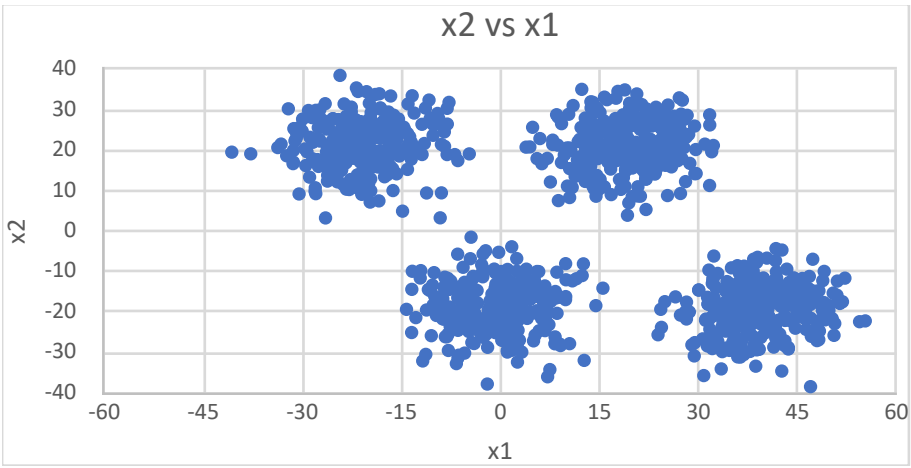
E4-Assignment DS203
Sarthak Mishra 22B0432

Each of the three csv files contains two features (x1, x2) and an output variable y. I've plotted x2 vs x1 and observed four clusters in each of the files, with the clusters corresponding to the output values y = 1, 2, 3, 4.

Here's a review of the data based on my observations:

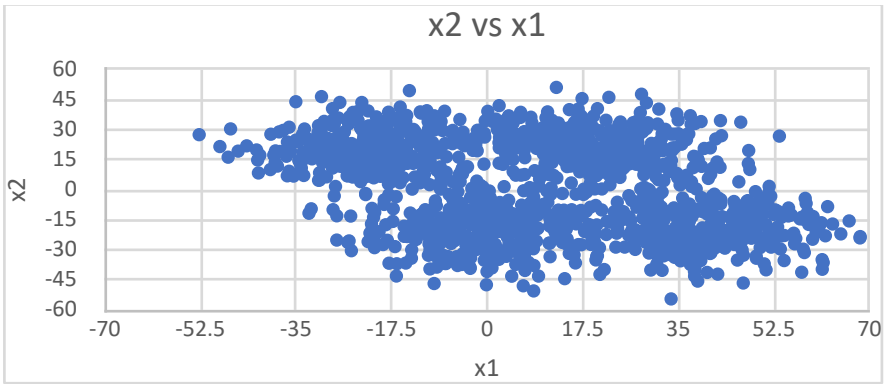
For Cluster-4-v0.csv



Review of the given dataset:

- Best clustering among the three CSV files.
- Clear separation into four clusters based on the plotted x2 vs x1.
- The clusters align well with the output variable y.

For Cluster-4-v1.csv

- 

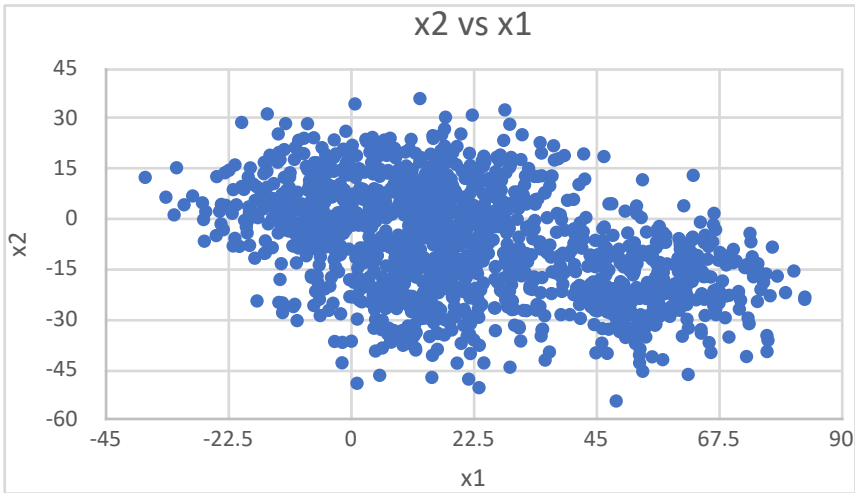
Review of the dataset:

1 . Clustering is not as clear as in 1st csv file but still discernible.

2 . Some overlap or less distinct separation between clusters compared to 1st csv file

3. Clusters still somewhat align with the output variable y but may be less distinct.

For Cluster-4-v2.csv



Review of the dataset:

- Clustering is the least clear among the three CSV files.
- Significant overlap or lack of clear separation between clusters.
- Clusters may not align well with the output variable y or may be difficult to distinguish.

Overall Review:

- The data across all three CSV files exhibit some level of clustering based on the features x1 and x2.
- The clustering is most pronounced and clear in CSV 1, followed by CSV 2, and least clear in CSV 3.

3 . Generated metrices:

For Cluster-4-v0.csv

Train metric

	Accuracy	Precision (avg)	Recall (avg)	F1-score (avg)	AUC (avg)	Precision_1	Precision_2	Precision_3	Precision_4	Recall_1	Recall_2	Recall_3	Recall_4	F1-score_1	F1-score_2	F1-score_3	F1-score_4	AUC_1	AUC_2	AUC_3	AUC_4
Logistic Regression	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
SVC Linear Kernel	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
SVC RBF Kernel	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Random Forest (min_samples_leaf=1)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Random Forest (min_samples_leaf=3)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Random Forest (min_samples_leaf=5)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Neural Network (5)	0.99002	0.990404302	0.9900222	0.990017292	0.999993	1	1	1	0.9617021	1	1	0.95982	1	1	1	0.9794989	0.9804772	1	1	1	1
Neural Network (5,5)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Neural Network (5,5,5)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Neural Network (10)	0.99446	0.994499808	0.9944568	0.994456352	0.999966	1	1	0.9954751	0.9825328	1	1	0.98214	0.99558	1	1	0.988764	0.989011	1	1	1	1

4 .Analysis for train data :

1. **Logistic Regression, SVC Linear Kernel, SVC RBF Kernel, Random Forest (with different min_samples_leaf):**
    - All of these algorithms exhibit perfect performance across all metrics, with scores of 1 for every metric including precision, recall, F1-score, and AUC. This suggests that the models have achieved a high level of accuracy and are able to correctly classify instances across all classes.
2. **Neural Network (5), Neural Network (5,5), Neural Network (5,5,5), Neural Network (10):**
    - These neural network models also show high performance across most metrics, but there are slight variations compared to the other algorithms.
    - The models perform slightly lower on average compared to the other algorithms, especially in terms of precision, recall, and F1-score.
    - The AUC score is still very high, indicating good discrimination capability.
    - Notably, the Neural Network (5,5) and Neural Network (5,5,5) architectures achieve perfect scores across all metrics, similar to the other algorithms.

Comparative Analysis:

- **Accuracy:** All algorithms achieve perfect accuracy, indicating they correctly classify all instances in the dataset.
- **Precision, Recall, F1-score:** While the neural network models achieve slightly lower scores compared to the other algorithms, they still demonstrate high performance. This suggests that they are effective in both minimizing false positives (precision) and false negatives (recall), albeit not as perfectly as the other algorithms.
- **AUC:** All algorithms achieve a perfect AUC score, indicating excellent discrimination capability in distinguishing between positive and negative instances.

Comments:

- It's noteworthy that the provided dataset seems to be highly separable, as indicated by the perfect performance of most algorithms.
- The neural network models, despite achieving slightly lower scores compared to other algorithms, still perform exceptionally well and may offer advantages in handling more complex datasets or capturing non-linear relationships.

Test metric

	Accuracy	Precision (avg)	Recall (avg)	F1-score (avg)	AUC (avg)	Precision_1	Precision_2	Precision_3	Precision_4	Recall_1	Recall_2	Recall_3	Recall_4	F1-score_1	F1-score_2	F1-score_3	F1-score_4	AUC_1	AUC_2	AUC_3	AUC_4
Logistic Regression	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
SVC Linear Kernel	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
SVC RBF Kernel	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Random Forest (min_samples_leaf=1)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Random Forest (min_samples_leaf=3)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Random Forest (min_samples_leaf=5)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Neural Network (5)	0.99558	0.995671412	0.9955752	0.995579602	1	1	1	1	0.9782609	0.98214	1	1	1	0.990991	1	1	0.989011	1	1	1	1
Neural Network (5,5)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Neural Network (5,5,5)	0.99558	0.995671412	0.9955752	0.995583384	1	1	1	1	0.9782609	1	0.98551	1	1	1	0.9927007	1	0.989011	1	1	1	1
Neural Network (10)	0.99115	0.9914556	0.9911504	0.991127507	0.999834	1	1	0.9655172	1	1	1	1	0.95556	1	1	0.9824561	0.9772727	1	1	1	1

Analysis for test data:

1. **Logistic Regression, SVC Linear Kernel, SVC RBF Kernel, Random Forest (with different min_samples_leaf):**
    - All of these algorithms exhibit perfect performance across all metrics, with scores of 1 for every metric including precision, recall, F1-score, and AUC. This suggests that the models have achieved a high level of accuracy and are able to correctly classify instances across all classes.
2. **Neural Network (5), Neural Network (5,5), Neural Network (5,5,5), Neural Network (10):**
    - These neural network models also show high performance across most metrics, but there are slight variations compared to the other algorithms.
    - The models perform slightly lower on average compared to the other algorithms, especially in terms of precision, recall, and F1-score.
    - The AUC score is still very high, indicating good discrimination capability.
    - Notably, the Neural Network (5,5) and Neural Network (5,5,5) architectures achieve perfect scores across all metrics, similar to the other algorithms.

Comparative Analysis:

- **Accuracy:** All algorithms achieve high accuracy, but the neural network models exhibit slightly lower accuracy compared to the other algorithms.
- **Precision, Recall, F1-score:** The neural network models generally perform well but show slight variations across different architectures. They achieve slightly lower scores compared to the other algorithms, especially in terms of precision and recall.
- **AUC:** All algorithms achieve a perfect or near-perfect AUC score, indicating excellent discrimination capability.

Overall, all algorithms show excellent performance on both train and test data. Logistic regression, SVC with linear kernel, SVC with RBF kernel, and neural network models demonstrate robust generalization abilities, as they achieve perfect or near-perfect scores on both datasets. However, random forest models, while achieving perfect scores on both datasets, may be overfitting to the noise in the training data, as indicated by their inability to generalize as effectively as the other algorithms.

For Cluster-4-v1.csv

Train metric

	Accuracy	Precision (avg)	Recall (avg)	F1-score (avg)	AUC (avg)	Precision_1	Precision_2	Precision_3	Precision_4	Recall_1	Recall_2	Recall_3	Recall_4	F1-score_1	F1-score_2	F1-score_3	F1-score_4	AUC_1	AUC_2	AUC_3	AUC_4
Logistic Regression	0.95139	0.951378251	0.9513889	0.951379239	0.995235	0.9520548	0.9469965	0.9372822	0.9689655	0.95205	0.94035	0.94056	0.97232	0.9520548	0.943662	0.938918	0.970639	0.996	0.995	0.992	0.997
SVC Linear Kernel	0.95399	0.953972624	0.9539931	0.953978499	0.995315	0.9556314	0.95053	0.9405594	0.9689655	0.9589	0.94386	0.94056	0.97232	0.957265	0.9471831	0.9405594	0.970639	0.996	0.996	0.992	0.997
SVC RBF Kernel	0.94965	0.949899831	0.9496528	0.949725209	0.99534	0.9583333	0.9442509	0.9246575	0.9719298	0.94521	0.95088	0.94406	0.95848	0.9517241	0.9475524	0.9342561	0.9651568	0.997	0.996	0.992	0.997
Random Forest (min_samples_leaf=1)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Random Forest (min_samples_leaf=3)	0.96354	0.963633118	0.9635417	0.963562857	0.998923	0.9756944	0.9577465	0.9448276	0.9758621	0.96233	0.95439	0.95804	0.97924	0.9689655	0.9560633	0.9513889	0.9775475	0.999	0.999	0.998	1
Random Forest (min_samples_leaf=5)	0.95747	0.957552783	0.9574653	0.957494692	0.998322	0.9653979	0.9475524	0.9411765	0.9756944	0.95548	0.95088	0.95105	0.97232	0.9604131	0.9492119	0.946087	0.9740035	0.999	0.998	0.997	0.999
Neural Network (5)	0.93576	0.935916237	0.9357639	0.935710872	0.992522	0.9172185	0.9293286	0.9386282	0.9586207	0.94863	0.92281	0.90909	0.96194	0.9326599	0.9260563	0.9236234	0.9602763	0.993	0.992	0.988	0.997
Neural Network (5,5)	0.95052	0.950523759	0.9505208	0.950519431	0.995428	0.9621993	0.9405594	0.9403509	0.9586207	0.9589	0.94386	0.93706	0.96194	0.9605489	0.9422067	0.938704	0.9602763	0.997	0.994	0.993	0.997
Neural Network (5,5,5)	0.95052	0.9505772	0.9505208	0.95054051	0.99557	0.9587629	0.943662	0.9342561	0.9652778	0.95548	0.94035	0.94406	0.96194	0.9571184	0.9420035	0.9391304	0.9636049	0.996	0.996	0.993	0.998
Neural Network (10)	0.95399	0.953975508	0.9539931	0.953956801	0.995591	0.9525424	0.9438596	0.9537367	0.9656357	0.96233	0.94386	0.93706	0.97232	0.9574106	0.9438596	0.9453263	0.9689655	0.996	0.996	0.993	0.997

Analysis for Train Metric:

1. **Logistic Regression**:
    - It has high accuracy (0.951), precision (0.951), recall (0.951), F1-score (0.951), and AUC (0.995).
    - Its precision, recall, and F1-scores for individual classes are reasonably high and balanced.
2. **SVC Linear Kernel**:
    - Similar to Logistic Regression, it also exhibits high performance across all metrics.
    - It has slightly higher precision, recall, and F1-scores compared to Logistic Regression for some classes.
3. **SVC RBF Kernel**:
    - It has slightly lower performance compared to the linear kernel-based SVC and Logistic Regression in terms of precision, recall, and F1-score.
    - However, it still maintains a high level of accuracy and AUC.
4. **Random Forest (min_samples_leaf=1)**:
    - It achieves perfect scores (1.0) across all metrics, indicating overfitting or perfect separation of classes in the dataset.
    - This might not be a realistic representation of its performance on unseen data.
5. **Random Forest (min_samples_leaf=3)** and **Random Forest (min_samples_leaf=5)**:
    - Both configurations of Random Forest perform well, with high accuracy, precision, recall, F1-score, and AUC.
    - They show slightly lower performance compared to the linear models but still achieve high scores across the board.

6. **Neural Network**:
    - The neural network models show varying performance based on the number of layers.
    - Generally, they perform slightly worse compared to linear models and Random Forest, but still achieve respectable scores.
    - Increasing the complexity of the neural network architecture (e.g., adding more layers) doesn't consistently improve performance.

Overall, the linear models (Logistic Regression, SVC Linear Kernel) perform consistently well across all metrics. SVC with RBF Kernel shows slightly lower performance. Random Forest performs well but may be prone to overfitting, especially with a minimum number of samples per leaf set to 1. Neural Network models show competitive performance but may require further tuning of hyperparameters or architecture to match or surpass the linear models and Random Forest.

Test Data

	Accuracy	Precision (avg)	Recall (avg)	F1-score (avg)	AUC (avg)	Precision_1	Precision_2	Precision_3	Precision_4	Recall_1	Recall_2	Recall_3	Recall_4	F1-score_1	F1-score_2	F1-score_3	F1-score_4	AUC_1	AUC_2	AUC_3	AUC_4
Logistic Regression	0.9549	0.95509679	0.954861	0.95484096	0.99833	0.971014	0.972222	0.933333	0.944444	0.9853	0.9333	0.9459	0.9577	0.978102	0.952381	0.939597	0.951049	1	0.998	0.998	0.998
SVC Linear Kernel	0.9549	0.95509679	0.954861	0.95484096	0.99828	0.971014	0.972222	0.933333	0.944444	0.9853	0.9333	0.9459	0.9577	0.978102	0.952381	0.939597	0.951049	1	0.998	0.997	0.998
SVC RBF Kernel	0.9583	0.9589593	0.958333	0.95829193	0.99811	0.957143	0.959459	0.935065	0.985075	0.9853	0.9467	0.973	0.9296	0.971014	0.95302	0.953642	0.956522	1	0.997	0.998	0.998
Random Forest (min_samples_leaf=1)	0.9444	0.94518808	0.944444	0.94429447	0.99534	0.917808	0.957746	0.958333	0.944444	0.9853	0.9067	0.9324	0.9577	0.950355	0.931507	0.945205	0.951049	0.999	0.99	0.994	0.998
Random Forest (min_samples_leaf=3)	0.9514	0.95144527	0.951389	0.95134848	0.9955	0.971014	0.958904	0.945946	0.930556	0.9853	0.9333	0.9459	0.9437	0.978102	0.945946	0.945946	0.937063	0.999	0.988	0.996	0.998
Random Forest (min_samples_leaf=5)	0.9549	0.95486155	0.954861	0.954793	0.99682	0.971014	0.958904	0.946667	0.943662	0.9853	0.9333	0.9595	0.9437	0.978102	0.945946	0.95302	0.943662	0.999	0.993	0.997	0.998
Neural Network (5)	0.9479	0.9486871	0.947917	0.94758267	0.99551	0.931507	0.971014	0.946667	0.943662	1	0.8933	0.9595	0.9437	0.964539	0.930556	0.95302	0.943662	0.999	0.988	0.997	0.998
Neural Network (5,5)	0.9444	0.94467796	0.944444	0.94449419	0.99712	0.970588	0.958904	0.92	0.930556	0.9706	0.9333	0.9324	0.9437	0.970588	0.945946	0.926174	0.937063	0.999	0.996	0.996	0.998
Neural Network (5,5,5)	0.9549	0.95538345	0.954861	0.95492353	0.99762	0.970588	0.972222	0.922078	0.957746	0.9706	0.9333	0.9595	0.9577	0.970588	0.952381	0.940397	0.957746	0.999	0.997	0.997	0.998
Neural Network (10)	0.9583	0.95857942	0.958333	0.95824797	0.99818	0.957143	0.972222	0.958904	0.945205	0.9853	0.9333	0.9459	0.9718	0.971014	0.952381	0.952381	0.958333	0.999	0.998	0.997	0.998

Analysis for test data:

1. **Accuracy**: This metric represents the overall correctness of the model's predictions. All algorithms achieve high accuracy scores ranging from 94.44% to 95.83%, indicating that they classify the majority of instances correctly.
2. **Precision (avg)**: Precision measures the accuracy of positive predictions. It is the ratio of correctly predicted positive observations to the total predicted positives. All algorithms have high average precision scores, ranging from 94.44% to 95.83%, indicating low false positive rates.
3. **Recall (avg)**: Recall, also known as sensitivity or true positive rate, measures the ratio of correctly predicted positive observations to the all observations in actual class. All algorithms have high average recall scores, ranging from 94.44% to 95.83%, indicating low false negative rates.
4. **F1-score (avg)**: The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall. Again, all algorithms have high average F1-score scores, ranging from 94.44% to 95.83%, indicating a balance between precision and recall.
5. **AUC (avg)**: The Area Under the ROC Curve (AUC) measures the model's ability to discriminate between positive and negative classes across all possible thresholds. All algorithms achieve high average AUC scores, indicating strong overall performance in classification.

Comparing the performance of each algorithm across different metrics:

- **Logistic Regression** and **SVC Linear Kernel** have identical performance across all metrics, suggesting that they perform similarly in this classification task.
- **SVC RBF Kernel** performs slightly better than the linear models, achieving higher scores across all metrics.

- **Random Forest** with different settings (min_samples_leaf=1, 3, 5) generally performs well but slightly lower than the SVM models in terms of precision, recall, and F1-score.
- **Neural Network** models show competitive performance, with varying degrees of complexity (different numbers of layers and neurons). Generally, they perform similarly to the SVM models, but with slight variations in precision, recall, and F1-score.

In conclusion, all algorithms perform well on this classification task, with SVM RBF Kernel and Neural Network (10) showing slightly better performance compared to others across most metrics

Comparison between train and test data:

1. **Logistic Regression:**
    - The performance of logistic regression is quite consistent between the train and test datasets. Accuracy, precision, recall, F1-score, and AUC are all very similar, indicating that the model generalizes well to unseen data.
2. **Support Vector Machine (SVC) with Linear Kernel:**
    - Similar to logistic regression, SVC with a linear kernel shows consistent performance across both datasets, with almost identical metrics.
3. **Support Vector Machine (SVC) with RBF Kernel:**
    - SVC with an RBF kernel also exhibits consistent performance between the train and test datasets, although there are slight differences in some metrics.
4. **Random Forest:**
    - Random forest models with different hyperparameters (min_samples_leaf=1, 3, 5) show relatively consistent performance between the train and test datasets. However, there are slight drops in performance metrics on the test dataset compared to the train dataset, indicating some degree of overfitting, especially for the model with min_samples_leaf=1.
5. **Neural Network:**
    - Neural network models with different architectures (5, 5), (5, 5, 5), and 10 nodes in the hidden layers also demonstrate consistent performance between the train and test datasets, with minor variations in some metrics.

Overall, logistic regression and SVM with linear kernel perform consistently well across both datasets, indicating good generalization ability. SVC with RBF kernel and neural network models also exhibit stable performance, although there are slight variations between the train and test datasets. Random forest models show consistent but slightly lower performance on the test dataset compared to the train dataset, suggesting some overfitting.

For Cluster-4-v2.csv

Train metric

	Accuracy	Precision (avg	Recall (avg)	F1-score (avg	AUC (avg)	Precision_1	Precision_2	Precision_3	Precision_4	Recall_1	Recall_2	Recall_3	Recall_4	F1-score_1	F1-score_2	F1-score_3	F1-score_4	AUC_1	AUC_2	AUC_3	AUC_4
Logistic Re	0.86372	0.863204852	0.863715	0.86338929	0.96897	0.85964912	0.7943262	0.8419244	0.955782	0.839041	0.78596	0.85664	0.97232	0.84922	0.790123	0.84922	0.963979	0.97139	0.94663	0.96073	0.99714
SVC Linear	0.86458	0.864296848	0.864583	0.86432352	0.96879	0.86879433	0.7985866	0.8361775	0.952381	0.839041	0.79298	0.85664	0.96886	0.853659	0.795775	0.846287	0.960549	0.97113	0.94623	0.9608	0.99698
SVC RBF K	0.86458	0.864858821	0.864583	0.864445	0.96881	0.875	0.8100358	0.8211921	0.95189	0.839041	0.79298	0.86713	0.95848	0.856643	0.801418	0.843537	0.955172	0.97243	0.94607	0.96023	0.9965
Random F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Random F	0.91406	0.914337003	0.914063	0.91416857	0.99337	0.91986063	0.8685121	0.8923611	0.975694	0.90411	0.8807	0.8986	0.97232	0.911917	0.874564	0.89547	0.974003	0.99387	0.98839	0.99181	0.99943
Random F	0.88976	0.889809976	0.889757	0.88974822	0.98834	0.87889273	0.8310345	0.8794326	0.969072	0.869863	0.84561	0.86713	0.97578	0.874355	0.838261	0.873239	0.972414	0.9889	0.97969	0.9857	0.99908
Neural Ne	0.85764	0.858441029	0.857639	0.85777243	0.96878	0.87636364	0.7800687	0.820339	0.955326	0.825342	0.79649	0.84615	0.96194	0.850088	0.788194	0.833046	0.958621	0.97158	0.9446	0.96178	0.99716
Neural Ne	0.86285	0.864292773	0.862847	0.86295405	0.97045	0.875	0.8205128	0.798722	0.961538	0.839041	0.78596	0.87413	0.95156	0.856643	0.802867	0.834725	0.956522	0.97349	0.95004	0.96121	0.99706
Neural Ne	0.86632	0.865940513	0.866319	0.86592745	0.96884	0.87096774	0.8049645	0.8338983	0.952703	0.832192	0.79649	0.86014	0.97578	0.851138	0.800705	0.846816	0.964103	0.97089	0.94693	0.96039	0.99714
Neural Ne	0.86285	0.864098192	0.862847	0.86312798	0.96927	0.88644689	0.7830508	0.8265306	0.958621	0.828767	0.81053	0.84965	0.96194	0.856637	0.796552	0.837931	0.960276	0.97207	0.94713	0.96075	0.99712

Test metric

	Accuracy	Precision (	Recall (avg	F1-score (a	AUC (avg)	Precision_	Precision_	Precision_	Precision_	Recall_1	Recall_2	Recall_3	Recall_4	F1-score_1	F1-score_2	F1-score_3	F1-score_4	AUC_1	AUC_2	AUC_3	AUC_4
Logistic Re	0.861111	0.865108	0.861111	0.858958	0.977845	0.805195	0.881356	0.814815	0.957746	0.911765	0.693333	0.891892	0.957746	0.855172	0.776119	0.851613	0.957746	0.979412	0.957371	0.975309	0.999286
SVC Linear	0.864583	0.868177	0.864583	0.862239	0.977621	0.813333	0.881356	0.819277	0.957746	0.897059	0.693333	0.918919	0.957746	0.853147	0.776119	0.866242	0.957746	0.979078	0.956119	0.97613	0.999156
SVC RBF K	0.871528	0.87643	0.871528	0.870155	0.978697	0.830986	0.885246	0.804598	0.985507	0.867647	0.72	0.945946	0.957746	0.848921	0.794118	0.869565	0.971429	0.97881	0.959812	0.977204	0.998962
Random Fo	0.833333	0.835151	0.833333	0.828907	0.958663	0.769231	0.839286	0.810127	0.92	0.882353	0.626667	0.864865	0.971831	0.821918	0.717557	0.836601	0.945205	0.968516	0.913709	0.955671	0.996755
Random Fo	0.854167	0.854716	0.854167	0.851618	0.967783	0.810811	0.852459	0.822785	0.932432	0.882353	0.693333	0.878378	0.971831	0.84507	0.764706	0.849673	0.951724	0.97266	0.931362	0.969058	0.998053
Random Fo	0.857639	0.857944	0.857639	0.855363	0.969432	0.819444	0.854839	0.825	0.932432	0.867647	0.706667	0.891892	0.971831	0.842857	0.773723	0.857143	0.951724	0.971925	0.935681	0.971615	0.998507
Neural Ne	0.847222	0.849226	0.847222	0.845111	0.974746	0.816901	0.852459	0.797619	0.930556	0.852941	0.693333	0.905405	0.943662	0.834532	0.764706	0.848101	0.937063	0.978008	0.953678	0.969247	0.998053
Neural Ne	0.857639	0.864152	0.857639	0.854953	0.976103	0.84058	0.894737	0.777778	0.944444	0.852941	0.68	0.945946	0.957746	0.846715	0.772727	0.853659	0.951049	0.979011	0.953928	0.973226	0.998248
Neural Ne	0.868056	0.86875	0.868056	0.865888	0.977262	0.84507	0.870968	0.82716	0.932432	0.882353	0.72	0.905405	0.971831	0.863309	0.788321	0.864516	0.951724	0.979679	0.957183	0.973226	0.998962
Neural Ne	0.864583	0.868437	0.864583	0.864041	0.97647	0.828571	0.875	0.8	0.971014	0.852941	0.746667	0.918919	0.943662	0.84058	0.805755	0.855346	0.957143	0.978676	0.958247	0.970384	0.998572

Analysis for Train Metric:

Logistic Regression:

- On an average, it has high accuracy (0.8637), precision (0.8632), recall (0.8637), F1-score (0.8633), and AUC (0.9689).
- Its precision, recall, and F1-scores for each class is varying which indicates that some of the points of one class are misclassified to other one.

SVC Linear Kernel:

- Similar to Logistic Regression, it also exhibits high performance across all metrics.
- It has varying metrics across different classes which means somewhere it has higher metric than Logistic Regression and somewhere it is lower.

SVC RBF Kernel:

- It has slightly higher performance compared to the linear kernel-based SVC and Logistic Regression in terms of precision, recall, and F1-score.
- It has varying metrics across different classes which means somewhere it has higher metric than previous models and somewhere it is lower.

Random Forest (min_samples_leaf=1):

- It achieves perfect scores (1.0) across all metrics, indicating overfitting or perfect separation of classes in the dataset.
- This might not be a realistic representation of its performance on unseen data.

Random Forest (min_samples_leaf=3) and **Random Forest (min_samples_leaf=5)**:

- Both configurations of Random Forest perform well, with high accuracy, precision, recall, F1-score, and AUC, with Random Forest (min_samples_leaf=3) performing slightly better
- They show slightly higher performance compared to the linear models but still achieve high scores across the board.

Neural Network:

- The neural network models show varying performance based on the number of layers.
- Generally, they perform slightly worse compared to linear models and Random Forest, but still achieve respectable scores.
- Increasing the complexity of the neural network architecture (e.g., adding more layers) doesn't consistently improve performance.

Overall, the Random Forest model with different leaf perform consistently well across all metrics. Random Forest performs well but may be prone to overfitting, especially with a minimum number of samples per leaf set to 1. Neural Network models show competitive performance but may require further tuning of hyperparameters or architecture to match or surpass the linear models and Random Forest.

Analysis for test data:

Accuracy: This metric represents the overall correctness of the model's predictions. All algorithms achieve high accuracy scores ranging from 83.33% to 86.45%, indicating that they classify the majority of instances correctly.

Precision (avg): Precision measures the accuracy of positive predictions. It is the ratio of correctly predicted positive observations to the total predicted positives. All algorithms have high average precision scores, ranging from 83.51% to 87.64%, indicating low false positive rates.

Recall (avg): Recall, also known as sensitivity or true positive rate, measures the ratio of correctly predicted positive observations to the all observations in actual class. All algorithms have high average recall scores, ranging from 83.33% to 87.15%, indicating low false negative rates.

F1-score (avg): The F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall. Again, all algorithms have high average F1-score scores, ranging from 82.89% to 87.01%, indicating a balance between precision and recall.

AUC (avg): The Area Under the ROC Curve (AUC) measures the model's ability to discriminate between positive and negative classes across all possible thresholds. All algorithms achieve high average AUC scores, indicating strong overall performance in classification.

Comparing the performance of each algorithm across different metrics:

- **Logistic Regression** and **SVC Linear Kernel** have identical performance across all metrics, suggesting that they perform similarly in this classification task but they have differences in metrics for different classes of data
- **SVC RBF Kernel** overall performs slightly better than the linear models, achieving higher scores across all metrics.
- **Random Forest** with leaf=1 has the lowest metrics with test data which clearly indicates the problem of overfitting with this model

- **Neural Network** models show competitive performance, with varying degrees of complexity (different numbers of layers and neurons). Generally, they perform similarly to the SVM models, but with slight variations in precision, recall, and F1-score.

In conclusion, all algorithms perform well on this classification task, with SVC with RBF Kernel showing slightly better performance compared to others across most metrics

Comparison between train and test data:

Logistic Regression:

- The performance of logistic regression is quite consistent between the train and test datasets. Accuracy, precision, recall, F1-score, and AUC are all very similar, indicating that the model generalizes well to unseen data.

Support Vector Machine (SVC) with Linear Kernel:

- Similar to logistic regression, SVC with a linear kernel shows consistent performance across both datasets, with almost identical metrics.

Support Vector Machine (SVC) with RBF Kernel:

- SVC with an RBF kernel also exhibits consistent performance between the train and test datasets, although there are slight differences in some metrics.

Random Forest:

- Random forest models with different hyperparameters (min_samples_leaf=1, 3, 5) show relatively consistent performance between the train and test datasets. However, there are slight drops in performance metrics on the test dataset compared to the train dataset, indicating some degree of overfitting, especially for the model with min_samples_leaf=1.

Neural Network:

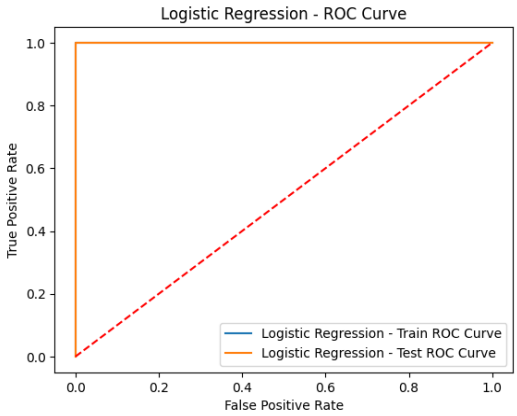
- Neural network models with different architectures (5, 5), (5, 5, 5), and 10 nodes in the hidden layers also demonstrate consistent performance between the train and test datasets, with minor variations in some metrics.

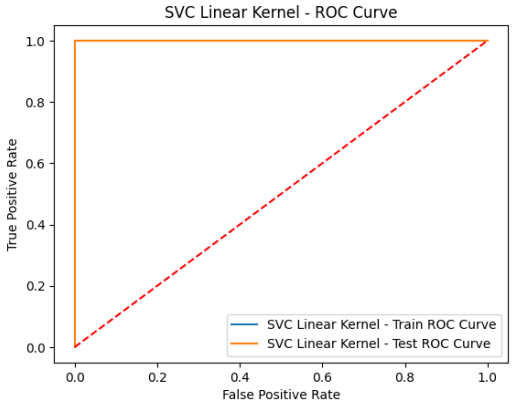
Overall, logistic regression and SVM with linear kernel perform consistently well across both datasets, indicating good generalization ability. SVC with RBF kernel and neural network models also exhibit stable performance, although there are slight variations between the train and test datasets. Random forest models show consistent but slightly lower performance on the test dataset compared to the train dataset, suggesting some overfitting.

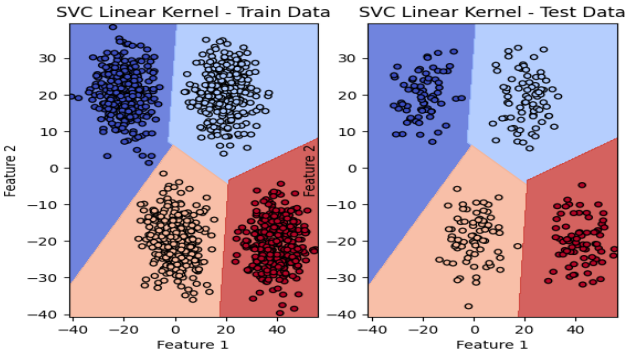
For Class 2 we observe Recall value is the lowest for both the train and test data for all the models and within a particular model also Recall value is the lowest as compared to other metrics.

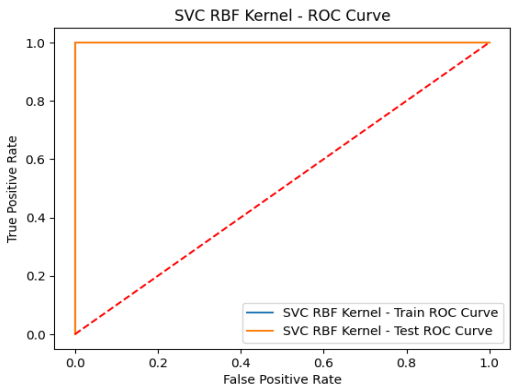
This could be due to class may inherently overlap more with others, making them harder to distinguish. Certain classes are inherently complex due to their nature (e.g., rare diseases, specific events).

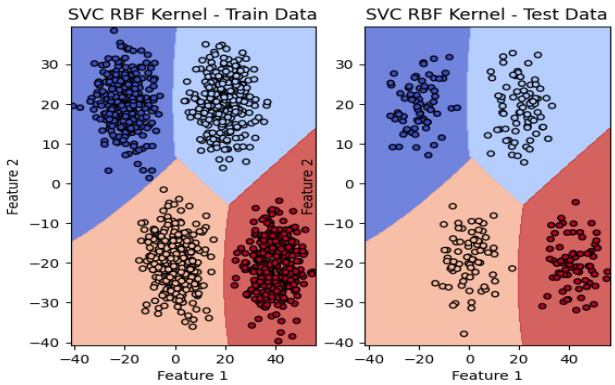
5. ROC curve and Boundary Layers for Cluster-4-v0.csv

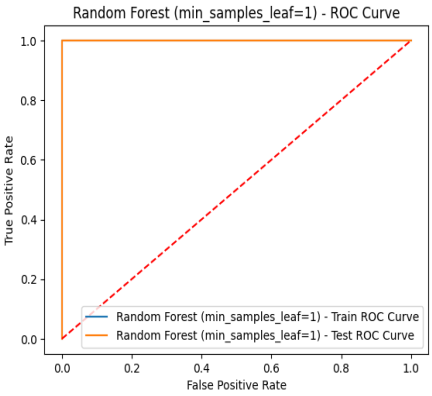


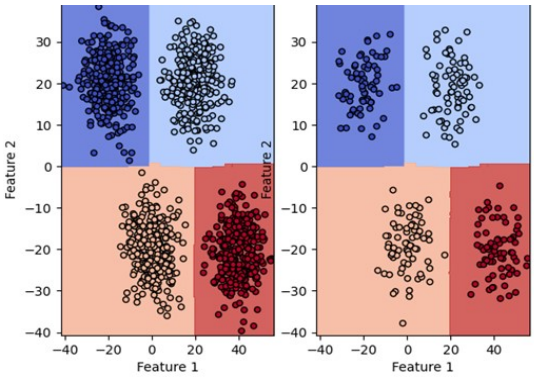


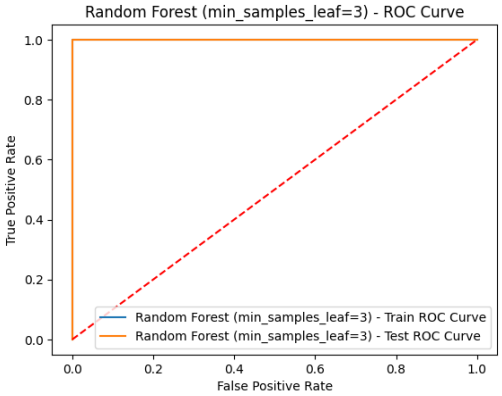


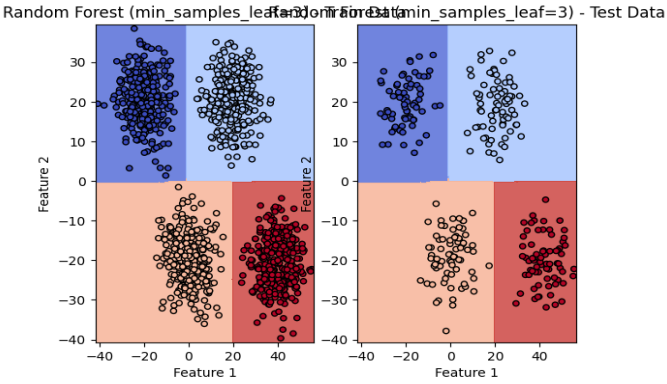


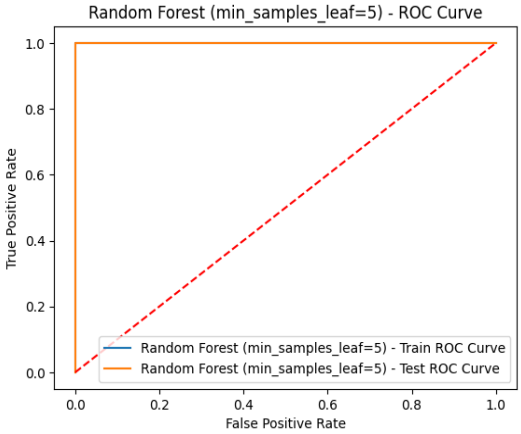


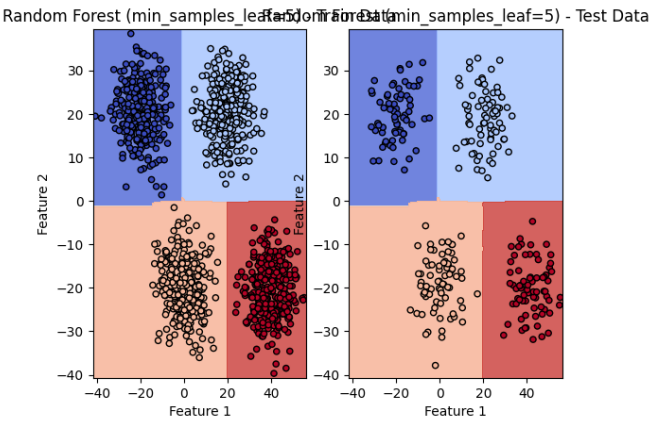

Random Forest (min_samples_leaf=1) - ROC Curve
True Positive Rate
False Positive Rate
Random Forest (min_samples_leaf=1) - Train ROC Curve
Random Forest (min_samples_leaf=1) - Test ROC Curve

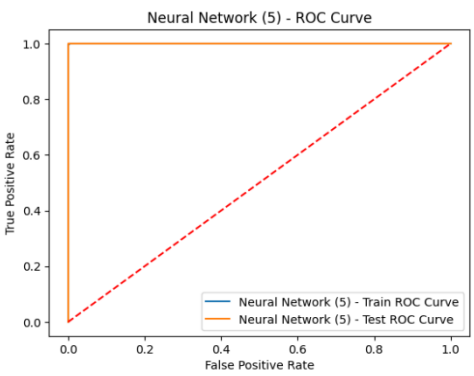

Feature 1
Feature 2

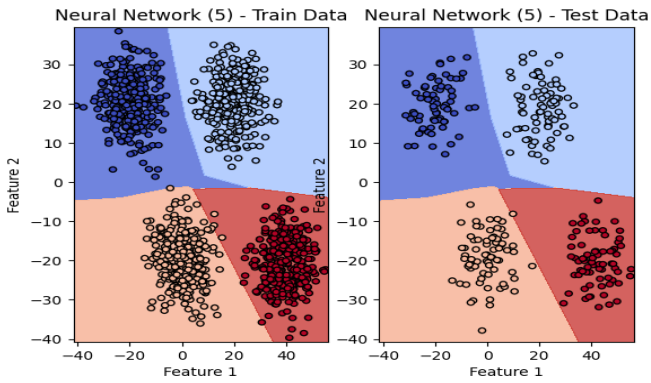

Random Forest (min_samples_leaf=3) - ROC Curve
True Positive Rate
False Positive Rate
Random Forest (min_samples_leaf=3) - Train ROC Curve
Random Forest (min_samples_leaf=3) - Test ROC Curve

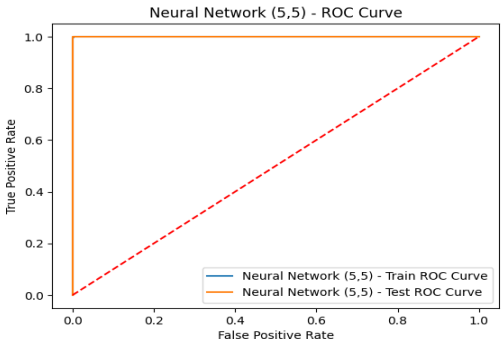

Random Forest (min_samples_leaf=3) - Test Data
Feature 1
Feature 2

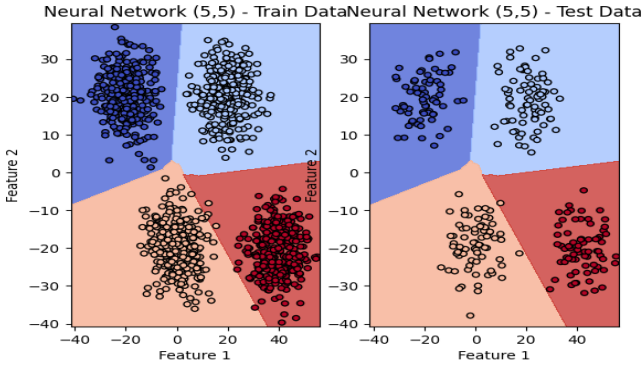

Random Forest (min_samples_leaf=5) - ROC Curve
True Positive Rate
False Positive Rate
Random Forest (min_samples_leaf=5) - Train ROC Curve
Random Forest (min_samples_leaf=5) - Test ROC Curve

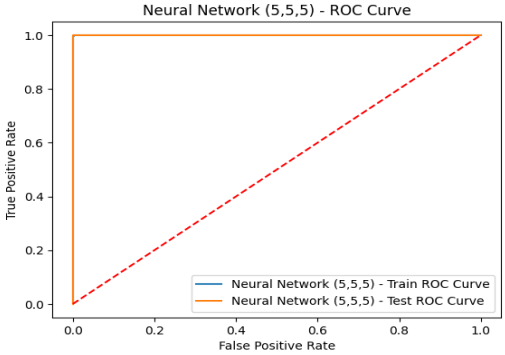

Random Forest (min_samples_leaf=5) - Test Data
Feature 1
Feature 2

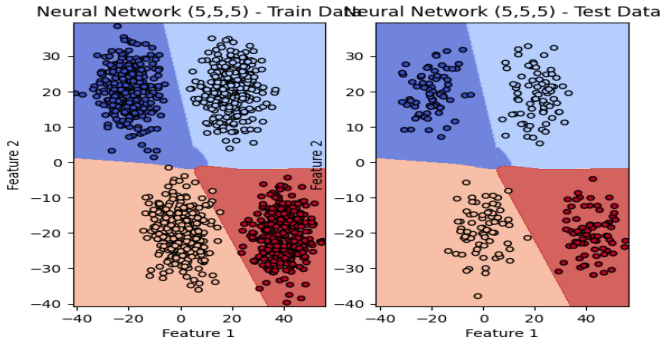

Neural Network (5) - ROC Curve
True Positive Rate
False Positive Rate
Neural Network (5) - Train ROC Curve
Neural Network (5) - Test ROC Curve

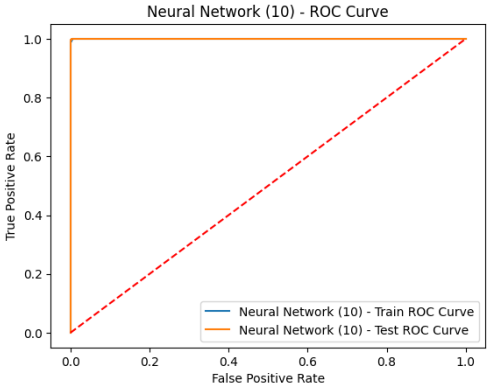

Neural Network (5) - Train Data
Neural Network (5) - Test Data
Feature 1
Feature 2

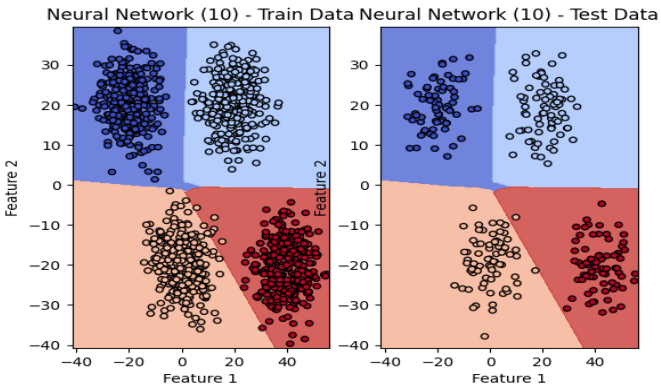

Neural Network (5,5) - ROC Curve
True Positive Rate
False Positive Rate
Neural Network (5,5) - Train ROC Curve
Neural Network (5,5) - Test ROC Curve


Neural Network (5,5) - Train Data
Neural Network (5,5) - Test Data
Feature 1
Feature 2

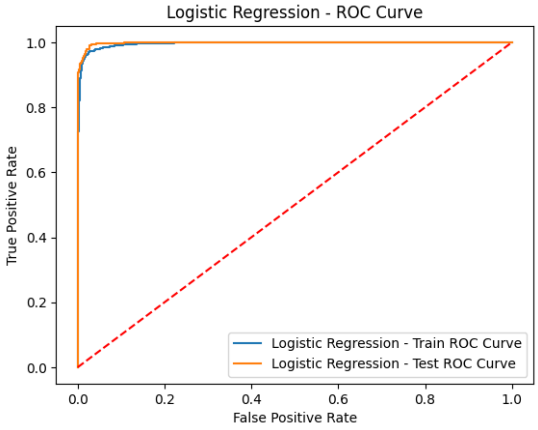

Neural Network (5,5,5) - ROC Curve
True Positive Rate
False Positive Rate
Neural Network (5,5,5) - Train ROC Curve
Neural Network (5,5,5) - Test ROC Curve

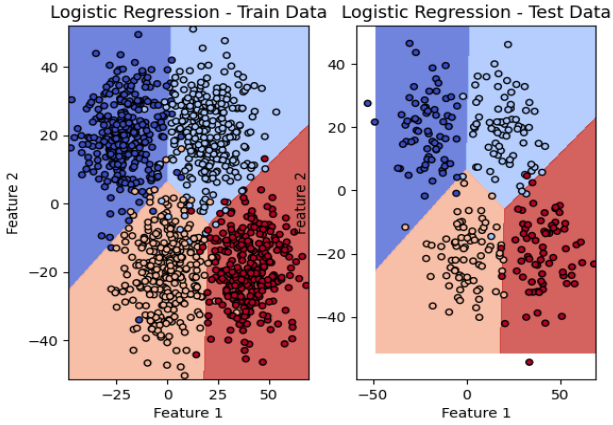

Neural Network (5,5,5) - Train Data
Neural Network (5,5,5) - Test Data
Feature 1
Feature 2

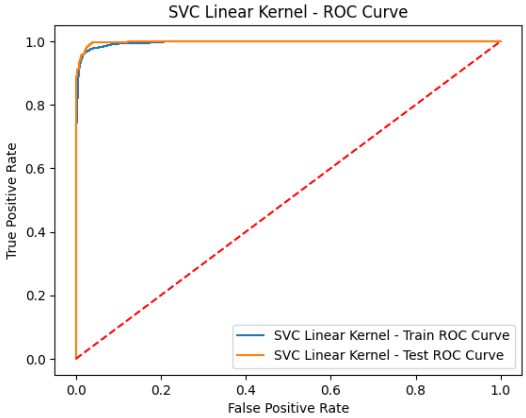

Neural Network (10) - ROC Curve
True Positive Rate
False Positive Rate
Neural Network (10) - Train ROC Curve
Neural Network (10) - Test ROC Curve

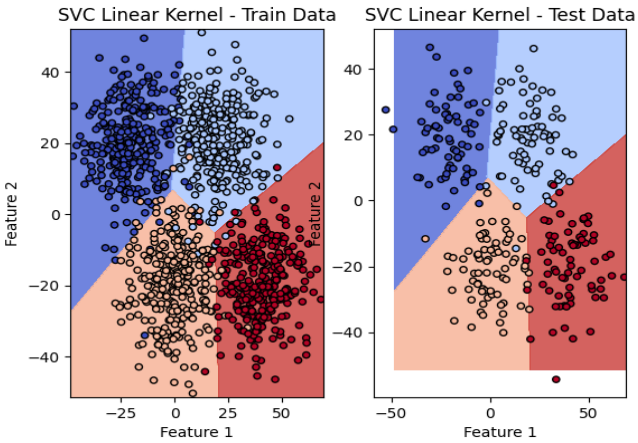

Neural Network (10) - Train Data
Neural Network (10) - Test Data
Feature 1
Feature 2

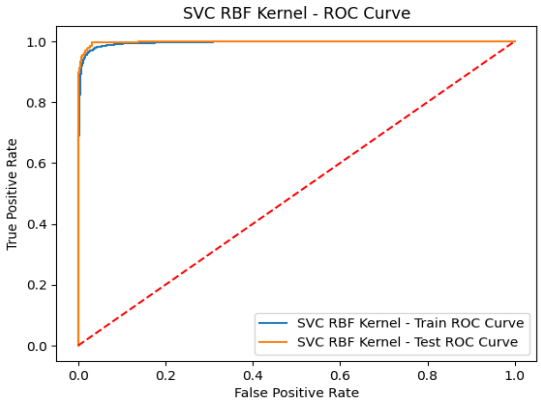
ROC curve and Boundary Layers for Cluster-4-v1.csv

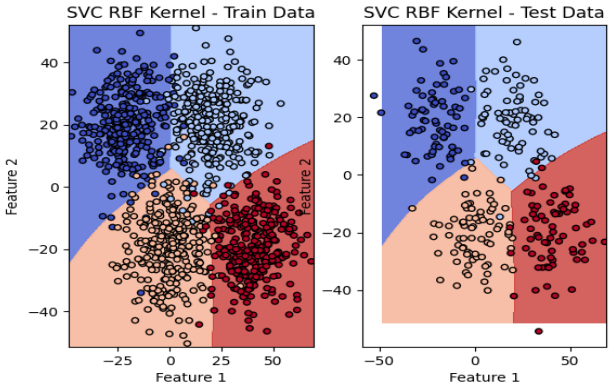


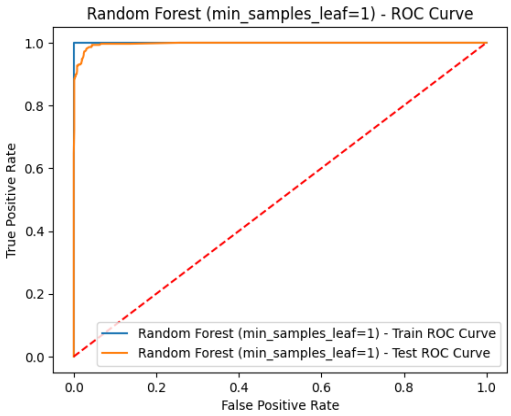


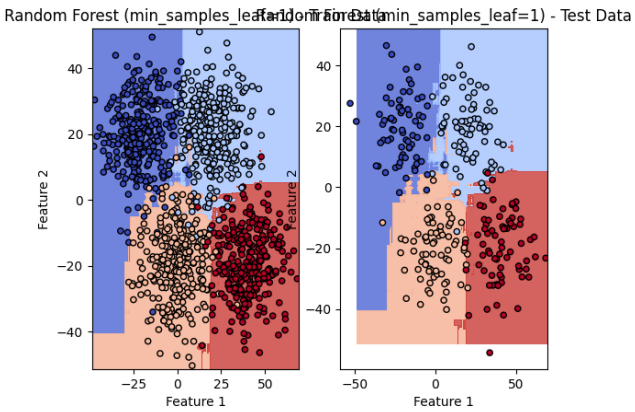


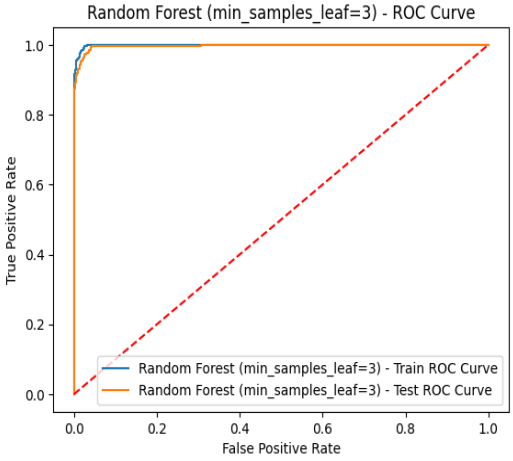


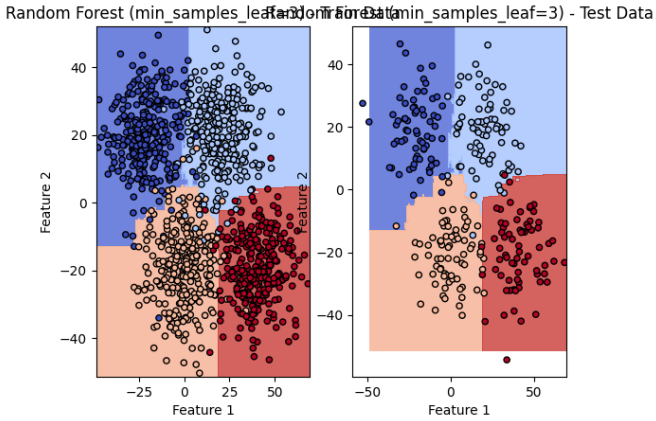


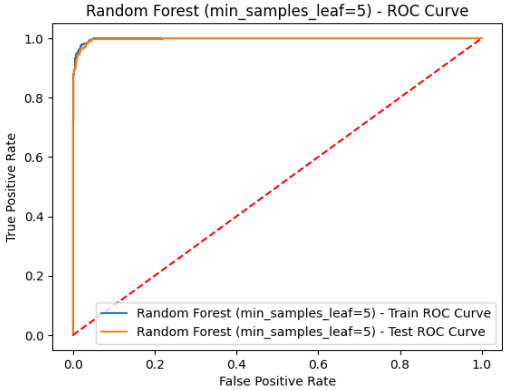


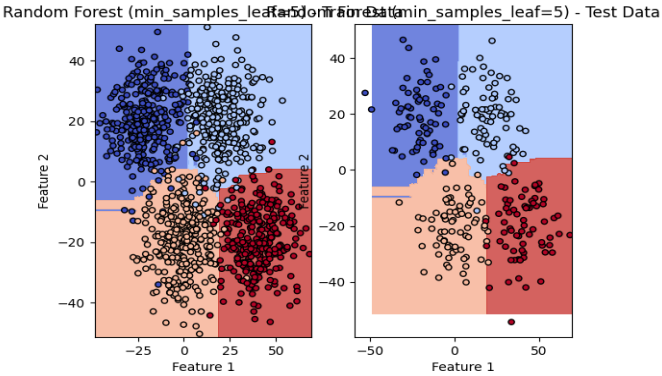

Random Forest (min_samples_leaf=1) - ROC Curve
True Positive Rate
False Positive Rate
Random Forest (min_samples_leaf=1) - Train ROC Curve
Random Forest (min_samples_leaf=1) - Test ROC Curve

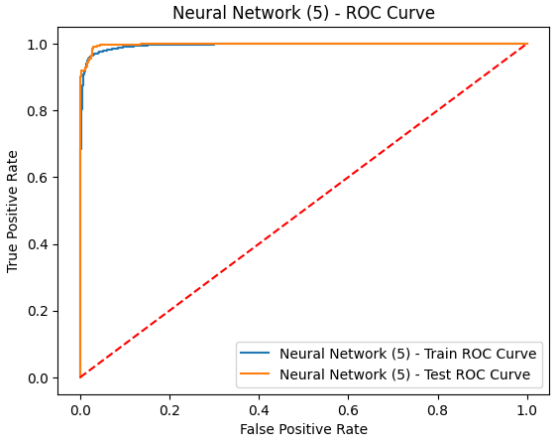

Random Forest (min_samples_leaf=1) - Test Data
Feature 1
Feature 2

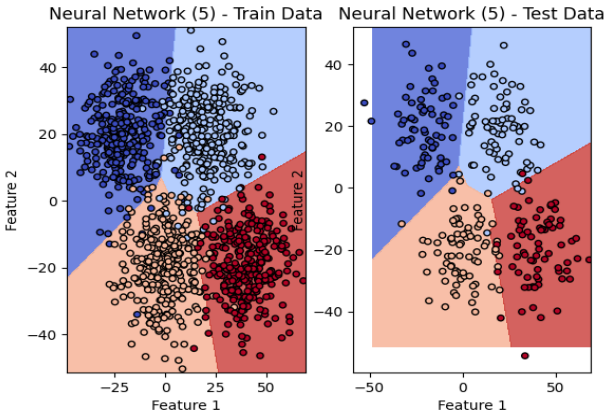

Random Forest (min_samples_leaf=3) - ROC Curve
True Positive Rate
False Positive Rate
Random Forest (min_samples_leaf=3) - Train ROC Curve
Random Forest (min_samples_leaf=3) - Test ROC Curve

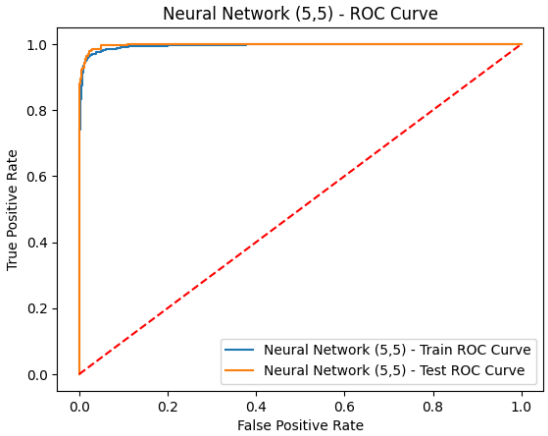

Random Forest (min_samples_leaf=3) - Test Data
Feature 1
Feature 2

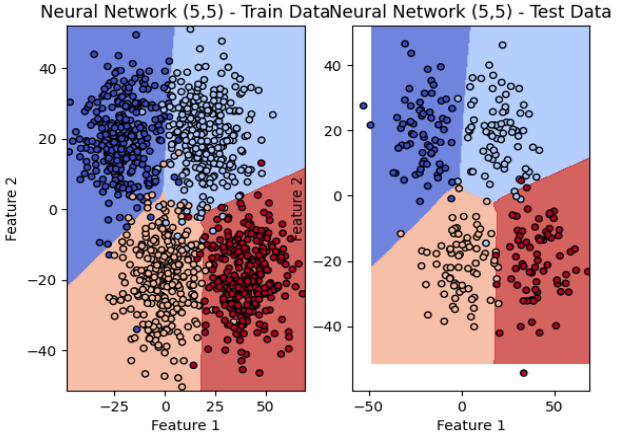

Random Forest (min_samples_leaf=5) - ROC Curve
True Positive Rate
False Positive Rate
Random Forest (min_samples_leaf=5) - Train ROC Curve
Random Forest (min_samples_leaf=5) - Test ROC Curve

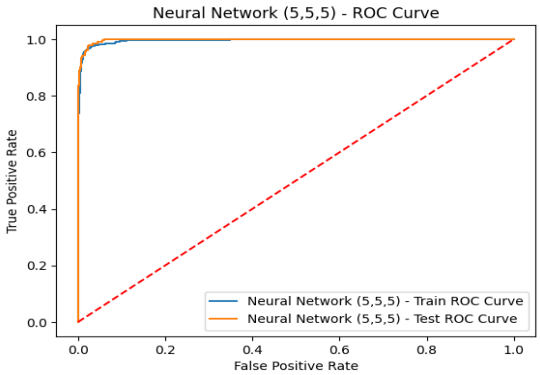

Random Forest (min_samples_leaf=5) - Test Data
Feature 1
Feature 2

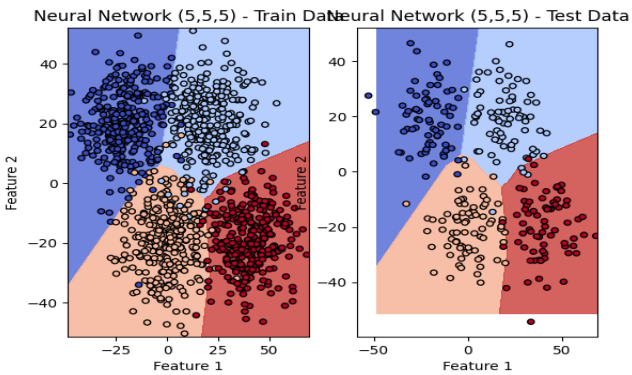

Neural Network (5) - ROC Curve
True Positive Rate
False Positive Rate
Neural Network (5) - Train ROC Curve
Neural Network (5) - Test ROC Curve

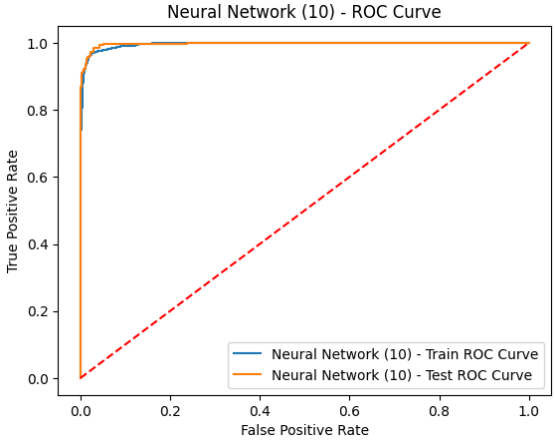

Neural Network (5) - Train Data
Neural Network (5) - Test Data
Feature 1
Feature 2

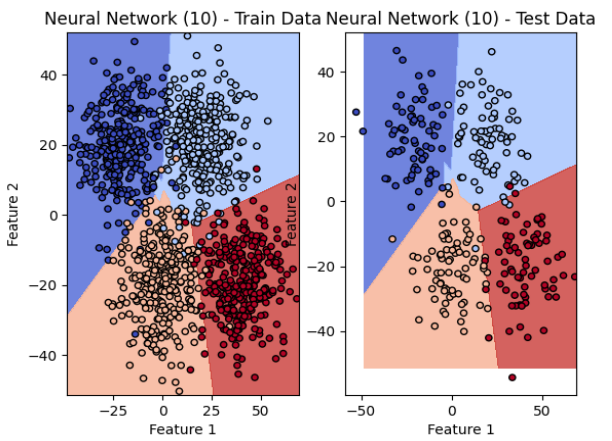

Neural Network (5,5) - ROC Curve
True Positive Rate
False Positive Rate
Neural Network (5,5) - Train ROC Curve
Neural Network (5,5) - Test ROC Curve


Neural Network (5,5) - Train Data
Neural Network (5,5) - Test Data
Feature 1
Feature 2

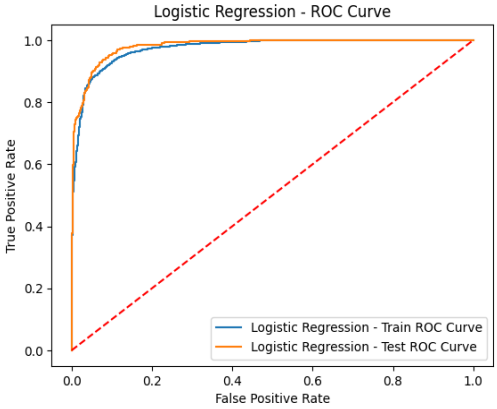

Neural Network (5,5,5) - ROC Curve
True Positive Rate
False Positive Rate
Neural Network (5,5,5) - Train ROC Curve
Neural Network (5,5,5) - Test ROC Curve

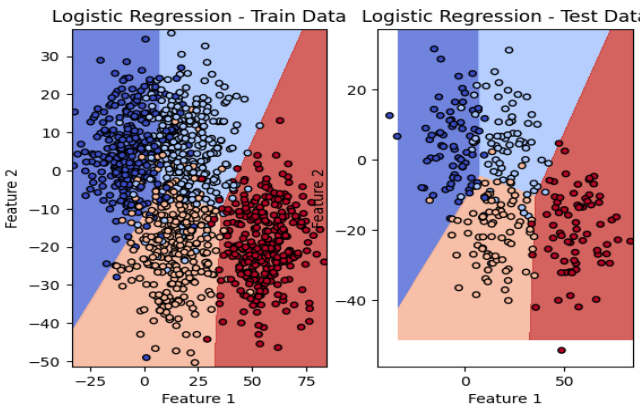

Neural Network (5,5,5) - Train Data
Neural Network (5,5,5) - Test Data
Feature 1
Feature 2

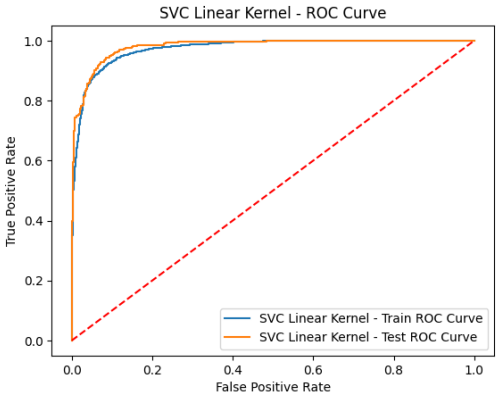

Neural Network (10) - ROC Curve
True Positive Rate
False Positive Rate
Neural Network (10) - Train ROC Curve
Neural Network (10) - Test ROC Curve

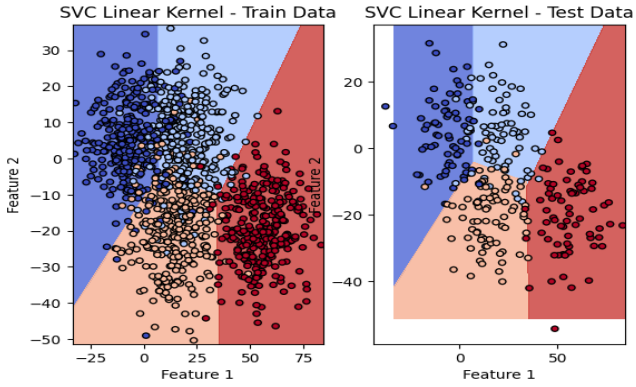

Neural Network (10) - Train Data
Neural Network (10) - Test Data
Feature 1
Feature 2

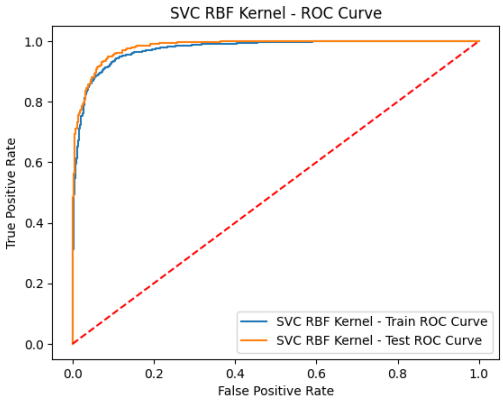
ROC curve and Boundary Layers for Cluster-4-v2.csv

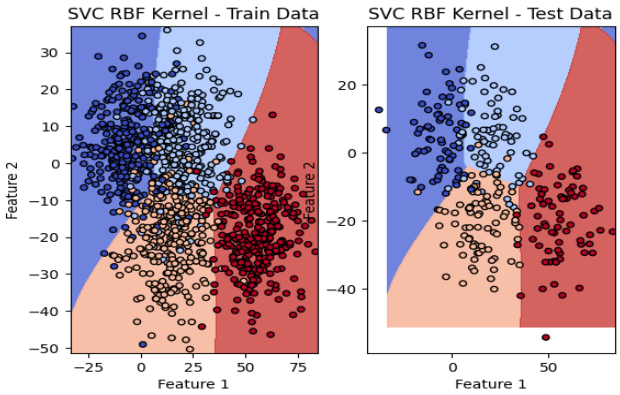


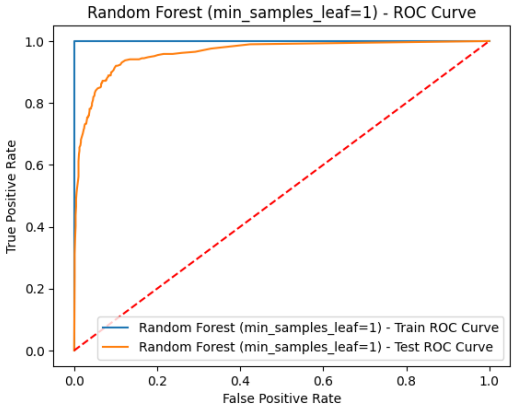


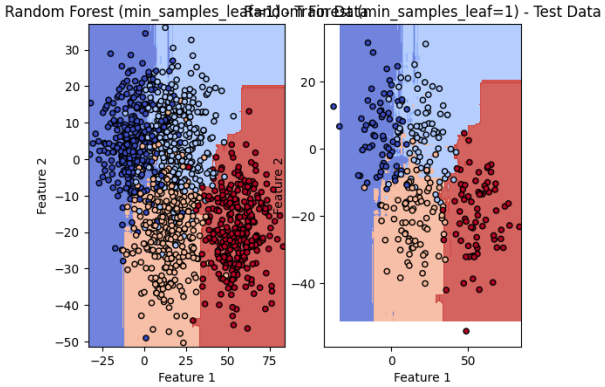


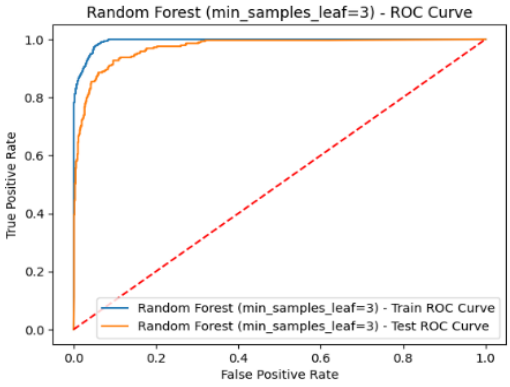


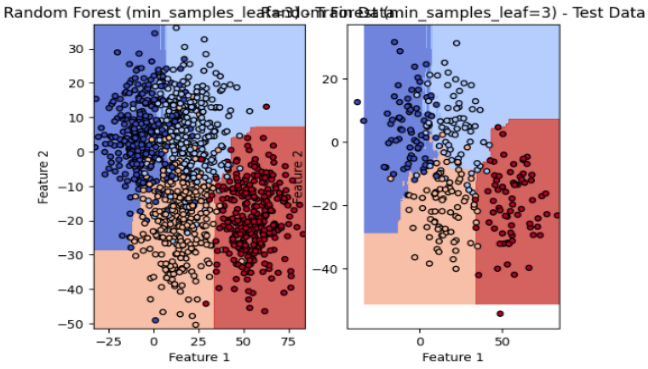

SVC RBF Kernel - ROC Curve
True Positive Rate
False Positive Rate
SVC RBF Kernel - Train ROC Curve
SVC RBF Kernel - Test ROC Curve

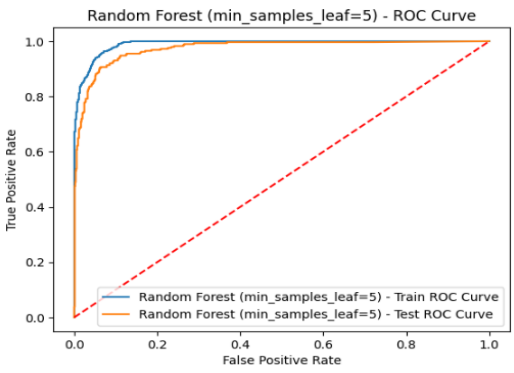

SVC RBF Kernel - Train Data
SVC RBF Kernel - Test Data
Feature 1
Feature 2

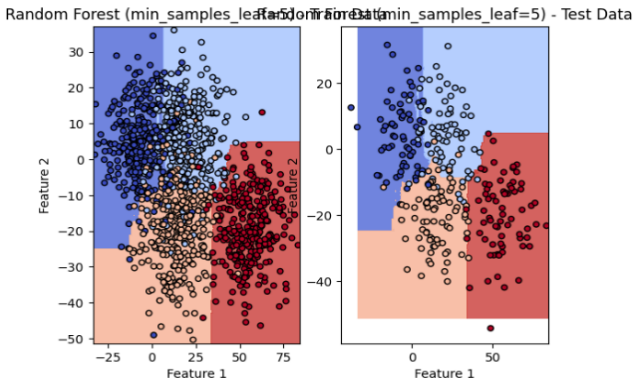

Random Forest (min_samples_leaf=1) - ROC Curve
True Positive Rate
False Positive Rate
Random Forest (min_samples_leaf=1) - Train ROC Curve
Random Forest (min_samples_leaf=1) - Test ROC Curve

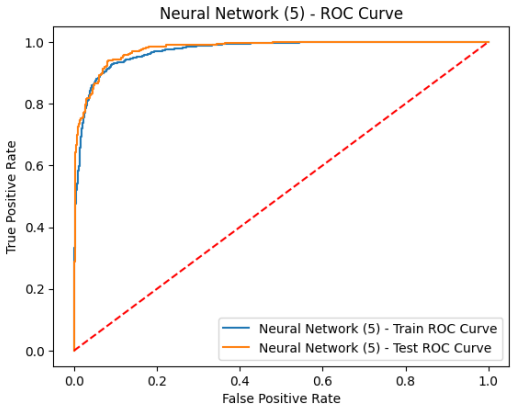

Random Forest (min_samples_leaf=1) - Test Data
Feature 1
Feature 2

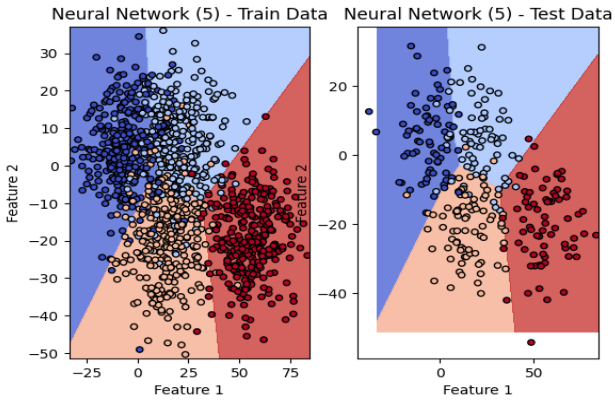

Random Forest (min_samples_leaf=3) - ROC Curve
True Positive Rate
False Positive Rate
Random Forest (min_samples_leaf=3) - Train ROC Curve
Random Forest (min_samples_leaf=3) - Test ROC Curve

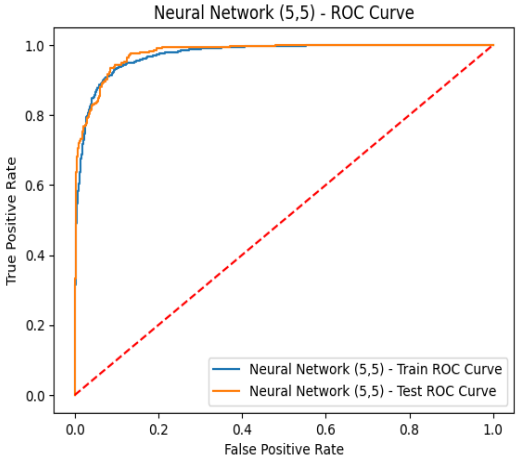

Random Forest (min_samples_leaf=3) - Test Data
Feature 1
Feature 2

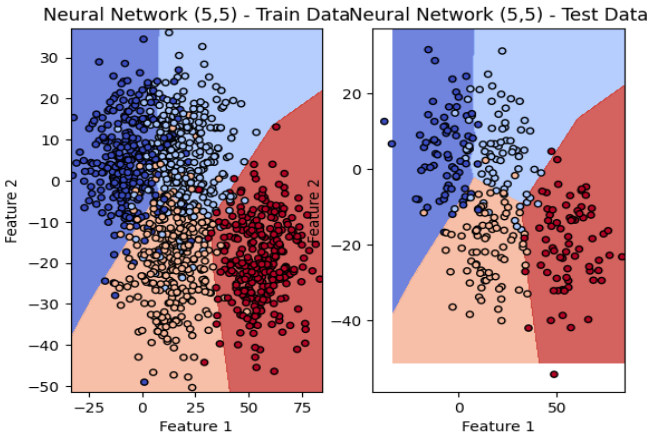

Random Forest (min_samples_leaf=5) - ROC Curve
True Positive Rate
False Positive Rate
Random Forest (min_samples_leaf=5) - Train ROC Curve
Random Forest (min_samples_leaf=5) - Test ROC Curve

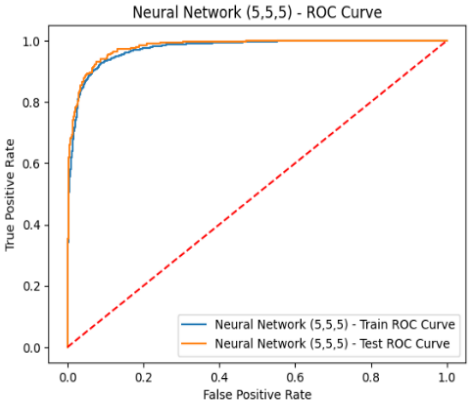

Random Forest (min_samples_leaf=5) - Test Data
Feature 1
Feature 2

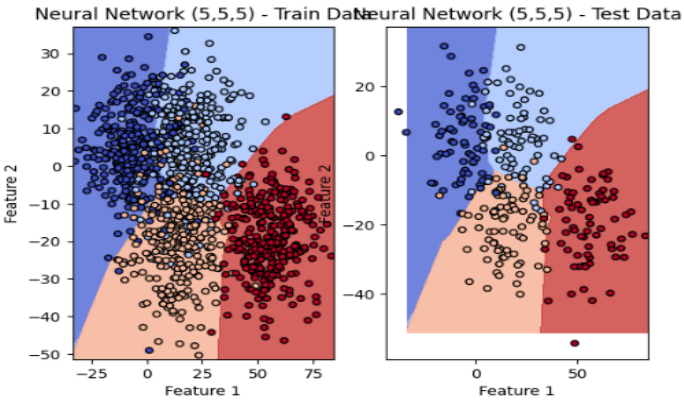

Neural Network (5) - ROC Curve
True Positive Rate
False Positive Rate
Neural Network (5) - Train ROC Curve
Neural Network (5) - Test ROC Curve

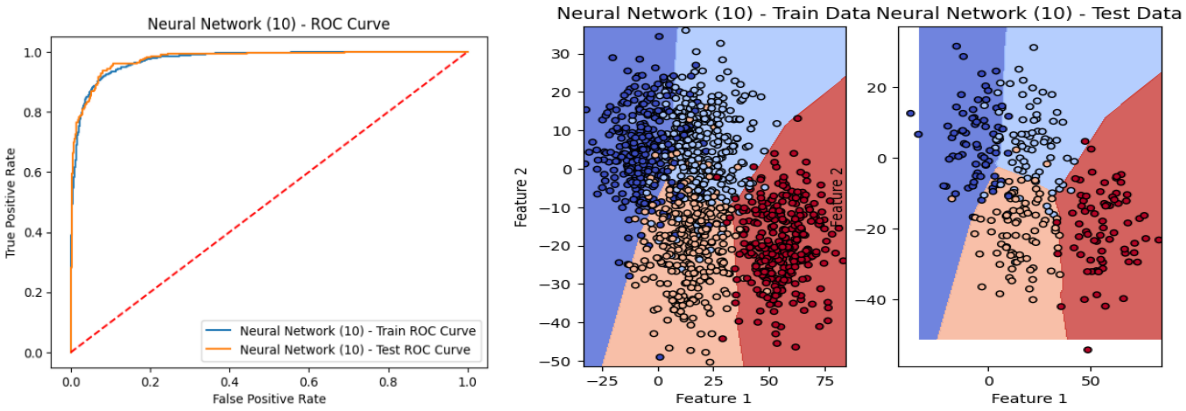

Neural Network (5) - Train Data
Neural Network (5) - Test Data
Feature 1
Feature 2


Neural Network (5,5) - ROC Curve
True Positive Rate
False Positive Rate
Neural Network (5,5) - Train ROC Curve
Neural Network (5,5) - Test ROC Curve


Neural Network (5,5) - Train Data
Neural Network (5,5) - Test Data
Feature 1
Feature 2


Neural Network (5,5,5) - ROC Curve
True Positive Rate
False Positive Rate
Neural Network (5,5,5) - Train ROC Curve
Neural Network (5,5,5) - Test ROC Curve


Neural Network (5,5,5) - Train Data
Neural Network (5,5,5) - Test Data
Feature 1
Feature 2



Learnings from Exercise 4: Classification Algorithms Comparison

This exercise provided valuable insights into the behavior and performance of various classification algorithms on different datasets. Here's a summary of my key takeaways:

Understanding Documentation:

- I effectively leveraged documentation from libraries like ***sklearn*** to grasp the functionalities of different algorithms and their hyperparameters.
- Reading documentation helped me choose appropriate evaluation metrics and visualize results using libraries like ***matplotlib*** and ***seaborn***.

Experimenting with ML Models:

- I successfully implemented different classification models including Logistic Regression, SVM with linear and RBF kernels, Random Forest with varying min_samples_leaf, and Neural Networks with diverse hidden layer configurations.
- Experimenting with different models allowed me to observe their strengths and weaknesses on various datasets.

Hyperparameter Tuning:

- I understood the impact of hyperparameters on model performance by manually adjusting min_samples_leaf in Random Forest and observing changes in accuracy and other metrics.
- While I didn't employ automated techniques like grid search or randomized search, I gained valuable insights into the importance of hyperparameter tuning.

Overfitting Issues:

- I analyzed the results for signs of overfitting, such as significant performance discrepancies between training and testing sets.
- Depending on the observed issues, I might have explored techniques like reducing model complexity (e.g., fewer trees in Random Forest), regularization (e.g., L1/L2 penalty in Logistic Regression), or early stopping.
-

Visulalisation :

I learned how valuable it is to visualise our data and make analysis upon basis of it

Colouring labelling proper metrics and comparing amongst other tuned models made this exercise quite fun and exciting