|  |  |
| --- | --- |
| 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 |
| 11/11/2022 | 31 | 3 hours |
| Activities done during the day:  **Standardization**  Standardization of datasets is a common requirement for many machine learning estimators implemented in scikit-learn.  They might behave badly if the individual features do not more or less look like standard normally distributed data: Gaussian with zero mean and unit variance.  Scaling of Features is an essential step in modeling the algorithms with the datasets.  The data that is usually used for the purpose of modeling is derived through various means such as:   * Questionnaire * Surveys * Research * Scraping, etc.   So, the data obtained contains features of various dimensions and scales altogether. Different scales of the data features affect the modeling of a dataset adversely.  It leads to a biased outcome of predictions in terms of misclassification error and accuracy rates. Thus, it is necessary to Scale the data prior to modeling.  Standardization is a scaling technique wherein it makes the data scale-free by converting the statistical distribution of the data into the below format:   * mean - 0 (zero) * standard deviation - 1   the entire data set scales with a zero mean and unit variance, altogether.  **Python sklearn StandardScaler() function**  StandardScaler() function to standardize the data values into a standard format.  **Syntax:**   |  | | --- | | object = StandardScaler()  object.fit\_transform(data) |   According to the above syntax, we initially create an object of the StandardScaler() function. Further, we use fit\_transform() along with the assigned object to transform the data and standardize it.  Standardization is only applicable on the data values that follows Normal Distribution.  **Standardizing data with StandardScaler() function**   |  | | --- | | from sklearn.datasets import load\_iris  from sklearn.preprocessing import StandardScaler    dataset = load\_iris()  object= StandardScaler()    # Splitting the independent and dependent variables  i\_data = dataset.data  response = dataset.target    # standardization  scale = object.fit\_transform(i\_data)  print(scale) |   **Explanation:**   * Import the necessary libraries required. We have imported sklearn library to use the StandardScaler function. * Load the dataset. Here we have used the IRIS dataset from sklearn.datasets library. You can find the dataset here. * Set an object to the StandardScaler() function. * Segregate the independent and the target variables as shown above. * Apply the function onto the dataset using the fit\_transform() function.   **Output:**  IMG_256  Reference:  <https://scikit-learn.org/stable/modules/preprocessing.html/> | | |