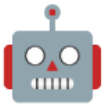

Machine Learning Algorithms

1.Linear Regression :

Linear regression is a type of supervised machine learning algorithm that is used for regression tasks, which involve predicting a continuous numeric value. It is a linear model that uses a set of coefficients to map input data onto an output value.

a. Types of linear regression

There are several different types of linear regression, including simple linear regression, which uses one independent variable to predict the output, and multiple linear regression, which uses multiple independent variables to predict the output.

b. Advantages of Linear Regression

One of the main advantages of linear regression is that it is simple and easy to implement and interpret. It is also relatively efficient, making it well-suited for working with large datasets.

c. Disadvantages of Linear Regression

One of the main disadvantages of linear regression is that it is limited to linear relationships between the input variables and the output, which may not always hold true in real-world data. It is also sensitive to outliers and may not always provide the most accurate predictions.

d. Sample code

```
# Import the necessary modules
from sklearn.linear_model import LinearRegression
from sklearn.datasets import make_regression

# Create the dataset
```

```
X, y = make_regression(n_samples=1000, n_features=1, noise=0.1, random_state=0)

# Create the linear regression model
lr = LinearRegression()

# Train the model on the dataset
lr.fit(X, y)

# Make predictions on new data
predictions = lr.predict(X_new)
```

2.Logistic Regression :

Logistic regression is a type of supervised machine learning algorithm that is used for classification tasks. It is a linear model that uses a set of coefficients to map input data onto a set of output classes or labels.

Unlike linear regression, which is used for predicting continuous numeric values, logistic regression is used for predicting binary or multiclass outcomes. For example, it could be used to predict whether an email is spam or not spam, or to predict which of several classes a given input belongs to.

a. Types of Logistic Regression :

- Binary logistic regression
  - *it has only two possible outcomes (e.g. 0 or 1). Some popular examples of its use include predicting if an e-mail is spam or not spam or if a tumor is malignant or not malignant.*
- Multinomial logistic regression
  - T*he dependent variable has three or more possible outcomes.*
  - *movie studios want to predict what genre of film a moviegoer is likely to see to market films more effectively. A multinomial logistic regression model can help the studio to determine the strength of influence a person's age, gender, and dating status may have on the type of film that they prefer.*
- Ordinal logistic regression
  - *This type of logistic regression model is leveraged when the response variable has three or more possible outcome, but in this case, these values do have a defined order. Examples of ordinal responses include grading scales from A to F or rating scales from 1 to 5.*

b. Use case :

- Fraud Detection

- Disease Prediction
- Churn Prediction

```
# Load Libraries
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression

#Feature and label splitting
X, y = load_iris(return_X_y=True)

#Model building
clf = LogisticRegression(random_state=0).fit(X, y)
clf.predict(X[:2, :])
Output : array([0, 0])

#Model prediction
clf.predict_proba(X[:2, :])
Output :array([[9.8...e-01, 1.8...e-02, 1.4...e-08],
       [9.7...e-01, 2.8...e-02, ...e-08]])

#Model Evaluation
clf.score(X, y)
Output: 0.97...
```

3.Decision Tree

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

1. Types of Decision Tree :

- ID3
- C4.5
- CART
-

2. How to choose the best attribute at each node

   Entropy values can fall between 0 and 1. If all samples in data set, S, belong to one class, then entropy will equal zero. If half of the samples are classified as one class and the other half are in another class, entropy will be at its highest at 1. In order to select the best feature to split on and find the optimal decision tree, the attribute with the smallest amount of entropy should be used. Information gain represents the difference in entropy before and after a split on a given attribute. The attribute with

the highest information gain will produce the best split as it's doing the best job at classifying the training data according to its target classification.

3. Advantages of Decision Tree

   - Easy to interpret :

     The Boolean logic and visual representations of decision trees make them easier to understand and consume. The hierarchical nature of a decision tree also makes it easy to see which attributes are most important, which isn't always clear with other algorithms, like neural networks.

   - Little to no data preparation required :

     Decision trees have a number of characteristics, which make it more flexible than other classifiers. It can handle various data types—i.e. discrete or continuous values, and continuous values can be converted into categorical values through the use of thresholds. Additionally, it can also handle values with missing values, which can be problematic for other classifiers, like Naïve Bayes

   - More flexible :

     Decision trees can be leveraged for both classification and regression tasks, making it more flexible than some other algorithms. It's also insensitive to underlying relationships between attributes; this means that if two variables are highly correlated, the algorithm will only choose one of the features to split on.

4. Disadvantages of Decision Tree

   1.Prone to overfitting :

   Complex decision trees tend to overfit and do not generalize well to new data. This scenario can be avoided through the processes of pre-pruning or post-pruning. Pre-pruning halts tree growth when there is insufficient data while post-pruning removes subtrees with inadequate data after tree construction.

   2.High variance estimators :

   Small variations within data can produce a very different decision tree. Bagging ,or the averaging of estimates, can be a method of reducing variance of decision trees. However, this approach is limited as it can lead to highly correlated predictors.

   3.More Costly :

Given that decision trees take a greedy search approach during construction, they can be more expensive to train compared to other algorithms.

4.Not fully supported in scikit-learn :

Scikit-learn is a popular machine learning library based in Python. While this library does have a Decision Tree Module.
(Decision Tree Classifier , link resides outside of ibm.com), the current implementation does not support categorical variables.

5. Sample code

```
# Import the necessary modules
from sklearn.tree import DecisionTreeClassifier
from sklearn.datasets import make_classification

# Create the dataset
X, y = make_classification(n_samples=1000, n_features=10, n_classes=2, random_state=0)

# Create the decision tree classifier
clf = DecisionTreeClassifier(random_state=0)

# Train the classifier on the dataset
clf.fit(X, y)

# Make predictions on new data
predictions = clf.predict(X_new)
```

4.Random Forest

Random forest is a type of supervised machine learning algorithm that can be used for both classification and regression tasks. It is an ensemble model, meaning that it is composed of multiple individual decision trees that work together to make predictions.

1.Types of random forest

There are two main types of random forests: classification forests, which are used for classification tasks, and regression forests, which are used for regression tasks.

2.Advantages of random forest

One of the main advantages of random forests is that they are very accurate and can handle large datasets with high dimensional data. They are also relatively

easy to train and interpret, and can handle missing or incomplete data without requiring data preprocessing.

3.Disadvantages of random forest

One of the main disadvantages of random forests is that they can be slow to make predictions, especially when working with large datasets. They also tend to overfit the training data, meaning that they may not generalize well to unseen data.

4.Sample code

```
# Import the necessary modules
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification

# Create the dataset
X, y = make_classification(n_samples=1000, n_features=10, n_classes=2, random_state=0)

# Create the random forest classifier
clf = RandomForestClassifier(n_estimators=100, random_state=0)

# Train the classifier on the dataset
clf.fit(X, y)

# Make predictions on new data
predictions = clf.predict(X_new)
```

5. Support Vector Machines

A support vector machine (SVM) is a type of supervised machine learning algorithm that can be used for classification and regression tasks. It is a linear model that uses a set of mathematical functions to map input data onto a set of output classes or values.

1.Types of SVM ?

There are two main types of SVMs: linear SVMs, which are used for classification tasks with linearly separable data, and non-linear SVMs, which are used for classification tasks with non-linearly separable data.

2.Advantages of SVM

One of the main advantages of SVMs is that they are very accurate and can handle large datasets with high dimensional data. They are also relatively easy to train and interpret, and can handle missing or incomplete data without requiring data preprocessing.

3.Disadvantage of SVM

One of the main disadvantages of SVMs is that they can be slow to make predictions, especially when working with large datasets. They also tend to be sensitive to noise and may overfit the training data if not properly regularized.

4.Sample code

```
# Import the necessary modules
from sklearn.svm import LinearSVC
from sklearn.datasets import make_classification

# Create the dataset
X, y = make_classification(n_samples=1000, n_features=10, n_classes=2, random_state=0)

# Create the linear SVM classifier
clf = LinearSVC(random_state=0)

# Train the classifier on the dataset
clf.fit(X, y)

# Make predictions on new data
predictions = clf.predict(X_new)
```

6. K Means Clustering

K-means clustering is a type of unsupervised machine learning algorithm that is used to classify data into a specified number of clusters based on their similarity. It is a simple and efficient algorithm that uses a random initialization method to determine the final clustering of the data.

1.Different types of k means clustering

There are no different types of K-means clustering, but there are different variations and extensions of the algorithm that can be used in different situations.

2.Advantages of k means clustering

One of the main advantages of K-means clustering is that it is fast and efficient, making it well-suited for working with large datasets. It is also relatively easy to implement and interpret, and can be used to identify clusters in complex data.

3.Disadvantages of k means clustering

One of the main disadvantages of K-means clustering is that it can be sensitive to the initial random seed and may not always converge to the global optimum solution. It is also limited to linear cluster boundaries and may not be able to handle more complex data structures.

4.Sample code

```
# Import the necessary modules
from sklearn.cluster import KMeans
from sklearn.datasets import make_blobs

# Create the dataset
X, y = make_blobs(n_samples=1000, n_features=10, centers=5, random_state=0)

# Create the K-means clustering model
kmeans = KMeans(n_clusters=5, random_state=0)

# Train the model on the dataset
kmeans.fit(X)

# Make predictions on new data
predictions = kmeans.predict(X_new)
```

7.Naive Bayes

Naive Bayes is a type of supervised machine learning algorithm that is used for classification tasks. It is a probabilistic model that uses Bayes' theorem to make predictions based on the relationship between different features and the target variable.

1.DIfferent types of Naive bayes

There are several different types of Naive Bayes algorithms, including Gaussian naive Bayes, which is used for continuous data, and Bernoulli Naive Bayes, which is used for binary data.

2.Advantages of Naive bayes

One of the main advantages of Naive Bayes is that it is very simple and efficient, making it well-suited for working with large datasets. It is also relatively easy to implement and interpret, and can handle missing or incomplete data without requiring data preprocessing.

3.Disadvantages of Naive bayes

One of the main disadvantages of Naive Bayes is that it is based on the assumption of independence between features, which may not always hold true in real-world data. This can lead to inaccurate predictions and reduced model performance.

4.Sample code

```
# Import the necessary modules
from sklearn.naive_bayes import GaussianNB
from sklearn.datasets import make_classification

# Create the dataset
X, y = make_classification(n_samples=1000, n_features=10, n_classes=2, random_state=0)

# Create the Gaussian naive Bayes classifier
gnb = GaussianNB()

# Train the classifier on the dataset
gnb.fit(X, y)

# Make predictions on new data
predictions = gnb.predict(X_new)
```