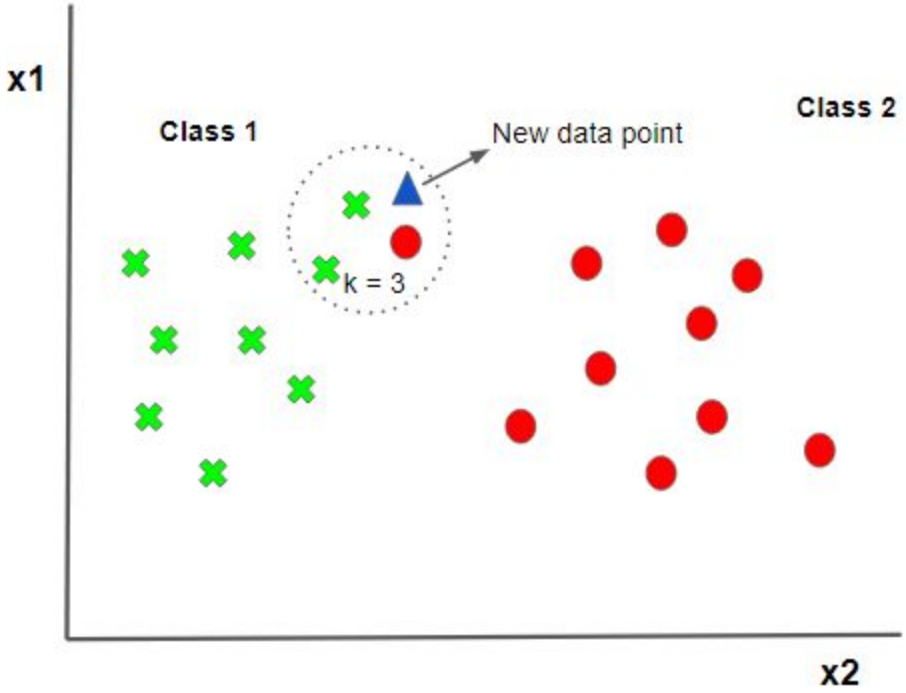
Step 1. Choose the k (number of nearest neighbors).

Step 2. Take the k nearest neighbors of new data point, based on distance (Euclidean)

Step 3. Among the k neighbors, count the number of data points in each category.

Step 4. Assign the new data point to the category with maximum counts.

k Nearest Neighbors (kNN) [Contd.]



Distance measures

The most commonly used distance measures in kNN are -

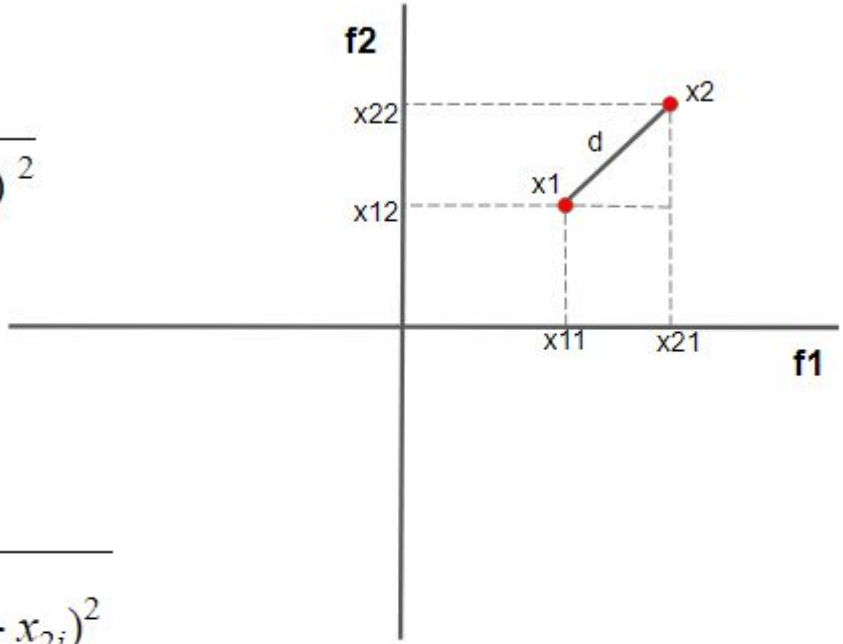
- Euclidean distance
- Manhattan distance
- Minkowski distance
- Cosine distance

Euclidean distance

- Euclidean distance (d)

  =length of shortest line from x1 to x2

$$d = \sqrt{(x_{21} - x_{11})^2 + (x_{22} - x_{21})^2}$$



- Euclidean distance is also represented as L2 norm.

$$d = \|x_1 - x_2\|_2 = \sqrt{\sum_{i=1}^{d} (x_{1i} - x_{2i})^2}$$

Manhattan Distance

- Manhattan distance (d)

  = Sum of absolute differences

$$d = |x_{21} - x_{11}| + |x_{22} - x_{21}|$$

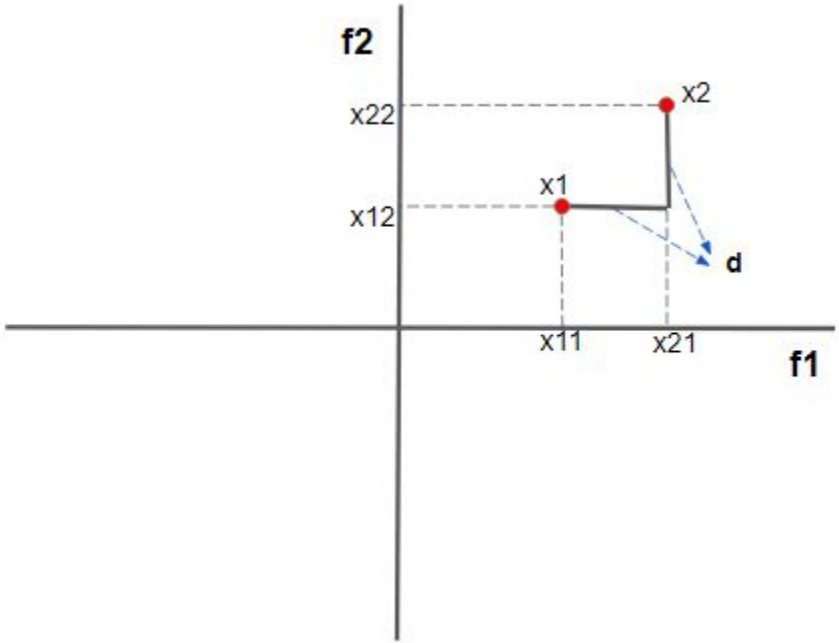
- Its name is from the city “Manhattan”

  Where roads are perpendicular to each other.

- Manhattan distance is also represented as

  L1 norm.

$$d = \|x_1 - x_2\|_1 = \sum_{i=1}^{d} |x_{1i} - x_{2i}|$$



Minkowski Distance

- It is a generalized distance measure.
- Minkowski distance is also represented as Lp norm.

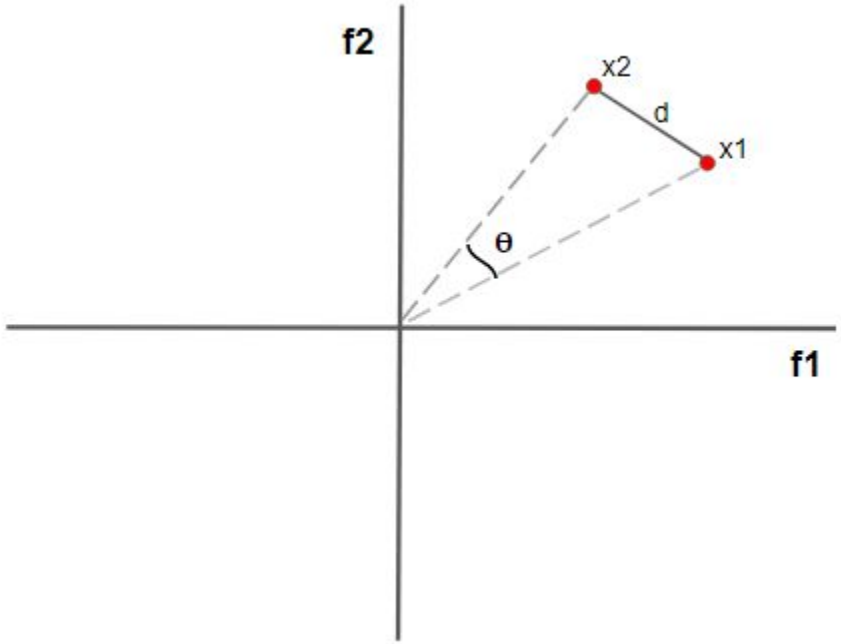
$$\left\| x_1 - x_2 \right\|_p = \left(\sum_{i=1}^{d} \left| x_{1i} - x_{2i} \right|^p \right)^{1/p}$$

- If p = 1, then Minkowski distance = Manhattan distance
- Similarly, if p = 2 then Minkowski distance = Euclidean distance

Cosine Distance

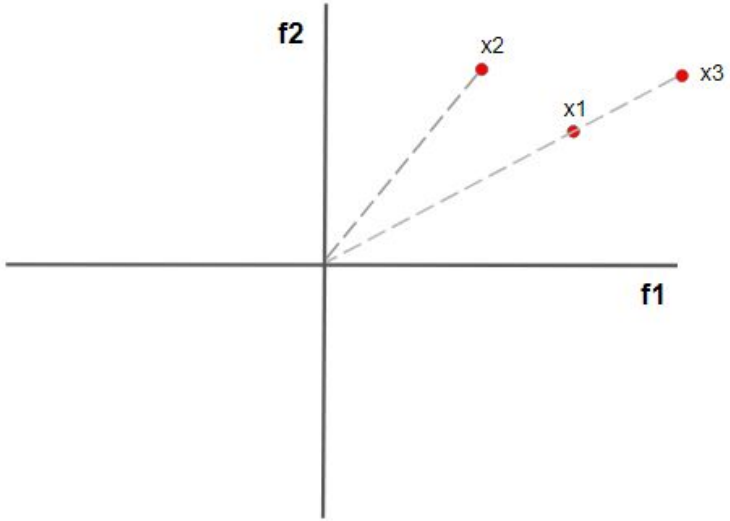
- Cosine distance = (1 - Cosine similarity)

  Where Cosine similarity ≅ Cos Θ and Θ is the angle between two vectors.



Quiz 1

Choose the correct statement(s) as per the following vectors.



- Cosine distance (x1, x3) < Euclidean distance (x1, x3)
- Euclidean distance (x2, x3) > Euclidean distance (x1, x2)
- Cosine distance (x1, x2) != 0
- All of the above

Quiz 2

Choose the correct statement(s).

- As the distance increases the similarity b/w two vectors(entities) also increases.
- As the distance decreases the similarity b/w two vectors(entities) also increases.
- Euclidean distance is similar to L2 norm.
- None of the above

Model Parameter vs Hyperparameter

Parameters	Hyperparameters
It is a configuration variable that is internal to the model	It is a configuration that is external to the model
They are estimated or learned from data (often not set manually)	They are often specified by the practitioner (or often set manually)
They are required by the model when making predictions	They are often tuned for a given predictive modeling problem. This process is known as Hyperparameter tuning.
E.g. Regression coefficients, neural network weights, etc.	E.g. test_size in train-test splitting process, k (# of nearest neighbors) in kNN, Kernel type in SVM, Maximum depth of tree in decision tree, etc.

Cross Validation

- Validation techniques are used to evaluate how well a model would generalize to new/unseen data.

- Cross-Validation is a validation technique used to estimate how well and accurately a predictive model will perform to an independent dataset (real time scenarios).

- Most commonly used cross-validation strategies are -
  - Validation set approach
  - Leave one out cross validation (LOOCV)
  - K-fold cross validation

Cross Validation [Contd.]

1. **Validation set approach**

- This approach divides the dataset into two equal parts, 50% of the dataset is reserved for training purpose, whereas the remaining 50% is reserved for validation purpose.

- **Disadvantage -** This approach generally leads to a high bias model as there always remains a possibility of missing out on relevant and meaningful information due to considering only 50% of the data.

2. **Leave one out cross validation (LOOCV)**

- As the name suggests, here we reserve only one data point for validation purpose and use rest of the data for training purpose.
- In this approach, number of folds = number of data points in the dataset.
- **Disadvantages -** 1. Higher execution time as it is repeated for n times.

2. This approach leads to a high variance model.

K-fold Cross Validation

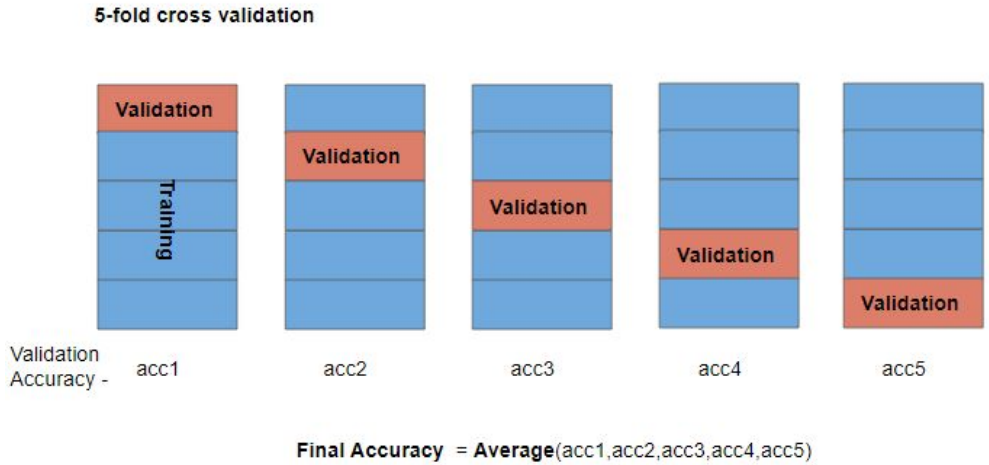
Step 1. Randomly split the entire dataset into **k** folds/subsets.

Step 2. In each iteration(or kth round), train the model using (k – 1) folds of the dataset and validate/test the model using the kth fold.

Step 3. Calculate the accuracy for this iteration.

Step 4. Repeat this process until each of the k-folds has served as the validation/test set.

Step 5. Take the average of all **k** such accuracies to get the final validation accuracy.



Quiz 3

In k-fold cross validation, how many subsets are used for validation purpose in each iteration?

- (k - 1) subsets
- K subsets
- One fixed subset in each iteration
- One randomly selected subset in each iteration

GridSearchCV

- Grid Search is the process of performing hyperparameter tuning in order to determine the optimal values of the hyperparameters for a given model.

- Manually performing hyperparameter tuning -
  - is a time consuming process.
  - Also, it is very hard to keep continuous track of hyperparameters which we have tried and still have to try.

- GridSearchCV is a built-in function in sklearn's model_selection package to perform hyperparameter tuning more efficiently and effectively.

- GridSearchCV tries all the combinations of the values passed and evaluates the estimator(model) for each combination.

- So GridSearchCV is used to find the optimal hyperparameters of a model which results in the most accurate predictions.

Evaluation metrics for Classification

- The most commonly used and simplest evaluation measure for classification is Accuracy.

- Accuracy is the ratio of total number of correctly classified observations to the total observations (total predictions made).

Mathematically,

$$Accuracy = \frac{Number\ of\ correct\ classifications}{Total\ number\ of\ predictions\ made}$$

Caveats - Accuracy gives a false sense of evaluation while dealing with imbalanced data, where most of observation belongs to one class(majority class).

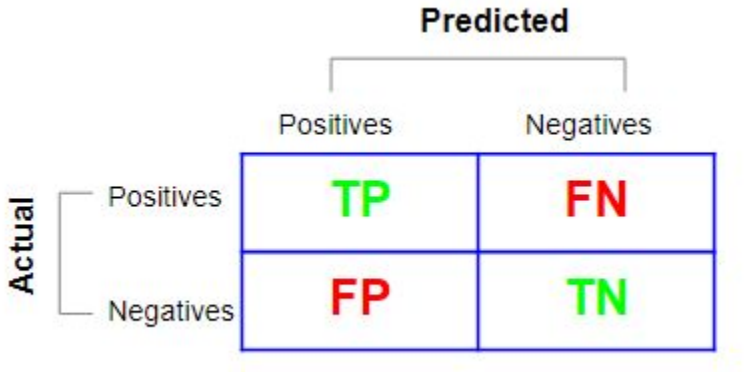
Therefore, we use various other types of evaluation metrics for classification tasks.

- Precision and Recall
- F1-score

Confusion Matrix

- Confusion matrix provides a more intuitive way to count/know the number of correct and incorrect classifications for all the classes.
- Confusion matrix is a NxN matrix, where N is the total number of classes.

For a binary classification problem, the confusion matrix is as shown below.



- **True Positive (TP)** - Actual class is positive and the model also predicted as positive.
- **False Positive (FP) -** Actual class is negative but the model predicted as positive.
- **False Negative (FN) -** Actual class is positive but the model predicted as negative.
- **True Negative (TN)** - Actual class is negative and the model also predicted as negative.

Evaluation measures using Confusion Matrix

$$Accuracy = (TP + TN) / (TP + FP + FN + TN)$$

Precision -

Precision tells how precise(sure) we are about the predictions. In other words, out of total positive predictions how many are actually positive.

$$Precision = TP / (TP + FP)$$

Recall -

Recall is out of total actual positives how many our model predicted as positive.

$$Recall = TP / (TP + FN)$$

F1-score -

F1-score combines both precision and recall. It is a weighted average (harmonic mean especially) of both precision and recall.

$$F1\,score = (2 * Precision * Recall) / (Precision + Recall)$$

Quiz 4

What is the precision for defaulter class based on the below confusion matrix?
Here, the rows (horizontal records) indicate the actual values and the columns (vertical records) indicate the predicted values.

	Defaulter	Non-defaulter
Defaulter	55	8
Non-defaulter	12	25

- 55 / (55+8)
- 55 / (55 +12)
- 12 / (55 + 12)
- 12 / (12 + 25)