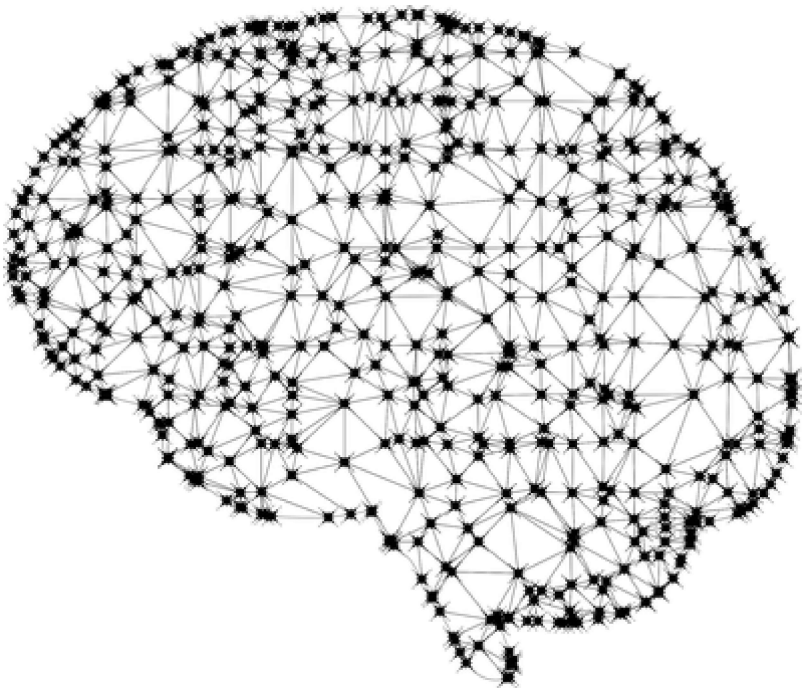
Regression Versus Classification Machine Learning: What's the Difference?

Dr. Michael J. Garbade Follow

Aug 11, 2018 · 4 min read



The difference between regression machine learning algorithms and classification machine learning algorithms sometimes confuse most data scientists, which make them to implement wrong methodologies in solving their prediction problems.

Andreybu, who is from Germany and has more than 5 years of machine learning experience, says that "understanding whether the machine learning task is a regression or classification problem is key for selecting the right algorithm to use."

Let's start by talking about the similarities between the two techniques.

Supervised machine learning

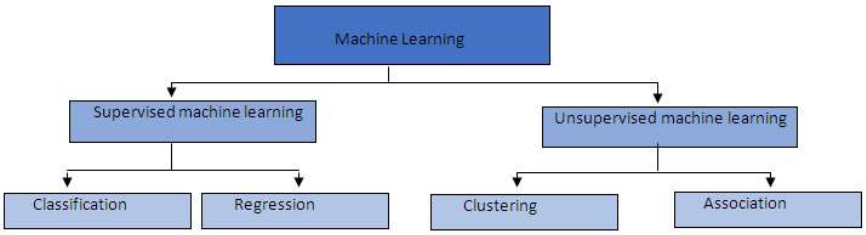
Regression and classification are categorized under the same umbrella of supervised machine learning. Both share the same concept of

utilizing known datasets (referred to as training datasets) to make predictions.

In supervised learning, an algorithm is employed to learn the mapping function from the input variable (x) to the output variable (y); that is y = f(X).

The objective of such a problem is to approximate the mapping function (f) as accurately as possible such that whenever there is a new input data (x), the output variable (y) for the dataset can be predicted.

Here is a chart that shows the different groupings of machine learning:



Unfortunately, there is where the similarity between regression versus classification machine learning ends.

The main difference between them is that the output variable in regression is numerical (or continuous) while that for classification is categorical (or discrete).

Regression in machine learning

In machine learning, regression algorithms attempt to estimate the mapping function (f) from the input variables (x) to numerical or continuous output variables (y).

In this case, y is a real value, which can be an integer or a floating point value. Therefore, regression prediction problems are usually quantities or sizes.

For example, when provided with a dataset about houses, and you are asked to predict their prices, that is a regression task because price will be a continuous output.

Examples of the common regression algorithms include linear regression, Support Vector Regression (SVR), and regression trees.

Some algorithms, such as logistic regression, have the name "regression" in their names but they are not regression algorithms.

Here is an example of a linear regression problem in Python:

```
import numpy as np

import pandas as pd

# importing the model

from sklearn.linear_model import LinearRegression

from sklearn.cross_validation import train_test_split

# importing the module for calculating the performance
metrics of the model

from sklearn import metrics

data_path = "http://www-
bcf.usc.edu/~gareth/ISL/Advertising.csv" # loading the
advertising dataset

data = pd.read_csv(data_path, index_col=0)

array_items = ['TV', 'radio', 'newspaper'] #creating an
array list of the items

X = data[array_items] #choosing a subset of the dataset

y = data.sales #sales

# dividing X and y into training and testing units

X_train, X_test, y_train, y_test = train_test_split(X, y,
random_state=1)

linearreg = LinearRegression() #applying the linear
regression model

linearreg.fit(X_train, y_train) #fitting the model to the
training data

y_predict = linearreg.predict(X_test) #making predictions
based on the testing unit

print(np.sqrt(metrics.mean_squared_error(y_test,
y_predict))) #calculating the RMSE number
```

```
#output gives the RMSE number as 1.4046514230328955
```

Classification in machine learning

On the other hand, classification algorithms attempt to estimate the mapping function (f) from the input variables (x) to discrete or categorical output variables (y).

In this case, y is a category that the mapping function predicts. If provided with a single or several input variables, a classification model will attempt to predict the value of a single or several conclusions.

For example, when provided with a dataset about houses, a classification algorithm can try to predict whether the prices for the houses "sell more or less than the recommended retail price."

Here, the houses will be classified whether their prices fall into two discrete categories: above or below the said price.

Examples of the common classification algorithms include logistic regression, Naïve Bayes, decision trees, and K Nearest Neighbors.

Here is an example of a classification problem that differentiates between an orange and an apple:

```
from sklearn import tree

# Gathering training data

# features = [[155, "rough"], [180, "rough"], [135,
"smooth"], [110, "smooth"]] # Input to classifier

features = [[155, 0], [180, 0], [135, 1], [110, 1]] #
scikit-learn requires real-valued features

# labels = ["orange", "orange", "apple", "apple"] # output
values

labels = [1, 1, 0, 0]

# Training classifier

classifier = tree.DecisionTreeClassifier() # using decision
tree classifier
```

```
classifier = classifier.fit(features, labels) # Find
patterns in data

# Making predictions

print (classifier.predict([[120, 1]]))

# Output is 0 for apple
```

Wrapping up

Selecting the correct algorithm for your machine learning problem is critical for the realization of the results you need.

As a data scientist, you need to know how to differentiate between regression predictive models and classification predictive models so that you can choose the best one for your specific use case.

Do you have any comments or questions?

Please post them below.