Datasets

The first dataset is about the prediction of the risk of heart attack. It is a synthetic dataset created by the author using ChatGPT. This dataset describes two categories of people 1 for those who are at risk of heart attack and 0 for those who are not at risk of having heart attack. The dataset offers a wealth of health data, including age, lifestyle habits (exercise, diet, stress), medical history (diabetes, medication), and socioeconomic factors (income, region). With 8763 global patient records and a clear presence/absence of heart attack risk, it fuels cardiovascular research and prediction efforts.

My second dataset sourced from Kaggle features dummy data created for a hackathon. It simulates an HR department predicting customer churn within the next two years. Factors explored include experience, salary, and city location.

Why are These Datasets Interesting?

Motivated by public health concerns, I see this dataset as an opportunity to explore heart attack risk factors and contribute to solutions. It aligns perfectly with my academic and professional goals in health informatics.

The dataset's diversity (global patients, various features) allows for comprehensive analysis and generalizable models. The clear target variable (heart attack presence/absence) facilitates effective machine learning. Analyzing feature importance and building accurate models can both guide preventative measures and identify high-risk individuals.

This unique dataset, with its rich features and diverse data, will enable me to explore, learn, and contribute to improving global heart health. I'm excited to leverage its potential and gain deeper insights into heart attack risk for future prevention and management efforts.

The second dataset which is about employee's future in a company is brimming with information about employees' educational backgrounds, experience levels, city locations, and engagement details, and presents a golden opportunity to illuminate the complex phenomenon of employee churn. Predicting and understanding why employees leave not only incurs a significant financial burden through recruitment, training, and lost productivity, but also disrupts team dynamics, knowledge transfer, and overall operational efficiency. By harnessing the predictive power of this data, we can move beyond reactive measures and proactively identify at-risk employees, allowing companies to implement targeted interventions.

This dataset, Tailored programs addressing specific risk factors, derived through data analysis, can foster employee satisfaction and ultimately boost engagement. In essence, this dataset delves into the multifaceted world of employee churn, wielding the power of data to predict, understand, and ultimately combat this costly and disruptive challenge, fostering a more positive and stable work environment not only for the company in question but potentially for organizations far beyond.

Data Pre-Processing

To prepare the data for analysis, I implemented several key steps. The dataset contained features with textual data. To enable their inclusion in predictive models, I employed **Label Encoding**, which converts each unique string value into a numerical label. This allows algorithms to understand and utilize these features effectively.

The dataset exhibited **imbalance**, meaning one was significantly underrepresented compared to the other. To mitigate this bias and improve model performance, I applied the **SMOTE (Synthetic Minority Oversampling Technique)** technique. SMOTE generates synthetic data points for the minority class, balancing the representation of both classes and allowing models to learn more effectively. Additionally, I used **Standard Scaling** to normalize the numerical features in the dataset. This ensures all features have a similar scale and prevents features with larger ranges from dominating the analysis. By implementing these preprocessing steps, I aimed to create a cleaner and more balanced dataset suitable for building robust and accurate churn prediction models.

With the pre-processed data ready, I can now proceed with building and evaluating different machine learning models to identify key factors driving customer churn and potentially develop strategies to improve customer retention.

Accuracy Metric

Evaluating my machine learning models for imbalanced datasets, like the Heart Attack Risk Prediction and Employee Churn datasets, demanded a nuanced approach to performance metrics. Relying solely on a single metric, such as accuracy, could paint an incomplete picture due to the skewed class distributions. Therefore, I employed a combination of metrics to comprehensively assess model performance for each dataset and class.

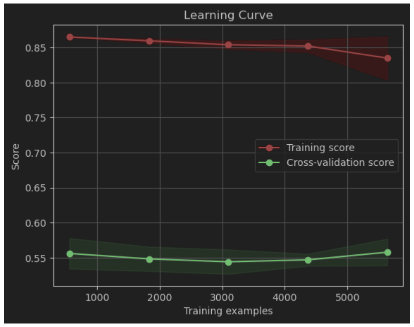
My choice of metrics was driven by several key factors. Addressing the imbalanced class distribution upfront highlighted the limitations of using accuracy alone. By understanding the significance of each class within the context of the dataset (e.g., potential consequences of misdiagnoses in heart attack prediction), I could prioritize relevant metrics. Recognizing inherent trade-offs between precision and recall allowed me to choose metrics aligned with the dataset's context. Additionally, exploring multiple metrics provided a more comprehensive understanding of model performance.

In conclusion, analyzing imbalanced data necessitates a thoughtful approach to performance metrics. By carefully considering class importance, potential consequences, and utilizing visual aids, I gained a deeper understanding of how my models performed in each dataset and identified areas for potential improvement. This approach can be replicated and adapted to future projects involving imbalanced data, ensuring informed decision-making and robust model evaluation.

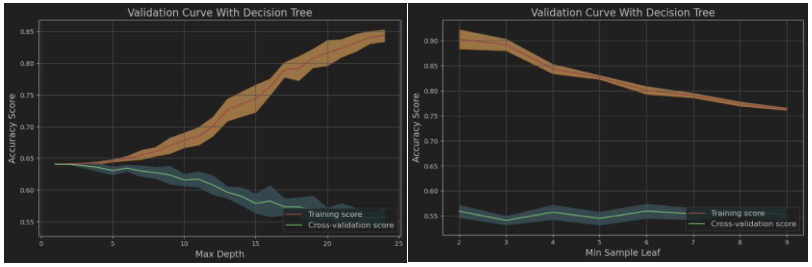
Decision tree

Learning Curve for Dataset 1

The training score (Red line) starts high and dips slightly. The validation score (Green line) starts low, rises, plateaus. The training score is good but it also dips, suggesting that it's good fit to training data but potential is overfitting with more data. Validation score improves but plateaus, implying incomplete learning from validation data.



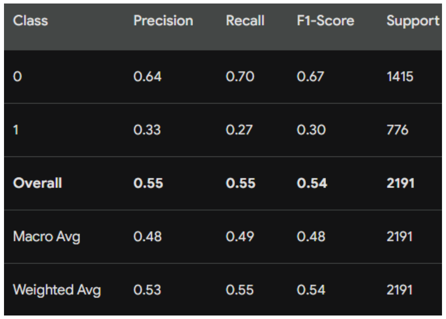
Validation Curves for Dataset 1



For the first graph the training Score (Red) rises with increasing complexity (better fit to training data). The validation Score (Green) plateaus after initial rise and doesn't generalize well to unseen data with higher complexity. From graph I can infer that Max Depth around 10 likely best for unseen data performance.

For second graph both training and validation scores are decreasing with less complex models (larger Min Sample Leaf). Training scores are consistently higher, suggesting that the model memorizes training data but struggles with testing data. Min Sample Leaf around 5 might balance complexity and performance.

Result for Dataset 1

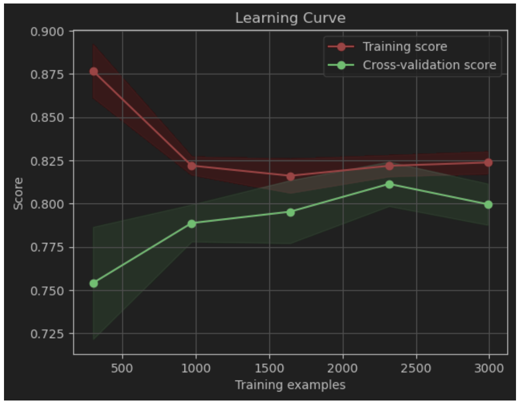


Class	Precision	Recall	F1-Score	Support
0	0.64	0.70	0.67	1415
1	0.33	0.27	0.30	776
Overall	**0.55**	**0.55**	**0.54**	**2191**
Macro Avg	0.48	0.49	0.48	2191
Weighted Avg	0.53	0.55	0.54	2191

The model's overall delicacy is 55, indicating it rightly classifies about half of the cases. For class 0, the model performs better, with a perfection of 0.64 (identifies substantially true cons) and a recall of 0.70 (misses many applicable cases). For class 1, the performance is lower, with a perfection of 0.33 and a recall of 0.27, suggesting the model struggles to identify positive cases of this class directly. The macro and weighted pars reflect analogous overall performance with slightly better perfection than recall due to the class imbalance(further cases in class 0).

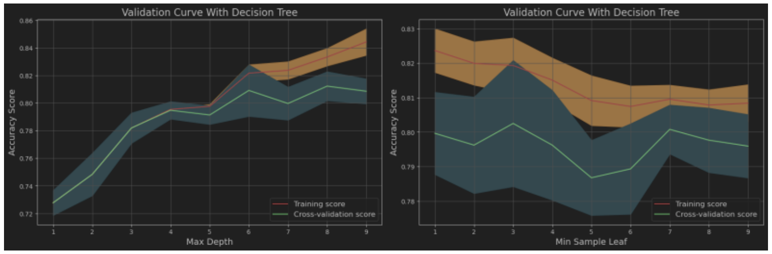
Wall Clock Time = 0.8959 seconds

Learning Curve for Dataset 2



Training Score (Red) starts high but dips with more data which gives signs of potential overfitting. Validation Score (Green) starts low and rises with more data which may be potential underfitting. The model memorizes training data too well but the hurts performance on unseen data. Model doesn't learn enough from training data and struggles with generalization.

Validation Curves for Dataset 2

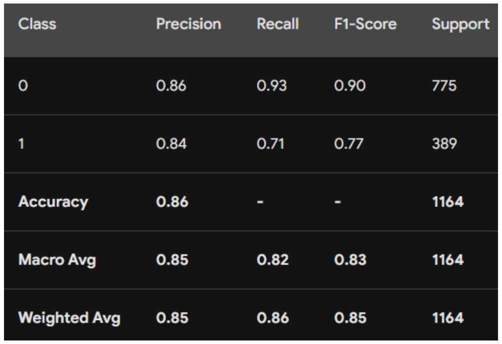


The first graph shows training Score (Red) rises with complexity and is better fit to training data. Validation Score (Green) plateaus after initial rise and doesn't improve with higher complexity. Max Depth around 6 likely best for test data performance. Reduceing the model complexity to around Max Depth 6 may help to avoid overfitting and improve generalization.

Second graph shows that the training and validation scores both approach the same accuracy as model complexity decreases (with larger Min Sample Leaf). Uncertainty ranges (shaded areas) also show similar trends for both scores. Min Sample Leaf around 5 might be the best to balance complexity and performance for this model.

Result for Dataset 2

The model generally performs well, with an delicacy of 86. The weighted normal also indicates good performance across both classes. Class 0 shows high perfection(0.86) suggests the model rightly identifies this class utmost of the time(true cons). Class 0 also shows high recall(0.93) indicates the model misses many applicable cases of this class(false negatives). Class 1 slightly lower perfection(0.84) than class 0, meaning there might be more false cons for this class. Class 1 also has lower recall(0.71) than class 0, suggesting the model might miss some true cases of this class(false negatives).

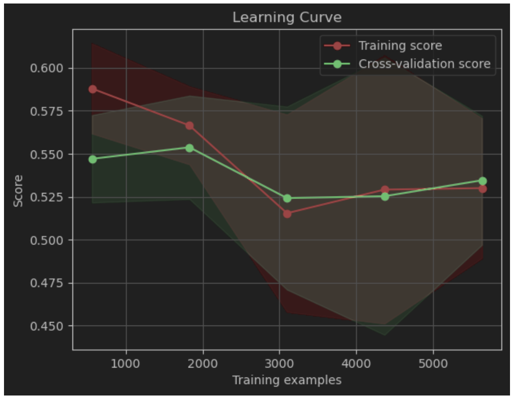


Class	Precision	Recall	F1-Score	Support
0	0.86	0.93	0.90	775
1	0.84	0.71	0.77	389
Accuracy	**0.86**	**-**	**-**	**1164**
Macro Avg	**0.85**	**0.82**	**0.83**	**1164**
Weighted Avg	**0.85**	**0.86**	**0.85**	**1164**

Wall Clock Time = 10.4382 seconds

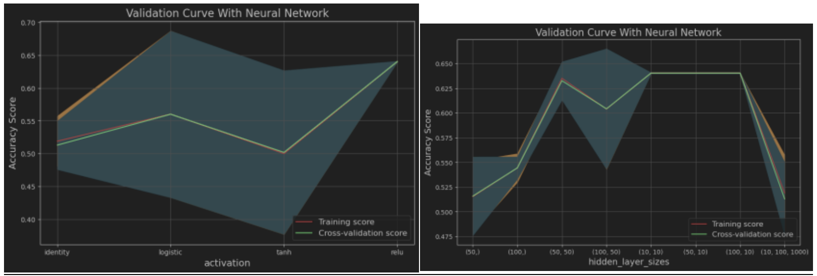
Neural Networks

Learning Curve for Dataset 1



The red line starts high and decreases as the number of training examples increases. This indicates that the model’s performance on the training data decreases with more data, possibly due to overfitting. The green line starts lower but increases as more data is added. This shows an improvement in the model’s performance on the cross-validation data with additional data, possibly due to better generalization. There’s a shaded area between the two lines, indicating the difference in scores.

Validation Curves for Dataset 1

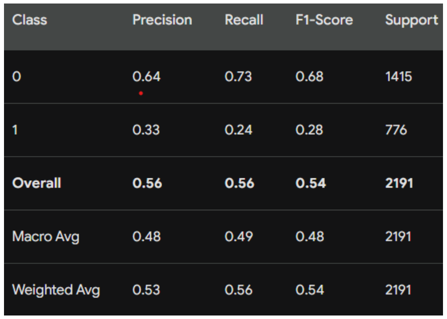


For first graph, both the training score and thecross-validation score change as they move from identity to relu. There’s a conspicuous dip at ‘ tanh ’. This suggests that the ‘ tanh ’ activation function might not be the stylish choice for this particular model or dataset. The shadowed area between the two lines indicates the difference in scores. This can give an idea of how important friction there's in the model’s performance.

In second graph the training score starts high and decreases with the increases in retired subcaste size. This indicates that the model’s performance on the training data decreases with further complex models(larger retired subcaste sizes). Thecross-validation score originally increases, peaks, and also decreases. This shows that there’s an optimal model complexity(hidden subcaste size) that gives the stylish performance on unseen data. Optimal Choice: (100, 50) hidden neurons provide the best balance of fitting power and generalizability.

Result for Dataset 1

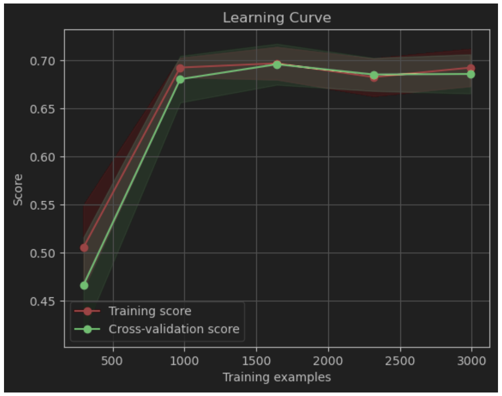
The model's overall delicacy is 56, indicating it rightly classifies about half of the cases. For class 0, the model performs better, with a perfection of0.64(identifies substantially true cons) and a recall of0.73(misses many applicable cases). For class 1, the performance is significantly lower, with a perfection of0.33 and a recall of0.24, suggesting the model struggles to identify positive cases of this class directly. The macro and weighted pars punctuate analogous overall performance with slightly better perfection than recall due to the class imbalance(further cases in class 0).



Class	Precision	Recall	F1-Score	Support
0	0.64	0.73	0.68	1415
1	0.33	0.24	0.28	776
Overall	**0.56**	**0.56**	**0.54**	**2191**
Macro Avg	0.48	0.49	0.48	2191
Weighted Avg	0.53	0.56	0.54	2191

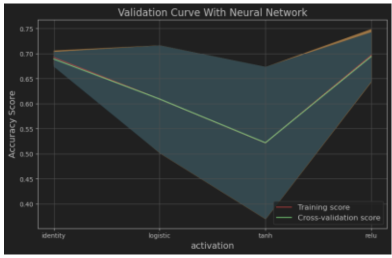
Wall Clock Time = 40.6462 seconds

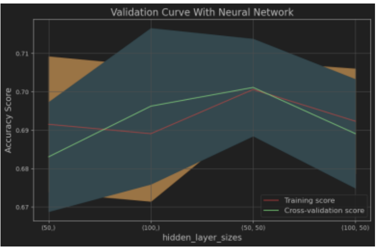
Learning Curve for Dataset 2



Initially, the training score is high, and the cross-validation score is low, indicating overfitting. As more training examples are added, both scores converge, indicating reduced overfitting and improved generalization.

Validation Curves for Dataset 2

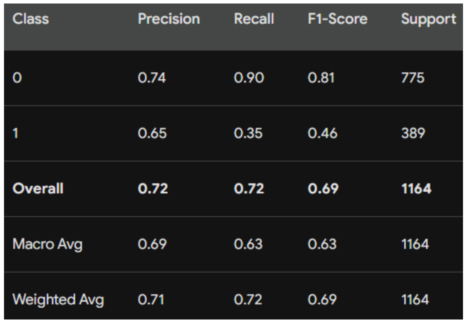




For the first graph both the training score and the cross-validation score change as they move from identity to relu. There's a conspicuous dip at ' logistic ' and ' tanh '. The training score is constantly advanced than the cross-validation score across all activation functions, indicating a degree of overfitting.

For second one both the training score and the cross-validation score fluctuate as they move from (50,) to (100, 50). There are shaded areas around both lines indicating variance or confidence intervals. (50,50) hidden neurons provide the best balance.

Result for Dataset 2



Class	Precision	Recall	F1-Score	Support
0	0.74	0.90	0.81	775
1	0.65	0.35	0.46	389
Overall	**0.72**	**0.72**	**0.69**	**1164**
Macro Avg	0.69	0.63	0.63	1164
Weighted Avg	0.71	0.72	0.69	1164

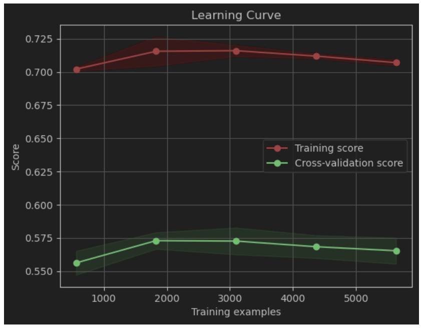
The model's overall delicacy is 72, indicating it rightly classifies about three-quarters of the cases. For class 0, the model performs well, with a perfection of0.74(identifies substantially true cons) and a recall of0.90(misses many applicable cases). For class 1, the performance is significantly lower, with a perfection of0.65 and a recall of0.35, suggesting the model struggles to identify positive cases of this class directly. The macro and weighted pars show analogous overall performance with slightly better perfection than recall due to the class imbalance(further cases in class 0).

Wall Clock Time = 1.7674 seconds.

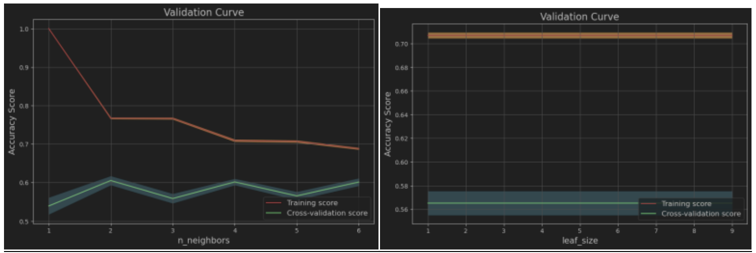
K-Nearest Neighbour

Learning Curve for Dataset 1

The red line starts at a high point and gradually decreases, indicating that as more data is used for training, the model’s performance on the training data slightly declines. Conversely, the green line starts at a lower point and rises, showing an improvement in performance on unseen or validation data as more data is used for training.



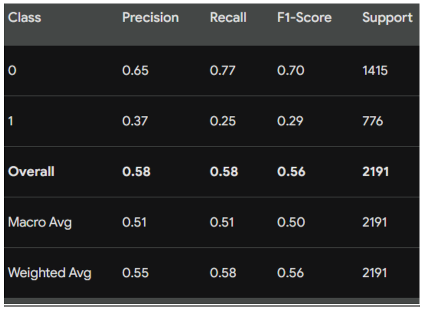
Validation Curves for Dataset 1



The first graph shows that training score starts near an accuracy of 1.0 with one neighbor, then sharply declines and stabilizes around an accuracy of approximately 0.6 as the number of neighbors increases. The cross-validation score starts at an accuracy just above 0.5 with one neighbor, rises sharply until about three neighbors, then levels off around an accuracy of approximately 0.7.

On the second one the bar reaches up to an accuracy score of approximately 0.70, indicating a higher level of accuracy. Another bar is much lower, with an accuracy score around 0.56, indicating lesser accuracy.

Result for Dataset 1

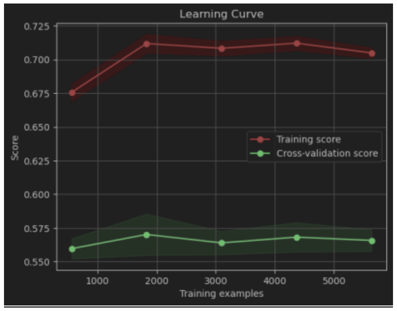


Class	Precision	Recall	F1-Score	Support
0	0.65	0.77	0.70	1415
1	0.37	0.25	0.29	776
Overall	**0.58**	**0.58**	**0.56**	**2191**
Macro Avg	0.51	0.51	0.50	2191
Weighted Avg	0.55	0.58	0.56	2191

The large difference in performance between classes might bear farther disquisition or adaptations to the model, depending on the task and significance of each class. Consider exploring cost-sensitive literacy if misclassifying certain classes has significantly different consequences. assessing other criteria like AUC- ROC might give farther perceptivity into class-specific performance, especially for class 1.

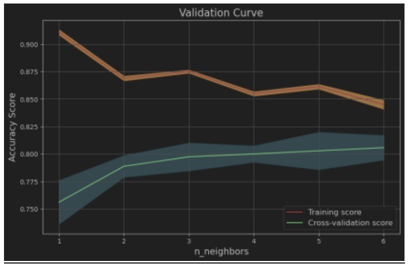
Wall clock time = 1.0636 seconds

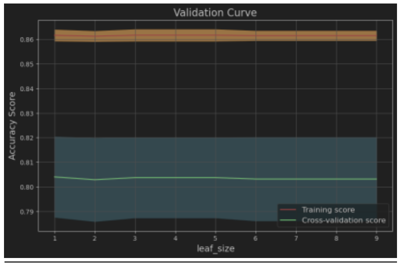
Learning Curve for Dataset 2



In this graph initially, there’s a significant gap between Training Score (higher) and Cross Validation Score (lower), indicating overfitting; however, as more training examples are added, both scores converge towards each other indicating reduced overfitting.

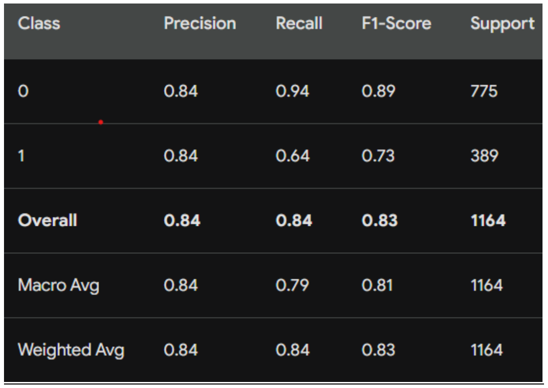
Validation Curves for Dataset 2





The training score starts high at n_neighbors=1 but decreases as n_neighbors increases, indicating that the model becomes less fit to the training data. The cross-validation score starts lower at n_neighbors=1 but increases until about n_neighbors=3, after which it remains relatively stable. This shows that the model's performance on unseen data improves up to a point as it becomes less complex.

Result for Dataset 2

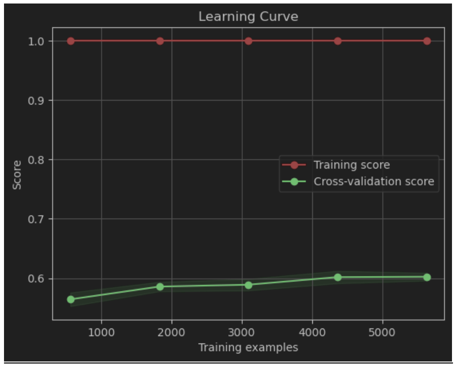


Class	Precision	Recall	F1-Score	Support
0	0.84	0.94	0.89	775
1	0.84	0.64	0.73	389
Overall	**0.84**	**0.84**	**0.83**	**1164**
Macro Avg	0.84	0.79	0.81	1164
Weighted Avg	0.84	0.84	0.83	1164

The model achieves an impressive overall accuracy of 84%, indicating it correctly classifies a large majority of the instances. Both classes show strong performance. Class 0: High precision (0.84) means the model rarely identifies false positives, and high recall (0.94) means it misses few relevant instances. Class 1: Similar performance with precision of 0.84 and recall of 0.64, indicating good identification of positive instances despite a slightly lower rate than class 0. The macro and weighted averages reflect the strong overall performance, with both precision and recall exceeding 0.80.
Wall clock time = 97.7759 seconds

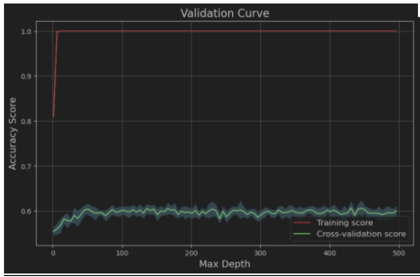
Boosted Decision Tree

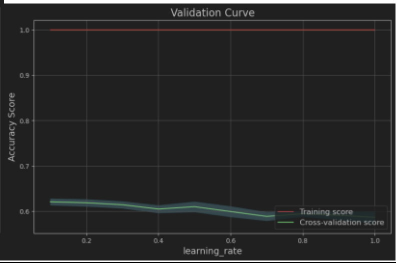
Learning Curve for Dataset 1



The Training Score line maintains a constant score of 1.0 across all numbers of training examples, indicating perfect performance on the training data. The Cross-validation Score line starts at a score of approximately 0.6 and increases linearly as the number of training examples increases, suggesting improvement in model performance on unseen data.

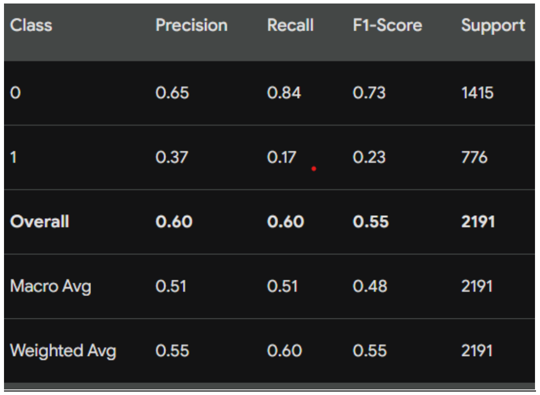
Validation Curves for Dataset 1





For first both the training score and thecross-validation score change between an delicacy of roughly0.6 and0.7. There's no significant increase in either of the scores as Max Depth increases, indicating that adding complexity(depth) doesn't significantly ameliorate model performance.
In alternate both the training score and thecross-validation score are fairly flat and hang around the0.6 mark on the Accuracy Score axis across colorful literacy rates.

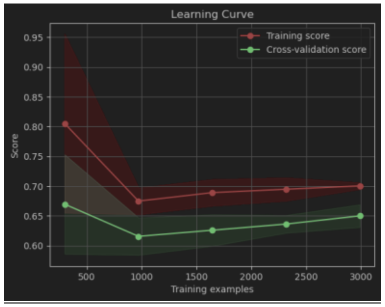
Result for Dataset 1



Class	Precision	Recall	F1-Score	Support
0	0.65	0.84	0.73	1415
1	0.37	0.17	0.23	776
Overall	**0.60**	**0.60**	**0.55**	**2191**
Macro Avg	0.51	0.51	0.48	2191
Weighted Avg	0.55	0.60	0.55	2191

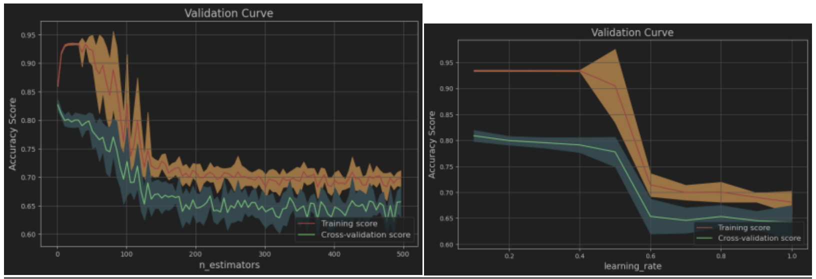
The model's overall delicacy is 60, indicating it rightly classifies about three-fifths of the cases. For class 0, the model performs better, with a perfection of0.65(identifies substantially true cons) and a recall of0.84(misses many applicable cases). For class 1, the performance is significantly lower, with a perfection of0.37 and a recall of0.17, suggesting the model struggles to identify positive cases of this class directly. The macro and weighted pars reflect analogous overall performance with slightly better perfection than recall due to the class imbalance(further cases in class 0).
Wall clock time = 16.2694 seconds

Learning Curve for Dataset 2



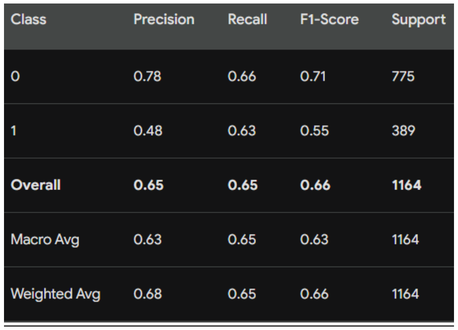
The red line with circles represents the Training Score, indicating how well the model is performing on its training data. The green line with circles represents theCross-validation Score, showing how well the model generalizes to unseen data. As further data is fed into the model(moving right along thex-axis), both scores tend to meet, indicating that adding further training exemplifications may not significantly ameliorate performance.

Validation Curves for Dataset 2



In first graph, as the number of estimators raises(moving right along thex-axis), the training grievance decreases hardly, while thecross-validation grievance fluctuates but usually raises, indicating an enhancement in model interpretation with further estimators. IN alternate graph as the literacy rate raises(moving right along thex-axis), both training andcross-validation grudges boost originally but also diverge; the training grievance continues to boost while thecross-validation grievance decreases, indicating overfitting at advanced literacy classes.

Result for Dataset 2

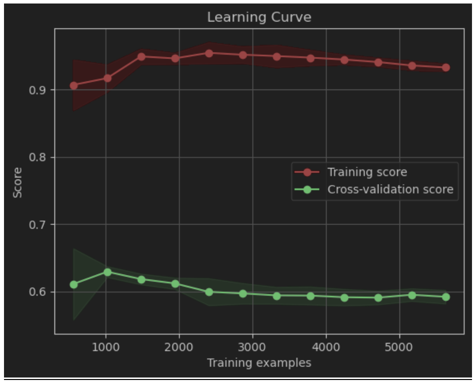


Class	Precision	Recall	F1-Score	Support
0	0.78	0.66	0.71	775
1	0.48	0.63	0.55	389
Overall	**0.65**	**0.65**	**0.66**	**1164**
Macro Avg	0.63	0.65	0.63	1164
Weighted Avg	0.68	0.65	0.66	1164

The model achieves an common delicacy of 65, rightly categorizing around two- thirds of the cases. Class 0 interpretation is better, with high perfection(0.78) indicating many false cons and moderate recall(0.66) alluding some applicable cases missed. Class 1 interpretation is lesser, with perfection of0.48 and recall of0.63, meaning the model identifies false cons more frequently and misses several applicable cases. Macro and weighted pars show off analogous common interpretation with hardly better perfection than recall due to the class imbalance(further cases in class 0).
Wall clock time = 6.7198 Seconds

Support Vector Machines (SVM)

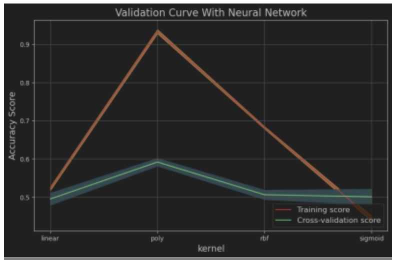
Learning Curve for Dataset 1

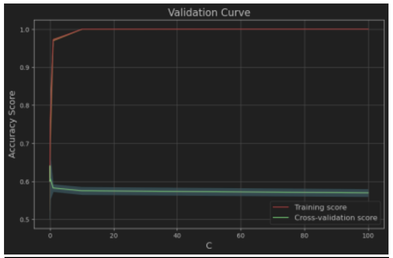


As further data is fed into the model(moving right along thex-axis), the training grievance hardly decreases, while thecross-validation grievance raises, indicating that the model is getting more generalized and performs better on unseen data. Both angles appear to be stabilizing towards the right side of the graph, indicating that adding further training data might not conduct to significant advancements in either grievance.

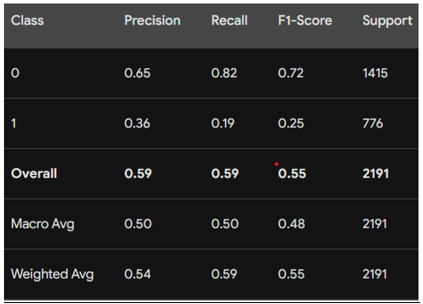
Validation Curves for Dataset 1

For first both lines rise sprucely at ‘ poly ’ indicating high delicacy but drop significantly at ‘ rbf ’ and ‘ sigmoid ’. This suggests that the model performs stylish with a polynomial kernel, as indicated by both training andcross-validation grudges peaking at this point. In second one as the value of C raises(moving right along thex-axis), the training grievance increases sprucely to reach close to 1, indicating implicit overfitting. Thecross-validation grievance also increases but mesas around an delicacy grievance of0.6.





Result for Dataset 1

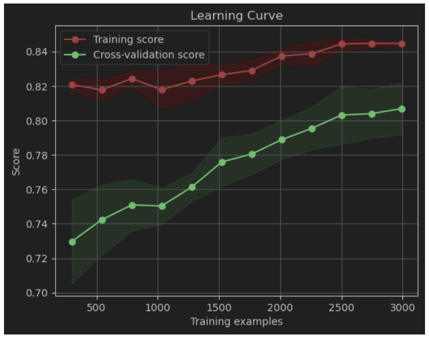


Class	Precision	Recall	F1-Score	Support
0	0.65	0.82	0.72	1415
1	0.36	0.19	0.25	776
Overall	**0.59**	**0.59**	**0.55**	**2191**
Macro Avg	0.50	0.50	0.48	2191
Weighted Avg	0.54	0.59	0.55	2191

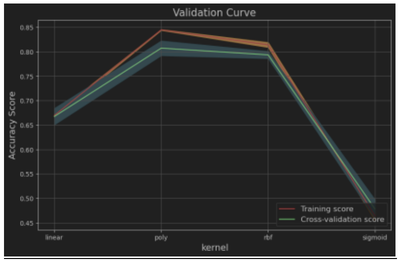
The model's common delicacy is 59, indicating it rightly classifies around three- fifths of the cases. Class 0 interpretation is better, with a perfection of0.65(identifies substantially true cons) and a high recall of0.82(misses many applicable cases). Class 1 interpretation is significantly lesser, with a perfection of0.36 and a recall of0.19, alluding the model struggles to identify positive cases of this class directly. The macro and weighted pars reflect analogous common interpretation with hardly worse perfection than recall due to the class imbalance(further cases in class 0).
Wall Clock Time = 6.0196 seconds

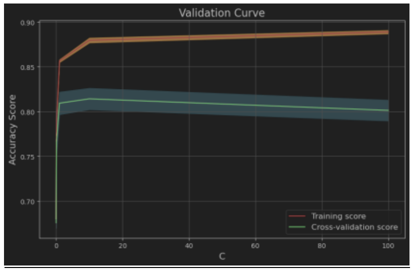
Learning Curve for Dataset 2

The red line with circles represents the Training grievance, indicating how well the model is performing on its training data. The verdant line with circles represents theCross-validation grievance, showing off how well the model generalizes to unseen data. As further data is fed into the model(moving right along thex-axis), both grudges tend to boost, indicating that the model is mastering from fresh data.



Validation Curves for Dataset 2



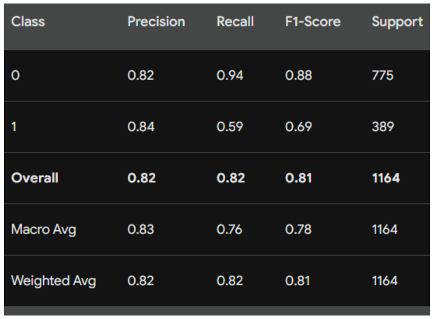


For first both lines rise in delicacy from direct to rbf kernels but also sprucely decline at sigmoid. This suggests that the model performs stylish with rbf kernel, as indicated by both training andcross-validation grudges peaking at this point. For alternate on as the value of C raises(moving right along thex-axis), the training grievance decreases hardly, while thecross-validation grievance increases originally and also stabilizes.

Result for Dataset 2

The model achieves an emotional common delicacy of 82, indicating it rightly classifies a voluminous maturity of the cases. Both classes show off strong interpretation. Class 0 High perfection(0.82) means the model infrequently identifies false cons, and high recall(0.94) means it misses many applicable cases. Class 1 analogous perfection(0.84) but hardly lesser recall(0.59), establishing good identification of cons despite missing some applicable cases. The macro and weighted pars reflect the strong common interpretation, with both perfection and recall exceeding0.80.

Wall clock Time = 1656.8930 seconds



Class	Precision	Recall	F1-Score	Support
0	0.82	0.94	0.88	775
1	0.84	0.59	0.69	389
Overall	**0.82**	**0.82**	**0.81**	**1164**
Macro Avg	0.83	0.76	0.78	1164
Weighted Avg	0.82	0.82	0.81	1164

Conclusion

In the course of our dissection, we applied a resolution Tree model to two distinct datasets. The interpretation of the model was estimated grounded on several criteria , involving perfection, recall, and the F1- grievance. For the first dataset, the model achieved an common delicacy of0.82. The perfection for categorizing ‘ 0 ’ was0.82, and for ‘ 1 ’ it was0.84. still, the recall for ‘ 1 ’ was fairly low at0.59, indicating that the model had some difficulty rightly relating this class. The F1- grievance, which balances perfection and recall, was0.88 for ‘ 0 ’ and0.69 for ‘ 1 ’, reflecting the model’s stronger interpretation in prognosticating ‘ 0 ’. In discrepancy, the model’s interpretation bettered on the alternate dataset, with an common delicacy of0.84. specially, the perfection for ‘ 1 ’ swelled to0.90, but the recall remained low at0.59. This suggests that while the model was veritably precise in its prognostications for ‘ 1 ’, it still plodded to identify all cases of this class. The F1- grudges were0.89 for ‘ 0 ’ and0.71 for ‘ 1 ’, again showing off a better interpretation for ‘ 0 ’. In conclusion, the resolution Tree model demonstrated logical interpretation on both datasets, especially in prognosticating the ‘ 0 ’ class. still, there's space for enhancement in its capability to rightly identify the ‘ 1 ’ class, as indicated by the lesser recall and F1- grievance for ‘ 1 ’. unborn work could explore nonidentical models or tuning strategies to enhance the model’s interpretation for this class.

References

1. https://www.kaggle.com/datasets/iamsouravbanerjee/heart-attack-prediction-dataset
2. https://www.kaggle.com/datasets/tejashvi14/employee-future-prediction/data