**Performance of various Machine Learning Algorithms on Electrical Impedance Tomography Images**

1. **Tools**
   1. Programming Language: Python
   2. Libraries: numpy, scipy, pandas, matplotlib
   3. OpenCV (cv2) for image processing
   4. Sci-kit Learn for Machine Learning
2. **Reading Images**
   1. Image is read into code in the form of a three-dimensional matrix where in each dimension represents intensities of the respective colour code
   2. This three-dimensional matrix is then converted to two-dimensional matrix (representation of grayscale image) with intensities ranging from 0 to 1
   3. Image is re generated to observe distribution using contour plots
3. **Generating Images**
   1. Based on the data obtained and observation from the graphs, random multidimensional matrices are generated
   2. Using radial basis function on these matrices, values ranging from 0 to 1 are created
   3. 1000 random-related images are created based on the matrices and its values
4. **Image Classification**
   1. The generated images are read back into code and are plotted to observe the distribution of intensities
   2. Mean intensity ranges are calculated and are assigned labels (colours) correspondingly
5. **Dataset Generation**
   1. The generated images are parsed and respective intensity ranges, its count of pixels and percentages are calculated
   2. A dataset of 8 intensity ranges (columns) and 1000 values (rows) are created
6. **Dataset Target Generation**
   1. Mean of pixel count of all ranges are taken in consideration and is used as a criterion for assigning targets
   2. Binary targets are generated and are appended to the existing dataset as a target column
7. **Machine Learning – Classification**
   1. Dataset is split into training set and test set
   2. Feature scaling is applied, data is fit into various classifiers and models are evaluated for performance
   3. Confusion matrices and classification reports are generated
   4. List of classifiers/algorithms used along with performance accuracy (%):
      1. K – Nearest Neighbours
      2. Decision Tree Classifier
      3. Kernel Support Vector Machines
      4. Logistic Regression Classifier
      5. Naïve Bayes Classifier
      6. Random Forest Classifier
      7. Support Vector Machines
8. **Results**

