|  |  |
| --- | --- |
| Name Of The Student | Vaishnavi G |
| Internship Project Topic | Build a Classification Model for Drug Trials Dataset |
| Name of the Organization | TCS iON |
| Name of the Industry Mentor | Himdweep Walia |
| Name of the Institute | SRM Institute of Science and Technology |

|  |  |  |
| --- | --- | --- |
| Date | Day # | Hours Spent |
| 1/11/2022 | 21 | 5 hours |
| Activities done during the day:  **Balanced accuracy:**  Balanced accuracy is a metric we can use to assess the performance of a classification model.  It is calculated as:   |  | | --- | | Balanced accuracy = (Sensitivity + Specificity) / 2 |   where:  **Sensitivity:** The “true positive rate” – the percentage of positive cases the model is able to detect.  **Specificity:** The “true negative rate” – the percentage of negative cases the model is able to detect.  This metric is particularly useful when the two classes are imbalanced – that is, one class appears much more than the other.  **To calculate the balanced accuracy of the model, we’ll first calculate the sensitivity and specificity:**  **Sensitivity:** The “true positive rate” = 15 / (15 + 5) = 0.75  **Specificity:** The “true negative rate” = 375 / (375 + 5) = 0.9868  We can then calculate the balanced accuracy as:  Balanced accuracy = (Sensitivity + Specificity) / 2  Balanced accuracy = (0.75 + 9868) / 2  Balanced accuracy = 0.8684  The balanced accuracy for the model turns out to be 0.8684.  The following example shows how to calculate the balanced accuracy for this exact scenario using the balanced\_accuracy\_score() function from the sklearn library in Python.   |  | | --- | | import numpy as np  from sklearn.metrics import balanced\_accuracy\_score  #define array of actual classes  actual = np.repeat([1, 0], repeats=[20, 380])  #define array of predicted classes  pred = np.repeat([1, 0, 1, 0], repeats=[15, 5, 5, 375])  #calculate balanced accuracy score  balanced\_accuracy\_score(actual, pred)  0.868421052631579 |   **OUTPUT:**  The balanced accuracy is 0.8684.  **To calculation of the accuracy of a classification model, we must first train a model for any classification-based problem**   |  | | --- | | from sklearn.datasets import make\_classification  from sklearn.model\_selection import train\_test\_split  from sklearn.metrics import accuracy\_score  from sklearn.linear\_model import LogisticRegression  nb\_samples = 1000  x, y = make\_classification(n\_samples=nb\_samples, n\_features=2, n\_informative=2, n\_redundant=0, n\_clusters\_per\_class=1)  xtrain, xtest, ytrain, ytest = train\_test\_split(x, y, test\_size=0.2, random\_state=42)  model = LogisticRegression()  model.fit(xtrain, ytrain)  print(accuracy\_score(ytest, model.predict(xtest))) |   **OUTPUT:** 0.99  Reference:  <https://discuss.analyticsvidhya.com/t/how-to-calculate-the-accuracy-score-of-a-model-in-python/5622/>  <https://towardsdatascience.com/how-to-evaluate-your-machine-learning-models-with-python-code-5f8d2d8d945b/> | | |