Classification of Presence or Absence of Heart Attack Using Support Vector Machines

Group 4

Ameya Prasanna Mote 3104966@stud.srh-campus-berlin.de

Nanma Joseph 3104969@stud.srh-campus-berlin.de

Sayli Akarshe 3105190@stud.srh-campus-berlin.de

Chaitanya Malpure 3105038@stud-srh-campus-berlin.de

As we started with the journey of **Support Vector Machines**, we primarily decided to focus on the various kernels that have become associated with it. Our main purpose was to have a deeper understanding of SVM and not just understand how the code works. For this reason, we dedicated extra time to learn about **Linear, RBF**, and the various other kernels. A lot of papers were referred to get a better view of our hyperplanes work. Due to constant reading, a point finally came were even the most difficult aspects of this **Supervised Learning algorithm** could be explained by us. This was overall purpose of the Technology Project III. A good Project is only as good as it is presentation, and a lot of focus was also given to it. As our Dataset, we decided to Use the **Cleveland Heart Attack dataset** which has 76 attributes in the original form which was then toned down to 14 attributes. Such a dataset was chosen to provide simplicity as well as to determine a way to acknowledge the presence of heart attack in a person before unfortunate consequences happen.

We started off with a simple Introduction into the Support Vector Machines. This was very well explained by Chaitanya Malpure. It paved the road for what was to come. He explained the decision boundaries and the support vectors along with the graphs. Often a question comes as to why we should use Support Vector Machines and not choose from other Supervised learning Algorithms and this was also explained in our Presentation. There are always datasets that are not always linear in nature and SVM has accommodation for such kind of datasets too. Chaitanya also goes to explain how prevalent heart attacks have become in today’s world along with some statistics. Nanma Joseph then sets to explain the different hyperparameters that have come to be associated with SVM. Different kernels, C and the gamma value was explained in these slides. A very detailed study of the mathematics involved in the SVM was also explained in the corresponding slides. This was particularly emphasised on as anyone could look at the documentation and type in the code without understanding the background work that is happening. For this point itself, more time was spent on explaining each kernel and its performance along with deep dive into Support vectors and Hyperplanes. Then came the part where the code was supposed to explained. Ameya Mote started it of with SVM using Sklearn. Sklearn is one of the best machine learning libraries and has been time and again used to solve most complex mathematical algorithms. A walkthrough was given on different libraries that were used along with an explanation of the code and the accuracy was achieved of about 80%.

Realisation of the same code was proving to be a bit difficult on TensorFlow 2.3. Thus, using back grade the code was first ran on TensorFlow 1.8 and after tuning and retuning it, was again deployed on the newest version. Sayli Akarshe oversaw all of this and she explained the different parts perfectly along with the walkthrough of the code and in the end finish it with excellent visualised graphs. The group also had fun to test out new visualizations using TensorBoard. It offers tracking and visualizing metrics such as loss and accuracy. It also helps in displaying images, text and audio data.

This was a very good experience for all of us as we were able to learn a particular machine learning algorithm right from the basics and gain our confidence in its different aspects. Talking about the references we used, each research paper was thoroughly read by every member of the group to tackle any questions that would have been asked after the presentation. Each major point of this project is also properly documented for future purposes. For initial experiments, Jupyter Notebook was preferred and once the group was set on the code workflow, final code was done on Google Colab. Both offer excellent functionalities but Colab more so with its GPU power.

For future work, the group would really look forward to employing the model on a bigger dataset. It will require some fine tuning, but if some deadly disease patterns could be observed, it will go a long way in increasing human life span. At the end, we would like to thank Prof. Alexander lliev for giving us this exciting opportunity to dive into the different departments of Support Vector Machine and come out with a nice understanding of it.