**INTRODUCTION:**

Classification is the process of predicting the class from a set of parameters of a particular record. It is all about learning how to make predictions from past example. It is the type of supervised learning as the data used for training of a model is labelled(each record belongs to a class). This is the training data for our example which labelled as each entity belongs to category

|  |  |  |  |
| --- | --- | --- | --- |
| Table 1 | | | |
| Firm | WC | DC | Category |
| 1 | 3338.61 | 0.56555 | 1 |
| 2 | 3801.72 | 0.570567 | 1 |
| 3 | 2818.817 | 0.572058 | 1 |
| 4 | 1250.953 | 0.568258 | 1 |
| 5 | 2444.406 | 0.553276 | 1 |
| 6 | 937.917 | 0.561066 | 1 |
| 7 | 1600.792 | 0.534662 | 1 |
| 8 | 3128.813 | 0.564714 | 1 |
| 9 | 2486.803 | 0.564239 | 1 |
| 10 | 4220.996 | 0.58465 | 1 |
| 11 | 2585.41 | 0.572457 | 1 |
| 12 | 3512.085 | 0.550878 | 1 |
| 13 | 4170.333 | 0.569516 | 1 |
| 14 | 938.879 | 0.545574 | 1 |
| 15 | 1437.695 | 0.529922 | 1 |
| 16 | 627.985 | 0.51941 | 1 |
| 17 | 4430.049 | 0.567547 | 1 |
| 18 | 989.568 | 0.534501 | 1 |
| 19 | 3275.474 | 0.555306 | 1 |
| 20 | 1500.437 | 0.565886 | 1 |
| 21 | 848.989 | 0.548603 | 1 |
| 22 | 1386.494 | 0.56229 | 1 |
| 23 | 1554.257 | 0.562346 | 1 |
| 24 | 2228.338 | 0.565556 | 1 |
| 25 | 2568.391 | 0.54973 | 1 |
| 26 | 1720.128 | 0.568458 | 1 |
| 27 | 4106.106 | 0.57767 | 1 |
| 28 | 3500.883 | 0.557197 | 1 |
| 29 | 1217.846 | 0.525333 | 1 |
| 30 | 3544.406 | 0.568735 | 1 |
| 31 | 2082.873 | 0.557527 | 1 |
| 32 | 709.01 | 0.541673 | 1 |
| 33 | 2523.939 | 0.55366 | 1 |
| 34 | 2781.307 | 0.569188 | 1 |
| 35 | 309.577 | 0.557668 | 0 |
| 36 | 363.79 | 0.561751 | 0 |
| 37 | 341.399 | 0.550717 | 0 |
| 38 | 363.616 | 0.568882 | 0 |
| 39 | 323.673 | 0.554490 | 0 |
| 40 | 323.353 | 0.558233 | 0 |
| 41 | 350.371 | 0.566447 | 0 |
| 42 | 240.602 | 0.5656 | 0 |
| 43 | 220.057 | 0.544182 | 0 |
| 44 | 287.837 | 0.522119 | 0 |
| 45 | 274.6 | 0.551492 | 0 |
| 46 | 278.494 | 0.550846 | 0 |
| 47 | 234.267 | 0.554828 | 0 |
| 48 | 284.923 | 0.533586 | 0 |
| 49 | 190.62 | 0.54899 | 0 |
| 50 | 327.76 | 0.538896 | 0 |
| 51 | 211.94 | 0.551569 | 0 |
| 52 | 373.571 | 0.549753 | 0 |
| 53 | 219.891 | 0.546936 | 0 |
| 54 | 193.489 | 0.56059 | 0 |
| 55 | 204.333 | 0.550777 | 0 |
| 56 | 205.657 | 0.550677 | 0 |
| 57 | 362.361 | 0.551315 | 0 |
| 58 | 285.562 | 0.578965 | 0 |
| 59 | 352.649 | 0.541763 | 0 |
| 60 | 400.44 | 0.557809 | 0 |
| 61 | 307.301 | 0.578949 | 0 |
| 62 | 240.314 | 0.548355 | 0 |
| 63 | 322.995 | 0.569978 | 0 |
| 64 | 408.197 | 0.574972 | 0 |
| 65 | 209.027 | 0.554203 | 0 |
| 66 | 198.979 | 0.559771 | 0 |
| 67 | 340.418 | 0.57343 | 0 |
| 68 | 320.154 | 0.560660 | 0 |

This is the testing data of which we have to classify the category.

|  |  |  |
| --- | --- | --- |
| Table 2 | | |
| Firm | WC | DC |
| 1 | 4204.066 | 0.578231 |
| 2 | 1411.733 | 0.560415 |
| 3 | 4197.206 | 0.565368 |
| 4 | 1121.866 | 0.540554 |
| 5 | 820.683 | 0.566067 |
| 6 | 1349.887 | 0.524683 |
| 7 | 3128.736 | 0.547596 |
| 8 | 2551.433 | 0.57368 |
| 9 | 809.115 | 0.552148 |
| 10 | 2866.623 | 0.559484 |
| 11 | 1193.951 | 0.515996 |
| 12 | 2014.445 | 0.564598 |
| 13 | 4400.268 | 0.578645 |
| 14 | 266.396 | 0.550131 |
| 15 | 243.554 | 0.559966 |
| 16 | 172.184 | 0.566274 |
| 17 | 362.479 | 0.553563 |
| 18 | 249.981 | 0.55274 |
| 19 | 327.877 | 0.565451 |
| 20 | 286.696 | 0.572919 |
| 21 | 182.762 | 0.56313 |
| 22 | 338.347 | 0.546618 |
| 23 | 302.57 | 0.551846 |
| 24 | 1781.718 | 0.564307 |
| 25 | 3711.358 | 0.570857 |
| 26 | 2030.189 | 0.564332 |
| 27 | 845.019 | 0.550468 |
| 28 | 1925.183 | 0.574114 |
| 29 | 1549.089 | 0.538726 |
| 30 | 1953.371 | 0.577015 |
| 31 | 932.5 | 0.564721 |
| 32 | 924.554 | 0.554162 |
| 33 | 2386.011 | 0.545268 |
| 34 | 2112.875 | 0.560262 |
| 35 | 3568.877 | 0.561775 |
| 36 | 4104.984 | 0.570978 |
| 37 | 367.325 | 0.533232 |
| 38 | 347.513 | 0.552354 |
| 39 | 330.226 | 0.549790 |
| 40 | 178.106 | 0.574400 |
| 41 | 378.899 | 0.531441 |
| 42 | 257.212 | 0.565379 |
| 43 | 333.088 | 0.54545 |
| 44 | 182.324 | 0.569686 |
| 45 | 238.099 | 0.563344 |
| 46 | 329.643 | 0.558005 |
| 47 | 294.644 | 0.556574 |
| 48 | 1058.649 | 0.54729 |
| 49 | 956.021 | 0.546774 |
| 50 | 2089.824 | 0.572031 |
| 51 | 2198.033 | 0.558597 |
| 52 | 4538.527 | 0.560383 |
| 53 | 3137.934 | 0.544445 |
| 54 | 2002.459 | 0.58141 |
| 55 | 2136.376 | 0.562953 |
| 56 | 281.666 | 0.553904 |
| 57 | 308.086 | 0.553646 |
| 58 | 317.079 | 0.560538 |
| 59 | 245.139 | 0.567829 |
| 60 | 354.662 | 0.548939 |
| 61 | 292.256 | 0.557991 |
| 62 | 306.79 | 0.57065 |
| 63 | 222.396 | 0.547811 |
| 64 | 367.628 | 0.53711 |
| 65 | 342.115 | 0.562531 |
| 66 | 353.326 | 0.548094 |
| 67 | 336.39 | 0.539131 |
| 68 | 298.008 | 0.562856 |

In our example we have performed three classification techniques which are given below:

1-Support vector machine

2-Naive Bayes

3-Decision tree

Each of them is described below;

**1-SUPPORT VECTOR MACHINE:**

A support-vector machine makes a [hyperplane](https://en.wikipedia.org/wiki/Hyperplane) which can be used for [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis), or other tasks like outliers detection Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the [generalization error](https://en.wikipedia.org/wiki/Generalization_error) of the classifier.

We have selected randomly 40 records for the training. The description of Trained model is shown below in Fig 1

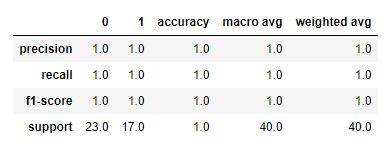
****

Fig 1

Confusion matrix for the trained model is shown which is indicating that the model achieved accuracy of 100%.

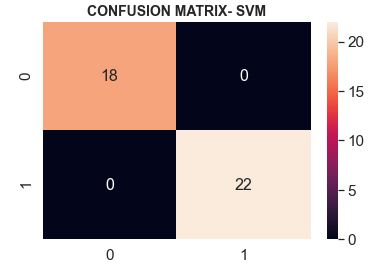
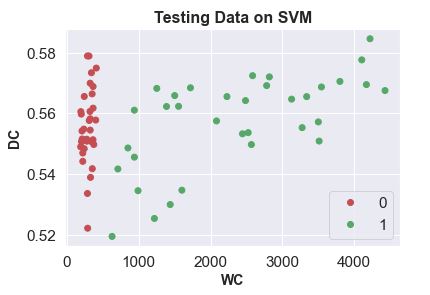
****

Fig 2

We tested the trained model on our tested data of 40 randomly selected samples and its successfully classified the dataset into respective category.

****

` Fig 3

|  |  |  |  |
| --- | --- | --- | --- |
| **Firm** | **WC** | **DC** | **Category** |
| 62 | 306.79 | 0.57065 | 0 |
| 58 | 317.079 | 0.560538 | 0 |
| 25 | 3711.358 | 0.570857 | 1 |
| 13 | 4400.268 | 0.578645 | 1 |
| 68 | 298.008 | 0.562856 | 0 |
| 56 | 281.666 | 0.553904 | 0 |
| 52 | 4538.527 | 0.560383 | 1 |
| 42 | 257.212 | 0.565379 | 0 |
| 20 | 286.696 | 0.572919 | 0 |
| 15 | 243.554 | 0.559966 | 0 |
| 45 | 238.099 | 0.563344 | 0 |
| 37 | 367.325 | 0.533232 | 0 |
| 22 | 338.347 | 0.546618 | 0 |
| 54 | 2002.459 | 0.58141 | 1 |
| 4 | 1121.866 | 0.540554 | 1 |
| 16 | 172.184 | 0.566274 | 0 |
| 57 | 308.086 | 0.553646 | 0 |
| 63 | 222.396 | 0.547811 | 0 |
| 3 | 4197.206 | 0.565368 | 1 |
| 48 | 1058.649 | 0.54729 | 1 |
| 51 | 2198.033 | 0.558597 | 1 |
| 9 | 809.115 | 0.552148 | 1 |
| 41 | 378.899 | 0.531441 | 0 |
| 38 | 347.513 | 0.552354 | 0 |
| 67 | 336.39 | 0.539131 | 0 |
| 5 | 820.683 | 0.566067 | 1 |
| 28 | 1925.183 | 0.574114 | 1 |
| 59 | 245.139 | 0.567829 | 0 |
| 2 | 1411.733 | 0.560415 | 1 |
| 29 | 1549.089 | 0.538726 | 1 |
| 53 | 3137.934 | 0.544445 | 1 |
| 11 | 1193.951 | 0.515996 | 1 |
| 6 | 1349.887 | 0.524683 | 1 |
| 46 | 329.643 | 0.558005 | 0 |
| 55 | 2136.376 | 0.562953 | 1 |
| 36 | 4104.984 | 0.570978 | 1 |
| 21 | 182.762 | 0.56313 | 0 |
| 18 | 249.981 | 0.55274 | 0 |
| 33 | 2386.011 | 0.545268 | 1 |
| 64 | 367.628 | 0.53711 | 0 |

**2-NAIVE BAYES:**

Naive Bayes classifiers based on applying [Bayes' theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem) with strong(naïve) independence assumption between the features. They are among the simplest [Bayesian network](https://en.wikipedia.org/wiki/Bayesian_network) models. But they could be coupled with [Kernel density estimation](https://en.wikipedia.org/wiki/Kernel_density_estimation) and achieve higher accuracy levels

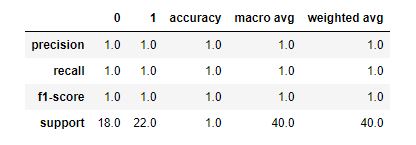
****We have selected randomly 40 records for the training. The description of Trained model is shown below:

Fig 4

Confusion matrix for the trained model is shown which is indicating that the model achieved accuracy of 100%

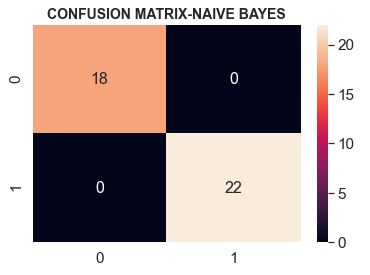
****

Fig 5

We tested the trained model on our tested data of 40 randomly selected samples and its successfully classified the dataset into respective category.

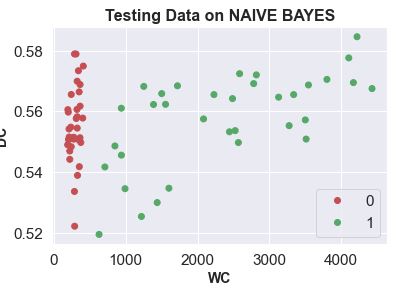
****

Fig 6

|  |  |  |  |
| --- | --- | --- | --- |
| **Firm** | **WC** | **DC** | **Category** |
| 50 | 2089.824 | 0.572031 | 1 |
| 4 | 1121.866 | 0.540554 | 1 |
| 63 | 222.396 | 0.547811 | 0 |
| 13 | 4400.268 | 0.578645 | 1 |
| 41 | 378.899 | 0.531441 | 0 |
| 24 | 1781.718 | 0.564307 | 1 |
| 5 | 820.683 | 0.566067 | 1 |
| 20 | 286.696 | 0.572919 | 0 |
| 42 | 257.212 | 0.565379 | 0 |
| 52 | 4538.527 | 0.560383 | 1 |
| 12 | 2014.445 | 0.564598 | 1 |
| 34 | 2112.875 | 0.560262 | 1 |
| 15 | 243.554 | 0.559966 | 0 |
| 9 | 809.115 | 0.552148 | 1 |
| 37 | 367.325 | 0.533232 | 0 |
| 46 | 329.643 | 0.558005 | 0 |
| 28 | 1925.183 | 0.574114 | 1 |
| 53 | 3137.934 | 0.544445 | 1 |
| 64 | 367.628 | 0.53711 | 0 |
| 51 | 2198.033 | 0.558597 | 1 |
| 62 | 306.79 | 0.57065 | 0 |
| 33 | 2386.011 | 0.545268 | 1 |
| 3 | 4197.206 | 0.565368 | 1 |
| 39 | 330.226 | 0.549799 | 0 |
| 14 | 266.396 | 0.550131 | 0 |
| 54 | 2002.459 | 0.58141 | 1 |
| 35 | 3568.877 | 0.561775 | 1 |
| 21 | 182.762 | 0.56313 | 0 |
| 36 | 4104.984 | 0.570978 | 1 |
| 31 | 932.5 | 0.564721 | 1 |
| 7 | 3128.736 | 0.547596 | 1 |
| 66 | 353.326 | 0.548094 | 0 |
| 38 | 347.513 | 0.552354 | 0 |
| 23 | 302.57 | 0.551846 | 0 |
| 18 | 249.981 | 0.55274 | 0 |
| 6 | 1349.887 | 0.524683 | 1 |
| 49 | 956.021 | 0.546774 | 1 |
| 27 | 845.019 | 0.550468 | 1 |
| 29 | 1549.089 | 0.538726 | 1 |
| 26 | 2030.189 | 0.564332 | 1 |

**3-DECISION TREE:**

A decision tree is a [flowchart](https://en.wikipedia.org/wiki/Flowchart)-like structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

We have selected randomly 40 records for the training. The description of Trained model is shown below in in Fig 7

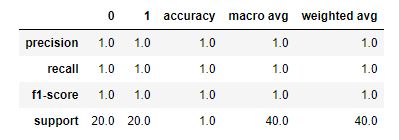
****

Fig 7

Confusion matrix for the trained model is shown which is indicating that the model achieved accuracy of 100%

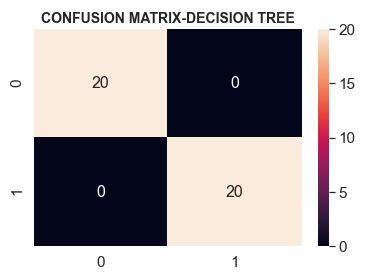
****

Fig 8

We tested the trained model on our tested data of 40 randomly selected samples and its successfully classified the dataset into respective category.

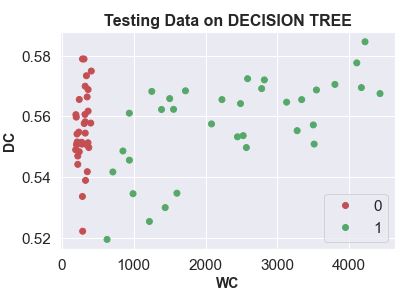
****

Fig 9

|  |  |  |  |
| --- | --- | --- | --- |
| **Firm** | **WC** | **DC** | **Category** |
| 48 | 1058.649 | 0.54729 | 1 |
| 19 | 327.877 | 0.565451 | 0 |
| 68 | 298.008 | 0.562856 | 0 |
| 56 | 281.666 | 0.553904 | 0 |
| 61 | 292.256 | 0.557991 | 0 |
| 16 | 172.184 | 0.566274 | 0 |
| 26 | 2030.189 | 0.564332 | 1 |
| 27 | 845.019 | 0.550468 | 1 |
| 62 | 306.79 | 0.57065 | 0 |
| 24 | 1781.718 | 0.564307 | 1 |
| 42 | 257.212 | 0.565379 | 0 |
| 47 | 294.644 | 0.556574 | 0 |
| 63 | 222.396 | 0.547811 | 0 |
| 50 | 2089.824 | 0.572031 | 1 |
| 4 | 1121.866 | 0.540554 | 1 |
| 6 | 1349.887 | 0.524683 | 1 |
| 34 | 2112.875 | 0.560262 | 1 |
| 38 | 347.513 | 0.552354 | 0 |
| 32 | 924.554 | 0.554162 | 1 |
| 58 | 317.079 | 0.560538 | 0 |
| 54 | 2002.459 | 0.58141 | 1 |
| 36 | 4104.984 | 0.570978 | 1 |
| 25 | 3711.358 | 0.570857 | 1 |
| 12 | 2014.445 | 0.564598 | 1 |
| 1 | 4204.066 | 0.578231 | 1 |
| 40 | 178.106 | 0.574406 | 0 |
| 59 | 245.139 | 0.567829 | 0 |
| 9 | 809.115 | 0.552148 | 1 |
| 35 | 3568.877 | 0.561775 | 1 |
| 37 | 367.325 | 0.533232 | 0 |
| 18 | 249.981 | 0.55274 | 0 |
| 52 | 4538.527 | 0.560383 | 1 |
| 51 | 2198.033 | 0.558597 | 1 |
| 64 | 367.628 | 0.53711 | 0 |
| 29 | 1549.089 | 0.538726 | 1 |
| 3 | 4197.206 | 0.565368 | 1 |
| 66 | 353.326 | 0.548094 | 0 |
| 20 | 286.696 | 0.572919 | 0 |
| 46 | 329.643 | 0.558005 | 0 |
| 15 | 243.554 | 0.559966 | 0 |