Attendance Score | SVKM's NMIMS
Mukesh Patel School of Technology Management and Engineering
B.Tech Integrated Program | 2022-2023

A Report on

INPLANT TRAINING

AT

Jio Platforms Limited

Reliance Corporate Park, MUMBAI – 400701

Submitted By

Sahil Shah

SAP ID: 70321019091

Under The Guidance Of

Prashasti Kanikar

Shri Vile Parle Kelvani Mandal's

Mukesh Patel School of Technology & Management Engineering

Department of Computer Engineering

Vile Parle (w), Mumbai- 400056

Inplant Training Period

2nd January 2023 to 29th April 2023

1.1 Projects Undertaken

The New Initiatives AI/ML department at Jio Platforms is involved in several projects to develop and implement AI and ML-based products and services. Some of the key projects assigned by the department include:

1.1.1 Machine Learning Approach to Attendance Score

Attendance is an important part of workplace since it allows employees to accomplish their work commitments while also maintaining a productive work environment. Although assessing and controlling employee timeliness can have a detrimental impact on productivity, expenses, and employee morale, it is a challenge faced by many organizations.

The main objective of this project is to develop a Machine Learning model that can accurately determine the attendance score of employees using only their entry and exit timings, without any prior knowledge of their work schedules or shifts. The model will be designed to classify users as either Fixed or Rotational, and then generate a score on their level of tardiness based on their attendance patterns.

Problem Definition

This project aims to develop a Machine Learning model that provides an attendance score for employees based solely on their entry and exit timings.

Constraints and Challenges

1. **Consecutiveness**
    a. Punctual → If consecutively on time, then attendance score will be incremented but that increment will be at a steady rate as well as have a max cap amount which can be regenerated (optional)
    b. Unpunctual → If consecutively tardy then attendance score will decrease at a linear rate.

2. Regularize

a. If considering stores or shops which have to be opened at a specific timing by a certain individual and if that particular individual is late then the attendance score of other individuals shouldn't be affected

3. Absent employees

a. when calculating Attendance score of a particular individual then need to consider absent employees and also should have an option to remove it from the number of company leaves allocated

4. shifts and field work.

a. In many cases different individuals have to step out of the office premises for company-related matters and based on that there attendance shouldn't be affected.

Tasks Undertaken

1. **Data Gathering:** Data Gathering involved the use of in-house data to create a dataset from which we were able to develop a more comprehensive understanding of employee shift patterns, behavior and performance.

   a. In-house data was examined.

   b. Relevant data was compiled into a dataset till the data requirement was satisfied.

2. **Data Preparation:** The dataset created was pre-processed was performed to ensure robustness by introducing a more diverse range of data to the model.

   a. Collected data was split into a train/val split structure (80%/20%)

   b. Data Cleaning techniques:

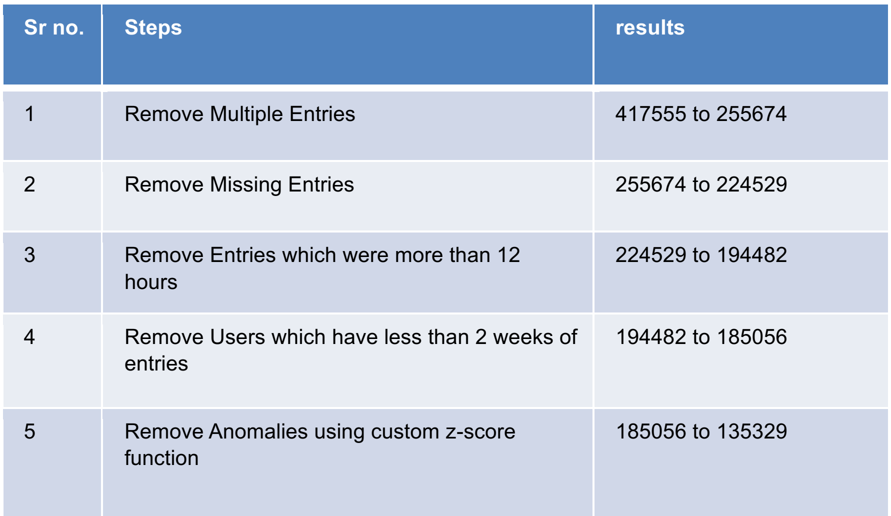
      i. Removal of Nan

      data ii. Removal of

noisy data iii. Removal

of outliers iv. Removal

of duplicate data



Sr no.	Steps	results
1	Remove Multiple Entries	417555 to 255674
2	Remove Missing Entries	255674 to 224529
3	Remove Entries which were more than 12 hours	224529 to 194482
4	Remove Users which have less than 2 weeks of entries	194482 to 185056
5	Remove Anomalies using custom z-score function	185056 to 135329

Fig 1.1.1 Tabular represnetation of cleaning results

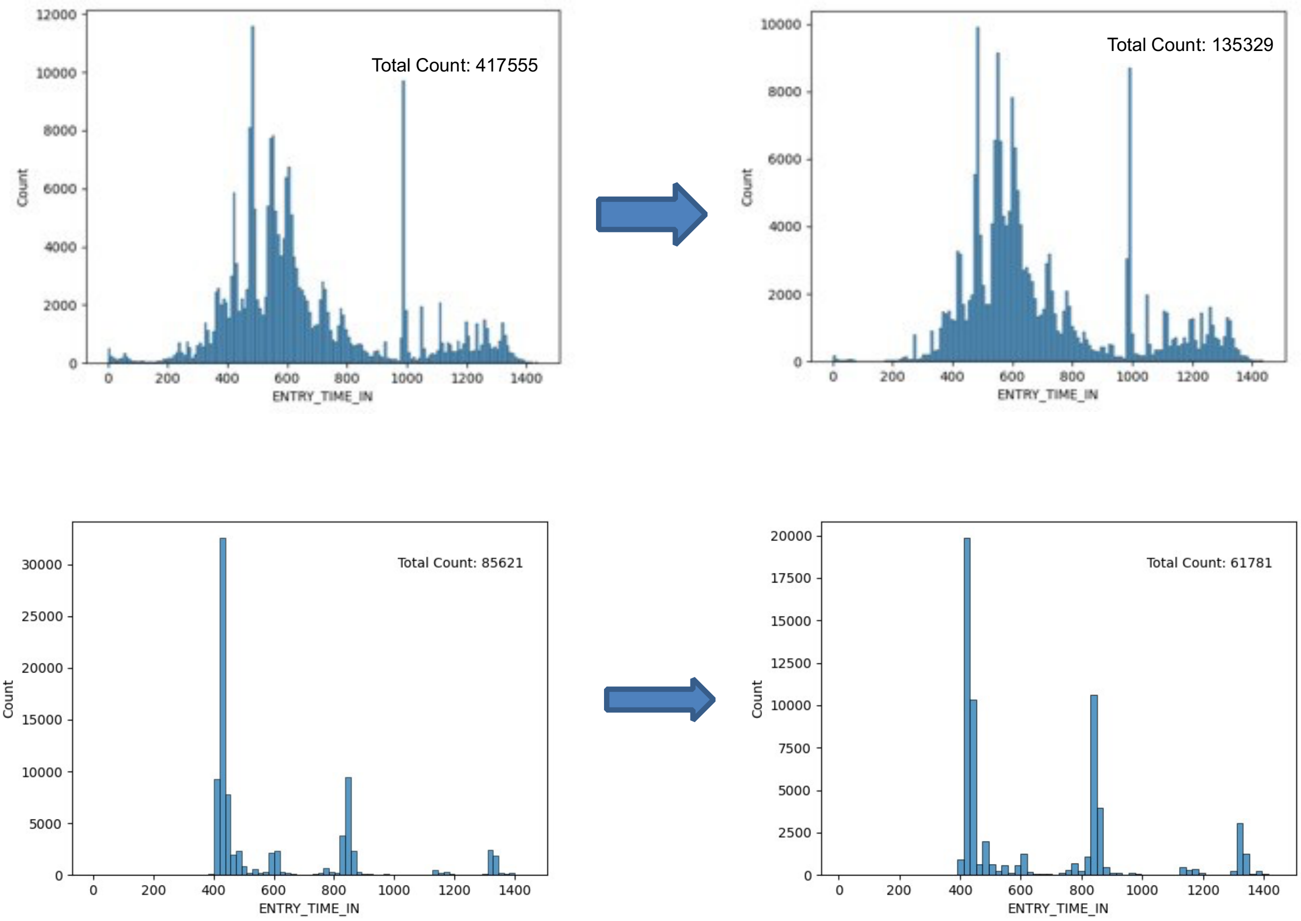


Fig 1.1.1 Count of Entries Before and After Data Cleaning

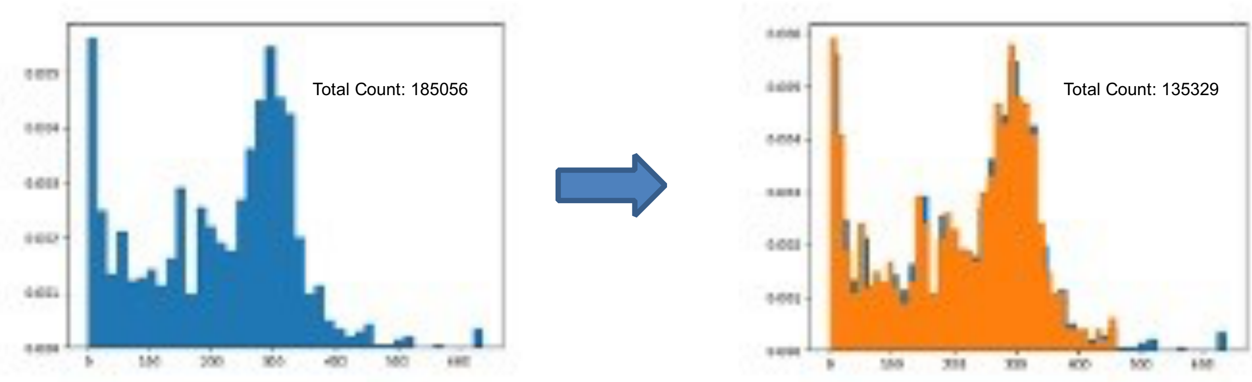
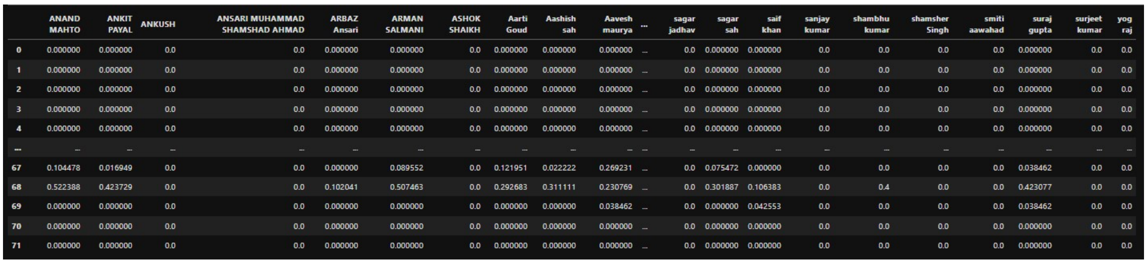


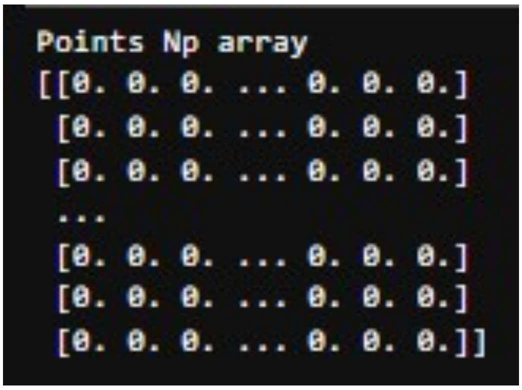
Fig 1.1.1 Pictorial Representation of data with and without Anomalises

c. Data transformation techniques:

i. Normalization ii. One Hot encoding iii. Feature Extraction iv. Convert into NumPy array to pass in the model



	ANAND MAHTO	ANKIT PAYAL	ANKUSH	ANSARI MUHAMMAD SHAMSHAD AHMAD	ARBAZ Ansari	ARMAN SALMANI	ASHOK SHAIKH	Aarti Goud	Aashish sah	Aavesh maurya	...	sagar jadhav	sagar sah	saif khan	sanjay kumar	shambhu kumar	shamsher Singh	smiti aawahad	suraj gupta	surjeet kumar	yog raj
0	0.000000	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	...	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
1	0.000000	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	...	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
2	0.000000	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	...	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
3	0.000000	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	...	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
4	0.000000	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	...	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
67	0.104478	0.016949	0.0	0.0	0.000000	0.089552	0.0	0.121951	0.022222	0.269231	...	0.0	0.075472	0.000000	0.0	0.0	0.0	0.0	0.038462	0.0	0.0
68	0.522388	0.423729	0.0	0.0	0.102041	0.507463	0.0	0.292683	0.311111	0.230769	...	0.0	0.301887	0.106383	0.0	0.4	0.0	0.0	0.423077	0.0	0.0
69	0.000000	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.038462	...	0.0	0.000000	0.042553	0.0	0.0	0.0	0.0	0.038462	0.0	0.0
70	0.000000	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	...	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0
71	0.000000	0.000000	0.0	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	...	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0	0.0



```
Points Np array
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

Fig 1.1.1Data Transformation Techniques

3. **Model Selection:** Scikit-learn, a framework built over python was our primary Machine learning framework. We proceeded to test different clustering algorithms to find an optimal and efficient algorithm for classifying users:

    a. K-Means Clustering

    b. K-Nearest Neighbor

    c. Density Based Clustering (DBSCAN)

4. **Model Training/Finetuning Pipeline:** Applied K-Means Clustering Algorithm to the dataset and parameters were tuned to obtain a model checkpoint that performs as per our requirement.

a. Different distance calculation metrics were used such as Euclidean, Manhattan and Cosine

b. Loaded the model on a preset range of 1 – 10 number of clusters and chose the best output on different datasets

c. Best performing tuned models were selected for further fine tuning over in-house dataset.

5. **Model Evaluation:** : To evaluate the performance of the model, we utilized both the silhouette score and the elbow method. These metrics provided insight into the quality of the clustering and helped determine the optimal number of clusters for accurately identifying shift information.

a. Model was tested on test dataset

b. Elbow Method is a graphical representation of finding the optimal 'K' in a K-means clustering.

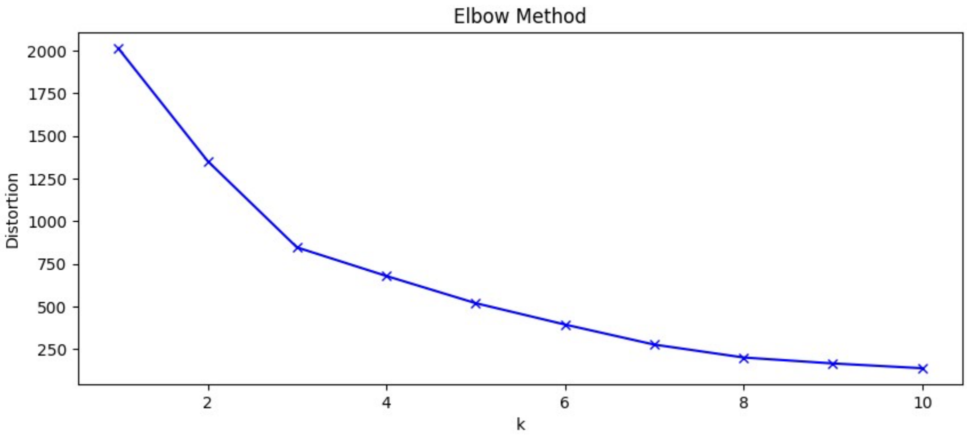


Fig 1.1.1 Elbow method for finding optimal number of clusters

c. silhouette score was used to measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation).

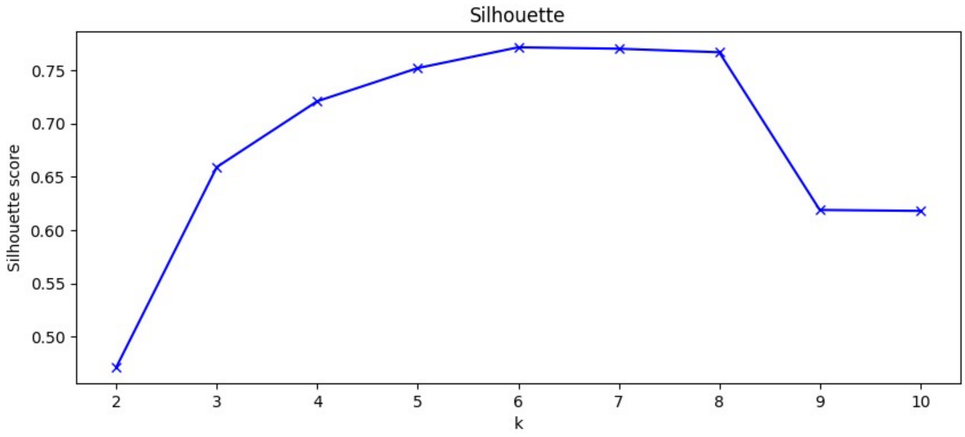


Fig 1.1.1 Silhoutte Score for finding optimal number of clusters

d. Davies-Bouldin Index was used to measure the similarity between clusters and is calculated as the average similarity between each cluster and its most similar cluster. A lower Davies-Bouldin Index indicates betterdefined clusters.

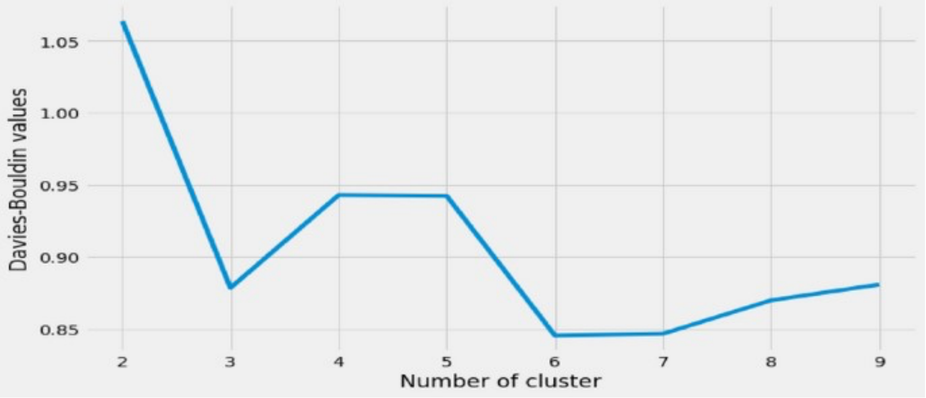


Fig 1.1.1 Dave-Bouldin Index for finding optimal number of clusters

e. Used manual inference to extract meaningful insights from the model, which allowed for a deeper understanding of the data and helped fine-tune the model's parameters for optimal performance.

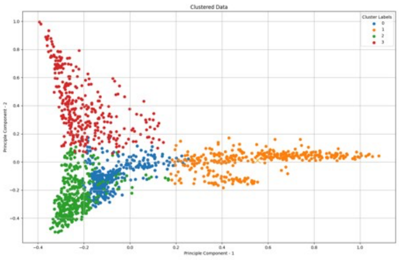
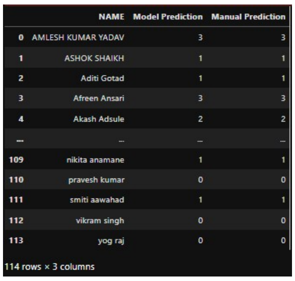
6. **Model Testing**: During the Model Testing phase, we classified each data point into a cluster and subsequently examined the clusters to ensure accurate classification of all data points.

    a. Created scripts to annotate the data and then manually compared the annotations with the model's predictions

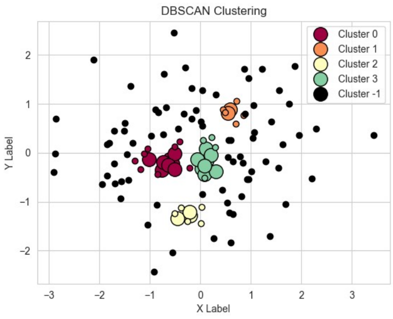
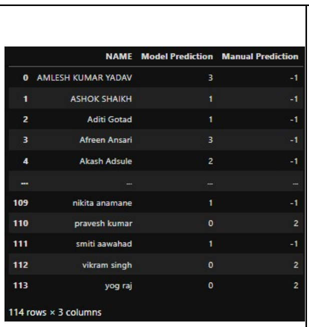
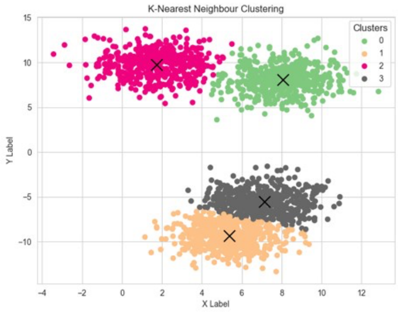
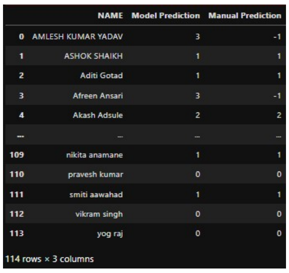
    b. Used Pairwise distance metric to evaluate each data points distance to a cluster and assigned the cluster with lowest distance

Data Point	Distance to Centroid 1	Distance to Centroid 2	Distance to Centroid 3	Distance to Centroid 4	Centroid Selection
1	1.50474	1.49749	**0.89774**	1.52502	Centroid 3
2	1.41984	1.36486	1.18093	**0.46165**	Centroid 4
…..	….	….	….	….	….
n	0.90217	**0.73952**	0.98847	1.22587	Centroid 2

Fig 1.1.1 Tabular Representation of pairwuse distance method working

Algorithm	Plot	Predictions	Accuracy
• K-Means Clustering			96%

```
                 NAME  Model Prediction  Manual Prediction
0   AMLESH KUMAR YADAV                3                  3
1         ASHOK SHAIKH                1                  1
2          Aditi Gotad                1                  1
3         Afreen Ansari               3                  3
4          Akash Adsule               2                  2
...                ...              ...                ...
109     nikita anamane                1                  1
110      pravesh kumar                0                  0
111      smiti aawahad                1                  1
112       vikram singh                0                  0
113            yog raj                0                  0
114 rows × 3 columns
```

<table>
<tr>
<td>• Density Based Clustering (DBSCAN)</td>
<td>
</td>
<td>
<table>
<tr><th></th><th>NAME</th><th>Model Prediction</th><th>Manual Prediction</th></tr>
<tr><td>0</td><td>AMLESH KUMAR YADAV</td><td>3</td><td>-1</td></tr>
<tr><td>1</td><td>ASHOK SHAIKH</td><td>1</td><td>-1</td></tr>
<tr><td>2</td><td>Aditi Gotad</td><td>1</td><td>-1</td></tr>
<tr><td>3</td><td>Afreen Ansari</td><td>3</td><td>-1</td></tr>
<tr><td>4</td><td>Akash Adsule</td><td>2</td><td>-1</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td></tr>
<tr><td>109</td><td>nikita anamane</td><td>1</td><td>-1</td></tr>
<tr><td>110</td><td>pravesh kumar</td><td>0</td><td>2</td></tr>
<tr><td>111</td><td>smiti aawahad</td><td>1</td><td>-1</td></tr>
<tr><td>112</td><td>vikram singh</td><td>0</td><td>2</td></tr>
<tr><td>113</td><td>yog raj</td><td>0</td><td>2</td></tr>
</table>
114 rows × 3 columns</td>
<td>78%</td>
</tr>
<tr>
<td>• K-Nearest Neighbor</td>
<td>
</td>
<td>
<table>
<tr><th></th><th>NAME</th><th>Model Prediction</th><th>Manual Prediction</th></tr>
<tr><td>0</td><td>AMLESH KUMAR YADAV</td><td>3</td><td>-1</td></tr>
<tr><td>1</td><td>ASHOK SHAIKH</td><td>1</td><td>1</td></tr>
<tr><td>2</td><td>Aditi Gotad</td><td>1</td><td>1</td></tr>
<tr><td>3</td><td>Afreen Ansari</td><td>3</td><td>-1</td></tr>
<tr><td>4</td><td>Akash Adsule</td><td>2</td><td>2</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td></tr>
<tr><td>109</td><td>nikita anamane</td><td>1</td><td>1</td></tr>
<tr><td>110</td><td>pravesh kumar</td><td>0</td><td>0</td></tr>
<tr><td>111</td><td>smiti aawahad</td><td>1</td><td>1</td></tr>
<tr><td>112</td><td>vikram singh</td><td>0</td><td>0</td></tr>
<tr><td>113</td><td>yog raj</td><td>0</td><td>0</td></tr>
</table>
114 rows × 3 columns</td>
<td>87%</td>
</tr>
</table>

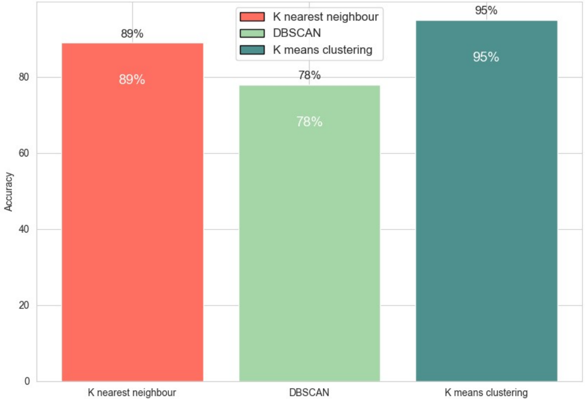


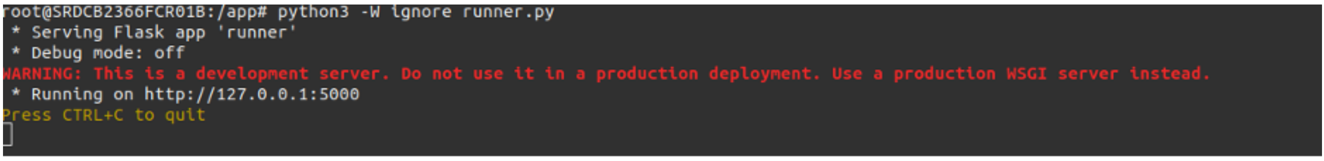
Fig 1.1.1 Accuracy for Algorithm

7. **Model Deployment**: We developed a custom Flask server and modularized the code to deploy the model in pickle format. We also created an API to convert the model into a SaaS solution. To test the API's functionality, we designed and executed a series of integration tests using a simulated client, which emulated real-world user requests. These tests helped ensure that the API was working correctly and provided reliable results..

Fig 1.1.1 Screenshot of Flask app deployed on server

1.1.2 Deep Learning approach to Attendance Score

The aim of this project is to build a deep learning model using a fully connected Neural Network that can classify users as either having fixed or rotating shifts. This binary classifier will enable the calculation of attendance scores based on entry and exit timings, providing organizations with an effective tool for workforce management.



```
root@SRDCB2366FCR01B:/app# python3 -W ignore runner.py
 * Serving Flask app 'runner'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

Problem Definition

The problem was defined as the need for a more accurate system that can recognize fixed or rotational user.

Tasks Undertaken

1. **Data Collection:** Inhouse dataset, which was created for machine learning approach was used to develop the classification model.
    a. Data from various sites was examined.
    b. Relevant data was compiled into a dataset till the data requirement was satisfied.
2. **Data Preparation:** The dataset created was pre-processed to ensure robustness by introducing a more diverse range of data to the model.
    a. Collected data was split into a train/test split structure (80%/20%)

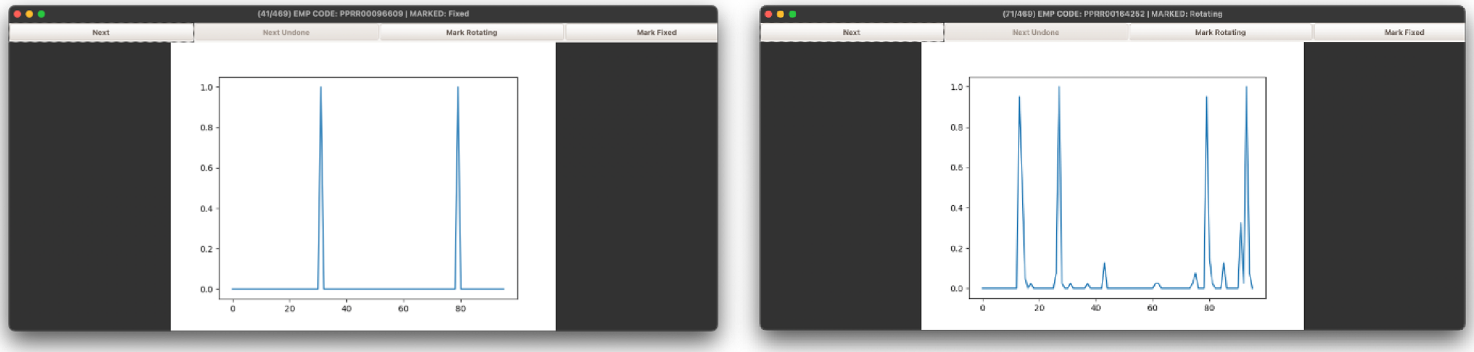
b. Implemented a python GUI based scripts which would display a plot of each users normalized data and then we used buttons to append whether user is fixed or rotational to the dataset

Fig 1.1.2 GUI interface of application developed to dynamically annonate data

c. Data was labelled as Fixed and Rotational data.

d. Data processing techniques were applied such as:

i. Data



(41/469) EMP CODE: PPRR00096609 | MARKED: Fixed

(71/469) EMP CODE: PPRR00164252 | MARKED: Rotating

Normalization ii.

One Hot Encoding

3. **Model Selection:** PyTorch was our primary deep learning framework. To identify the optimal model architecture, we conducted extensive experimentation, varying parameters such as the number of layers, the number of neurons in each layer, and the activation functions.

a. Research regarding current best performing text based classifier models was conducted.

b. batch size, learning rate, and dropout rate was also considered to ensure that the model was accurately trained on the underlying data.

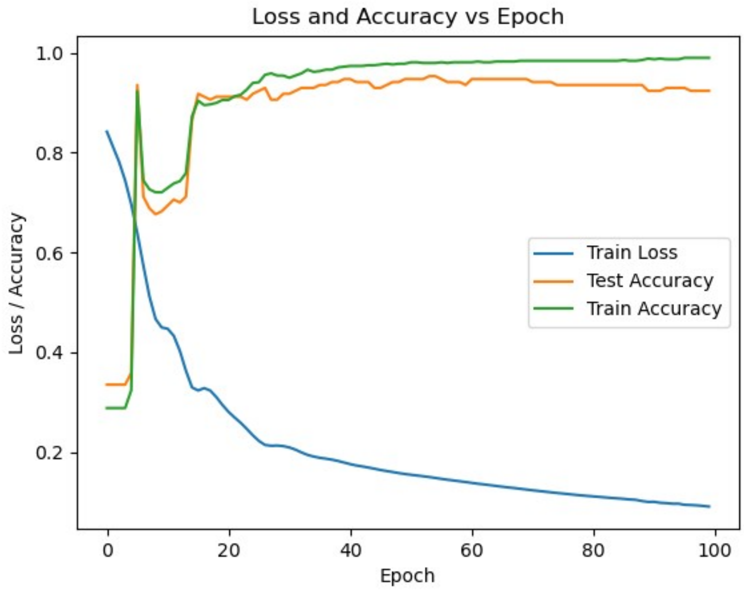


Fig 1.1.2 Chart for Loss and Accuracy vs Epoch

4. **Hyperparameter Tuning:** After training the initial model, we fine-tuned it by adjusting the hyperparameters to optimize its performance

    a. used techniques such as grid search or random search to find the best combination of hyperparameters

    b. The hyperparameters (optimizer, learn rate, loss function) of the best checkpoints were tuned to further improve model performance.

    c. Best performing tuned models were selected for further fine tuning over in-house dataset.

(Further plans)

5. ***Evaluation and testing***: After training the model and tuning the hyperparameters, we will evaluate its performance on a held-out test set to determine its effectiveness in predicting whether a user has fixed or rotational shifts. We will also calculate metrics such as accuracy, precision, recall, and F1 score to measure the performance of the model.

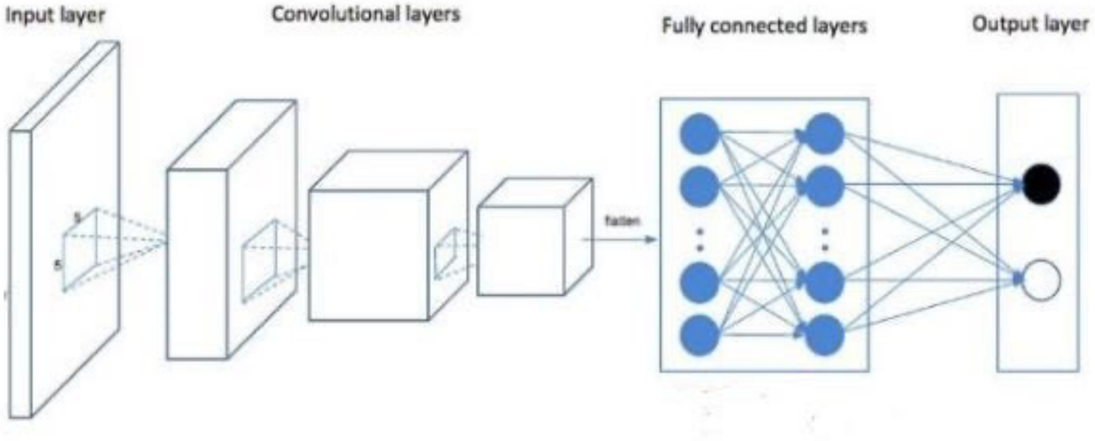


Fig 3.1.2 Binary Classifier Architecture

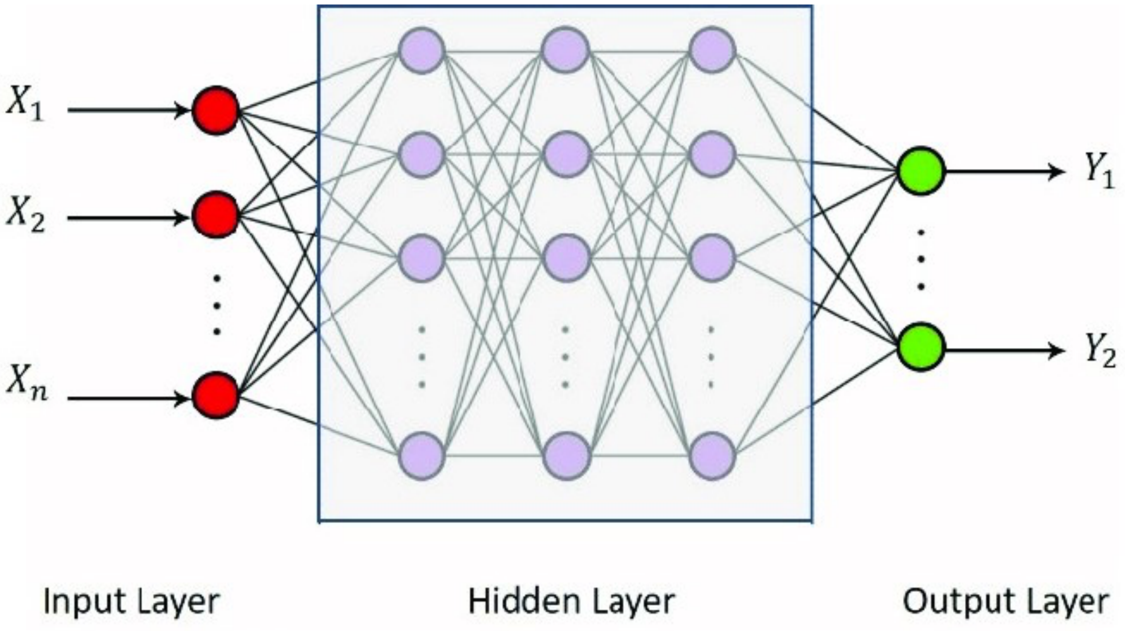


Fig 1.1.2 Neural Network Visual Representation

Ongoing development: As part of ongoing development, I plan to continue improving the model's performance by experimenting with different architectures, pre-processing techniques, and hyperparameters. I also plan to incorporate additional features such as user demographics and job roles to further improve the accuracy of the model. Additionally, I will continue to monitor the model's performance and update it as needed to ensure that it continues to provide accurate predictions.

1.1.3 Research Projects

- ***Website Development Deck***

  This initial assignment involves carrying out research and analysing the pipeline and flow of the website development process in order to figure out which activities in the flow could be automated or enhanced by AI.

  Such automation and support could improve developer productivity and efficiency while also tracking and enhancing code and the quality of the product. A presentation was produced and delivered to our department's upper management.

- ***Software Factory Implementation***

  The goal of this project is to build a software factory that will expedite the creation and deployment of AI and ML-based products and services. Based on the Scaled Agile Framework (SAFe), the software factory provides a standardised approach to software development, allowing the department to provide highquality goods and services in a timely and efficient way.

  Thorough research was conducted in order to establish the architecture, flow and components of the software factory and a presentation of the topic was created and presented to our superior of the department.

- ***Scaled Agile Framework (SAFe)***

  SAFe is a framework for scaling Agile development practices across multiple teams. The department has adopted the SAFe framework to ensure that its projects are delivered on time and with high quality, while also ensuring that teams are aligned and working together effectively

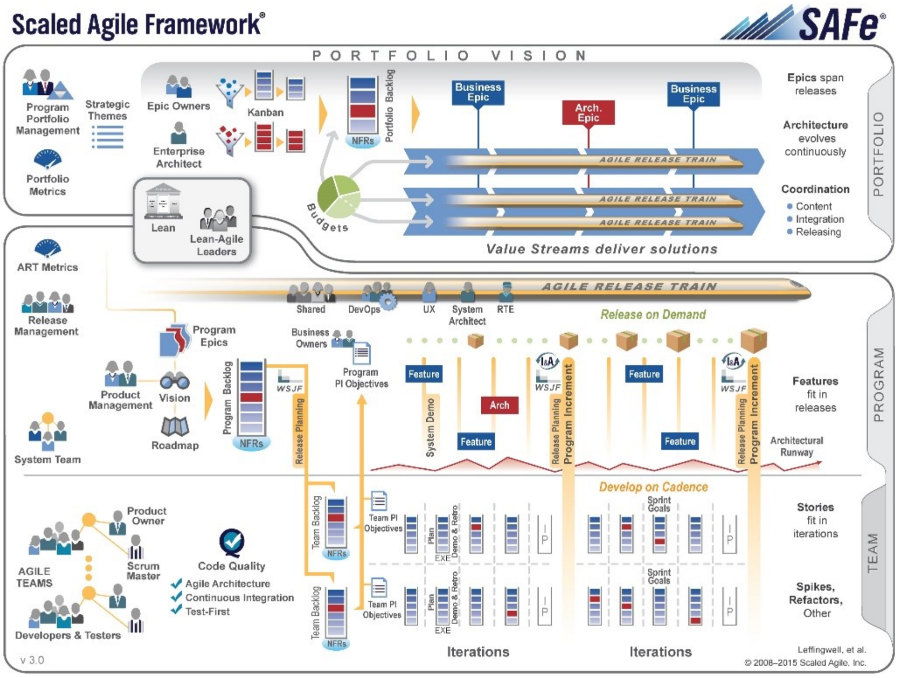


Fig 1.1.3 SAFe Architecture