<https://www.javatpoint.com/regression-vs-classification-in-machine-learning#:~:text=Difference%20between%20Regression%20and%20Classification%20%20%20,the%20de%20...%20%202%20more%20rows%20>

Regression vs. Classification

Regression and Classification algorithms are Supervised Learning algorithms. Both the algorithms are used for prediction in Machine learning and work with the labeled datasets. But the difference between both is how they are used for different machine learning problems.

The main difference between Regression and Classification algorithms that Regression algorithms are used to **predict the continuous** values such as price, salary, age, etc. and Classification algorithms are used to **predict/Classify the discrete values** such as Male or Female, True or False, Spam or Not Spam, etc.

## **Classification:**

Classification is a process of finding a function which helps in dividing the dataset into classes based on different parameters. In Classification, a computer program is trained on the training dataset and based on that training, it categorizes the data into different classes.

The task of the classification algorithm is to find the mapping function to map the input(x) to the discrete output(y).

**Example:** The best example to understand the Classification problem is Email Spam Detection. The model is trained on the basis of millions of emails on different parameters, and whenever it receives a new email, it identifies whether the email is spam or not. If the email is spam, then it is moved to the Spam folder.

**Types of ML Classification Algorithms:**

Classification Algorithms can be further divided into the following types:

* Logistic Regression
* K-Nearest Neighbours
* Support Vector Machines
* Kernel SVM
* Naïve Bayes
* Decision Tree Classification
* Random Forest Classification

**Regression:**

Regression is a process of finding the correlations between dependent and independent variables. It helps in predicting the continuous variables such as prediction of **Market Trends**, prediction of House prices, etc.

The task of the Regression algorithm is to find the mapping function to map the input variable(x) to the continuous output variable(y).

**Example:** Suppose we want to do weather forecasting, so for this, we will use the Regression algorithm. In weather prediction, the model is trained on the past data, and once the training is completed, it can easily predict the weather for future days.

**Types of Regression Algorithm:**

* Simple Linear Regression
* Multiple Linear Regression
* Polynomial Regression
* Support Vector Regression
* Decision Tree Regression
* Random Forest Regression

<https://www.analyticsvidhya.com/blog/2020/08/bias-and-variance-tradeoff-machine-learning/>

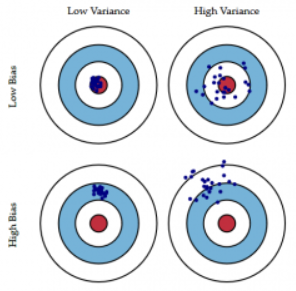
## What is Bias?

Bias is the difference between the Predicted Value and the Expected Value. When there is a high bias error, it results in a very simplistic model that does not consider the variations very well. Since it does not learn the training data very well, it is called **Underfitting.**

## What is a Variance?

Contrary to bias, the Variance is when the model takes into account the fluctuations in the data i.e. the noise as well. Since in the case of high variance, the model learns too much from the training data, it is called **overfitting.**

The balance between the Bias error and the Variance error is the **Bias-Variance Tradeoff**. The following bulls-eye diagram explains the tradeoff better:



## Conclusion

To summarize, in this article, we learned that an ideal model would be one where both the bias error and the variance error are low. However, we should always aim for a model where the model score for the training data is as close as possible to the model score for the testing data.

That’s where we figured out how to choose a model that is not too complex (High variance and low bias) which would lead to overfitting and nor too simple(High Bias and low variance) which would lead to underfitting.

Bias and Variance plays an important role in deciding which predictive model to use. I hope this article explained the concept well.

<https://arifromadhan19.medium.com/part-1-regression-and-classification-model-evaluation-bc7f6ab3b4dd>

Part 1: Regression and Classification Model Evaluation

Regression

* R-Squared – It is also known as the **coefficient of determination**. This metric gives an indication of how good a model fits a given dataset. It indicates how close the [regression line](https://www.studytonight.com/post/classification-problem-introduction-to-logistic-regression) (i.e the predicted values plotted) is to the actual data values. The **R squared value lies between 0 and 1** where 0 indicates that this model doesn't fit the given data and 1 indicates that the model fits perfectly to the dataset provided.

import numpy as np

X = np.random.randn(100)

y = np.random.randn(60) # y has nothing to do with X whatsoever

from sklearn.linear\_model import LinearRegression

from sklearn.cross\_validation import cross\_val\_score

scores = cross\_val\_score(LinearRegression(), X, y,scoring='r2')

* RMSE : Root Mean Squared Error [0- infinity] - RMSE is the standard deviation of the errors which occur when a prediction is made on a dataset. This is the same as MSE (Mean Squared Error) but the root of the value is considered while determining the accuracy of the model.

from sklearn.metrics import mean\_squared\_error

from math import sqrt

actual\_values = [3, -0.5, 2, 7]

predicted\_values = [2.5, 0.0, 2, 8]

mean\_squared\_error(actual\_values, predicted\_values)

# taking root of mean squared error

root\_mean\_squared\_error = sqrt(mean\_squared\_error)

* Mean Squared Error or MSE - MSE is calculated by taking the average of the square of the difference between the original and predicted values of the data.

from sklearn.metrics import mean\_squared\_error

actual\_values = [3, -0.5, 2, 7]

predicted\_values = [2.5, 0.0, 2, 8]

mean\_squared\_error(actual\_values, predicted\_values)

* MAE : Mean Absolute Error [0- infinity] – We know that an error basically is the absolute difference between the actual or true values and the values that are predicted. Absolute difference means that if the result has a negative sign, it is ignored. Hence, **MAE = True values – Predicted values**

from sklearn.metrics import mean\_absolute\_error

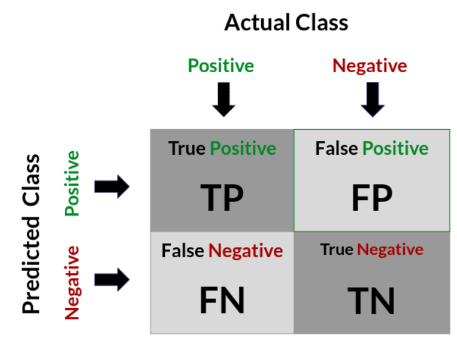
# predicting home prices in some area

predicted\_home\_prices = mycity\_model.predict(X)

mean\_absolute\_error(y, predicted\_home\_prices)

Classification

Before discussing model evaluation on classification, we should understand the confusion matrix. A confusion matrix is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.



Positive (P): Observation is positive (for example: is an apple).

Negative (N): Observation is not positive (for example: is not an apple).

True Positive (TP): Observation is positive, and is predicted to be positive.

False Negative (FN): Observation is positive, but is predicted negative.

True Negative (TN): Observation is negative, and is predicted to be negative.

False Positive (FP): Observation is negative, but is predicted positive.

* Accuracy

Accuracy = (TP + TN) / (TP + TN + FP + FN)  
Accuracy = (True positive+ True negative) /Total count of elements

* Precision

Precision = (TP) / (TP + FP)  
Precision = (True positive) / (True positive + False positive)

* Recall

Recall = (TP) / (TP + FN)  
Recall = (True positive) / (True positive + False negative)

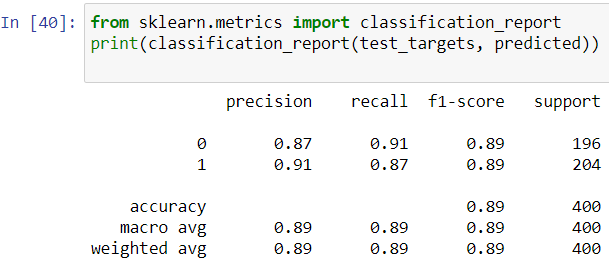
* Specificity

Specificity = (TN) / (TN + FP)  
Specificity = (True negative) / (True negative + False positive)

* F1 Score

F1 Score measure provides a way to combine both precision and recall into a single measure that captures both properties.

F1 Score = 2 \* (Recall\*Precision) / (Recall + Precision)



Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures (series / dataframes)

<https://www.tutorialspoint.com/python_pandas/python_pandas_series.htm>

<https://www.tutorialspoint.com/python_pandas/python_pandas_dataframe.htm>

# Python Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data.

<https://www.geeksforgeeks.org/python-numpy/>

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays.

<https://www.tutorialspoint.com/matplotlib/index.htm>

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of **[matplotlib](https://www.geeksforgeeks.org/python-introduction-matplotlib/" \t "_blank)** library and also closely integrated to the data structures from [pandas](https://www.geeksforgeeks.org/introduction-to-pandas-in-python/).

<https://www.geeksforgeeks.org/python-seaborn-lmplot-method/>

Apache Spark is written in Scala programming language. To support Python with Spark, Apache Spark community released a tool, PySpark. Using PySpark, you can work with RDDs in Python programming language also. It is because of a library called Py4j that they are able to achieve this. This is an introductory tutorial, which covers the basics of Data-Driven Documents and explains how to deal with its various components and sub-components.

<https://www.tutorialspoint.com/pyspark/index.htm>

Neo4j is the world's leading open source Graph Database which is developed using Java technology. It is highly scalable and schema free (NoSQL). A graph is a pictorial representation of a set of objects where some pairs of objects are connected by links. It is composed of two elements - nodes (vertices) and relationships (edges). Graph database is a database used to **model the data in the form of graph. In here, the nodes of a graph depict the entities while the relationships depict the association of these nodes**.

Neo4j is a popular Graph Database. Other Graph Databases are Oracle NoSQL Database, OrientDB, HypherGraphDB, GraphBase, InfiniteGraph, and AllegroGraph.

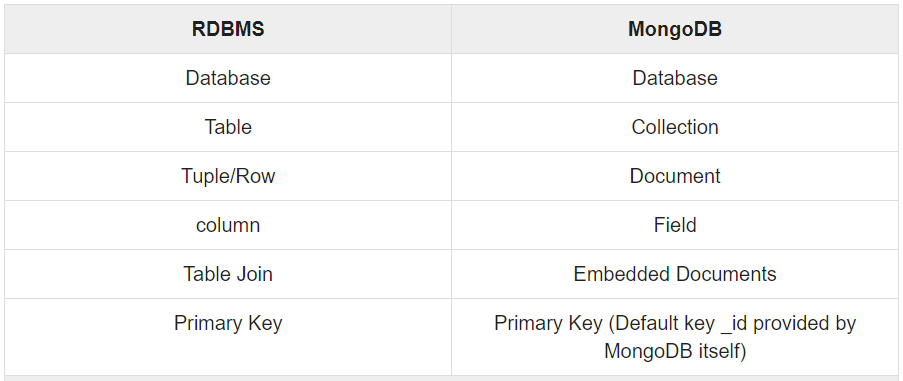
<https://www.tutorialspoint.com/neo4j/neo4j_overview.htm>

Apache Spark is a lightning-fast cluster computing technology, designed for fast computation. It is **based on Hadoop MapReduce** and it extends the MapReduce model to efficiently use it for more types of computations, which includes interactive queries and stream processing. **The main feature of Spark is its in-memory cluster computing that increases the processing speed of an application.**

Spark is designed to cover a wide range of workloads such as **batch applications**, iterative algorithms, interactive queries and **streaming**. Apart from supporting all these workload in a respective system, it reduces the management burden of maintaining separate tools.

<https://www.tutorialspoint.com/apache_spark/apache_spark_introduction.htm>

MongoDB is a cross-platform, document oriented database that provides, high performance, high availability, and easy scalability. MongoDB works on concept of collection and document.



## Advantages of MongoDB over RDBMS

* **Schema less** − MongoDB is a document database in which one collection holds different documents. **Number of fields, content and size of the document can differ from one document to another**.
* Structure of a single object is clear.
* No complex joins.
* Deep query-ability. MongoDB supports dynamic queries on documents using a document-based query language that's nearly as powerful as SQL.

<https://www.tutorialspoint.com/mongodb/mongodb_overview.htm>

**Machine Learning Algorithms:**

* Scikit-learn,
* Keras / Tensorflow,
* Fbprophet,
* Lightgbm,
* Xgboost,
* Shap / Lime