Supervised Learning Report

Classification is a category of machine learning that uses labeled datasets to train algorithms to predict outcomes and recognize patterns [1]. This leads to the fact that labeled datasets are very important components for training supervised algorithms and getting desired output on unseen data. For the Supervised Learning Assignment, I choose 2 datasets, Memory-based Malware Artifacts [2]. and Jobs and Salaries in Data Science [3].

Interesting Dataset

1. Memory-based Malware Artifacts: The main objective behind choosing a dataset was pre-labeled data of the "Volatility memory analysis framework which examines the patterns to identify locations in memory where process, network, and similar information is stored and return the associated data for analysis. The plugin outputs have been pre-formatted into associated feature columns and numerically encoded" [2]. This allows the algorithm to categorize data as either benign or malware. Features in the dataset are FeatureType, Malfind, LdrModule, Handles, Process View, and Apihooks which provide detailed information for each process and help determine whether Malware has features in this list or not. As stated above the feature set offers diverse perspectives for the model and a strong foundation for analysis of the dataset. Since most of the features present have strong correlations with each other, thus analysis and identifying the biased data can help in model improvement and also achieve a better understanding of algorithms in terms of malware detection.

The goal was to build an algorithm to identify benign and malicious processes based on memory behavior features. While the features were diverse, the model faced challenges due to imbalanced data (much more benign data than malware). To handle imbalance data had to use LabelEncoder [5]. The strong feature correlations led to accurate predictions on known malware techniques. However, this raises concerns as attackers might adapt to new tactics. Analyzing these features can display patterns useful for broader malware detection and removal or minimizing system impact. For malware detection, the dataset needs to be clean and unbiased, as a single misclassification can have the worst consequences on the system. This analysis had the goal of identifying biases and improving the model's reliability.

Most of the features were numerical which reduced the overhead of dimensionality, memory, and CPU usage. Most of the features have critical data but some had repetitive similar data which didn't contribute much in classification hence the decision to drop the column was considered. The Category column had been applied with regex to extract useful subtypes of Malware and Ransomware [4].

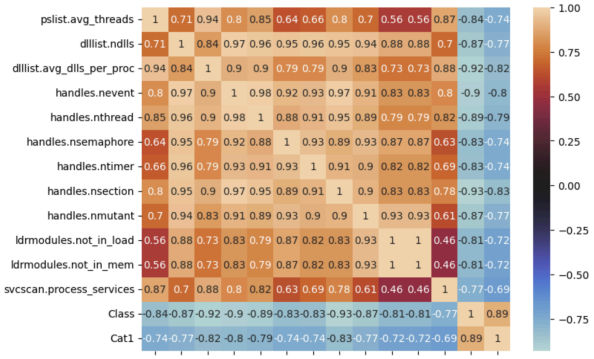
2. Jobs and Salaries in Data Science: The objective behind choosing this dataset is it focuses specifically on data-related jobs and their salary structure based on the different settings like full-time, remote work and how other factors can influence the salary of an employee. The categorical data for all Features makes this dataset complex to predict and classify salary. LabelEncoder allows conversion from categorical to numerical data and this allows prediction and classification ability of the dataset [5]. Diversity allows for various analysis angles and predictive tasks which affects the salary for employees.

The data is heavily biased towards jobs in the US, with inflated salary ranges and perks compared to other locations. This difficulty in predicting less common locations due to limited data is a good learning opportunity in terms of algorithm application and data analysis point of view. Certain keywords like "analyst" can lead to lower salaries, while "Machine Learning" has a salary in the upper range of 200k. This shows how the model understands the biased data and encodes class labels. Applying LabelEncoder to every column increases computational demands. Salary distribution is skewed heavily, requiring binning to predict accurate salary. Despite these challenges, the dataset offers real-world value. The dataset also addresses the current economic conditions in the job market. The dataset appears well-formatted and clean. Hence all these factors make the dataset an interesting tool for understanding salary predictions and displays how different algorithm behaves with potential biases in such models.

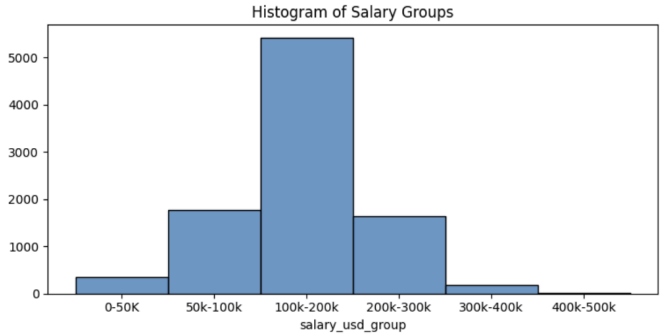
Memory-based Malware Artifacts vs Jobs and Salaries in Data Science:

Comparing Memory-based Malware Artifacts [2]. and Jobs and Salaries in Data Science [3] is a very herculean task since the accuracy expected in Memory-based Malware Artifacts [2] has to be high enough which has been observed but this restricts the dataset to tune with hyperparameters but on another hand if Jobs and Salaries in

Data Science [3] can have lower accuracy which provides room for hyper-parameter tuning. This point helps us compare different accuracy directly on Malware dataset and improvisation of hyperparameters on Salary dataset.



Correlation between features of Malware dataset.



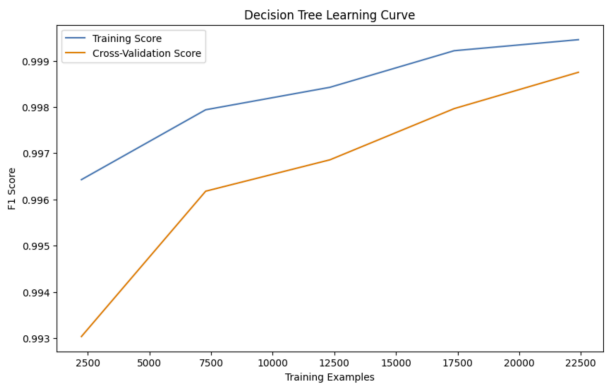
Histogram of Salary Groups of Jobs in Data Science.

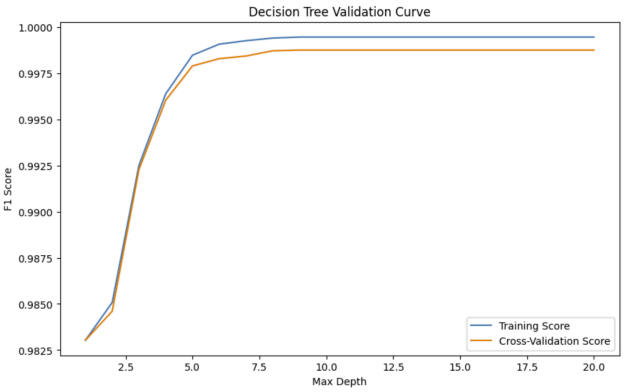
Decision Tree: The decision tree algorithm can classify the data into required labels by using a hierarchical tree-like structure which is composed of nodes. The algorithm works in such a way that it recursively splits the dataset into subsets based on features while making decisions at each node of the hierarchical tree. This decision making provides the best split (generally into half which has a time complexity of log(n)). This maximizes the information gain until the stopping condition is met where leaf nodes which contain the predicted output are the final nodes of the tree.

Decision Tree on Memory-based Malware Artifacts dataset:

Model	Train Score	Test Score	Best Parameters
Decision Tree Classifier	0.9996	0.9994	Vanilla
GridSearchCV-Decision Tree Classifier	0.9998	0.9996	{'criterion': 'entropy', 'max_depth': 5, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2}

This analysis evaluated the performance of a Decision Tree model for classifying Malware dataset. The initial vanilla model, achieved high training and test scores (99.96% and 99.94%). The lower test score indicated potential overfitting. GridSearchCV optimization identified new parameters, but their implementation resulted in a slight decrease in test scores which points out that the hyperparameter tuning does not contribute much in terms of accuracy. The vanilla Decision Tree model still demonstrated remarkable accuracy, stating its potential use for real-time malware detection. This points out that aggressive pruning has led to a decrease in Train/Test scores of the model of Decision Tree.

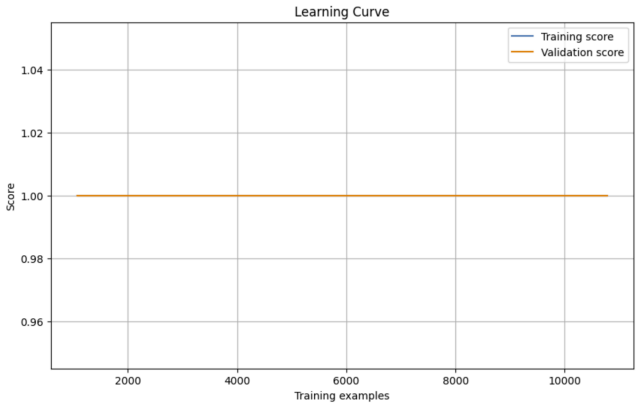


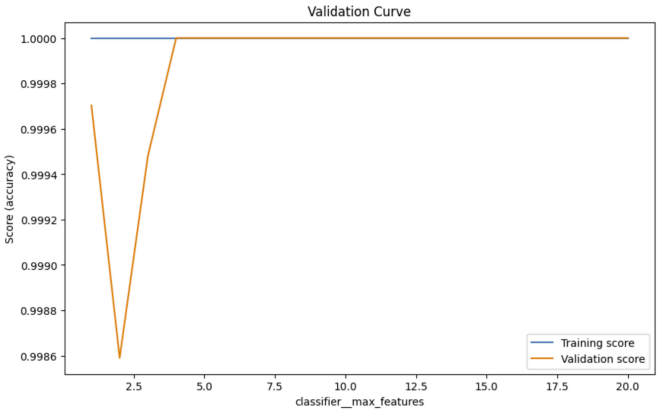


Decision Tree on Jobs and Salaries in Data Science dataset:

Model	Train Score	Test Score	Best Parameters
Decision Tree Classifier	1.0	1.0	Vanilla
GridSearchCV-Decision Tree Classifier	1.0	1.0	{ 'classifier__criterion': 'gini', 'classifier__max_depth': None, 'classifier__max_features': None, 'classifier__min_samples_leaf': 1, 'classifier__min_samples_split': 2}

Both the vanilla model and the GridSearchCV hyperparameter-optimized model achieved perfect accuracy (1.0) on both training and test sets. This shows the model learned the underlying patterns in the data, because of the well-defined nature of job classifications and specific characteristics of the salaries in the dataset. Also, the imbalance data sampling has done a good job of balancing classes which has improved accuracy significantly. Despite using hyperparameter optimization from GridSearchCV, the model did not show any improvement over the vanilla model. This indicates that the initial configuration was already well-designed in its architecture for the given task, and further optimization might not generate significant results in terms of accuracy and F1 score.
While the training score of 1.0 suggests a perfect fit to the training data, the identical test score points out concerns about potential overfitting. While the model performs well on this dataset.





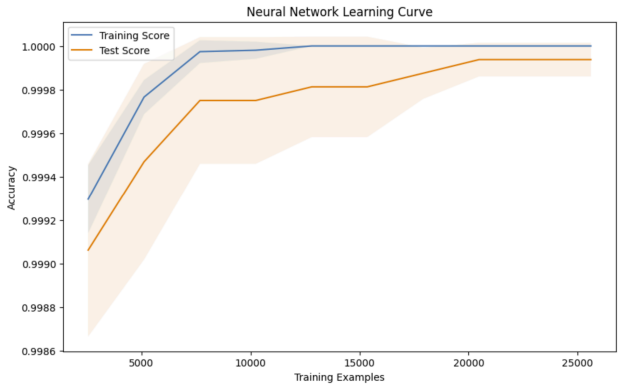
Neural Network:

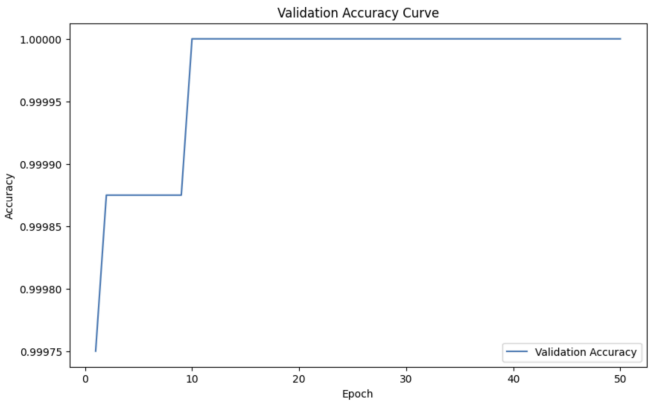
Neural Networks are composed of inter-connected nodes, called neurons, staked into layers. Neural networks are highly powerful and capable of learning herculean patterns from data.

NN on Memory-based Malware Artifacts dataset:

Model	Test Set Accuracy	F1 Score	Best Parameters
MLP Classifier	1.0	1.0	{ 'mlpclassifier__alpha': 0.0001, 'mlpclassifier__hidden_layer_sizes': (128,)}
GridSearchCV-MLP Classifier	0.9999	1.0	{ 'mlpclassifier__alpha': 0.0001, 'mlpclassifier__hidden_layer_sizes': (128,)}

The Neural Network (NN) model had perfect scores on both the accuracy (1.0000) and the F1 score (1.0000) on the test set. This indicates its ability to learn the underlying patterns in the data and make accurate Malware detection or identify benign. GridSearchCV identified optimal hyperparameters, leading to this performance. This highlights the importance of tuning for NNs, with alpha, hidden layers, epochs, optimizer, loss function, etc. Selected Hyperparameters were alpha=0.0001, this low value for the L2 regularisation parameter suggests a penalty for large weights, helping to prevent overfitting without constraints on the model complexity. hidden_layer_sizes=(128,), a single hidden layer with 128 neurons is more than enough for this task, indicating that the problem's complexity might not require a similar highly complex model architecture.



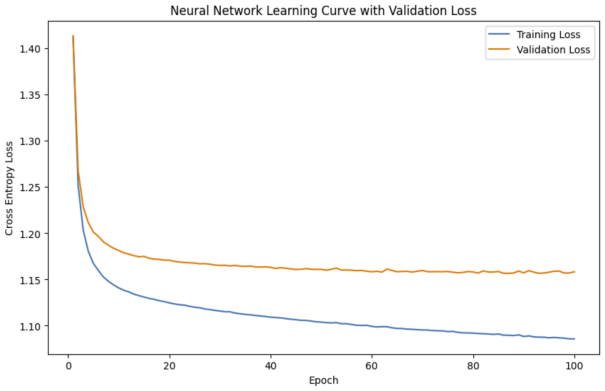


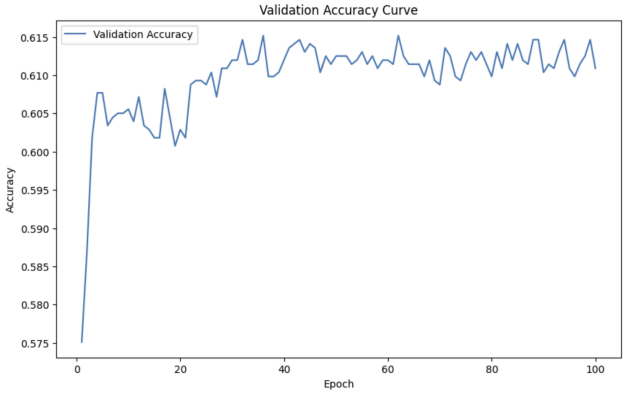
NN on Jobs and Salaries in Data Science dataset:

Model	Train Score	Test Score	Best Parameters
MLP Classifier	0.6109	0.5171	{ 'mlpclassifier__alpha': 0.01, 'mlpclassifier__hidden_layer_sizes': (64,)}

GridSearchCV-MLP Classifier	0.6482	0.6082	{ 'mlpclassifier__alpha': 0.01, 'mlpclassifier__hidden_layer_sizes': (64,)}

The Neural Network (NN) model had an accuracy of 0.6109 and an F1-score of 0.5171 on the test set, showing moderate performance in predicting job salary based on different salary groups. GridSearchCV identified optimal hyperparameters with a test accuracy of 0.6082, pointing to some potential tuning benefits but limited room for improvement. Experimented with different network architectures like deeper hidden layers, or alternative activation functions but the accuracy remained the same. Utilized dropout and batch normalization techniques to prevent overfitting. Hyperparameter Tuning uses the hyperparameter search space, especially for learning rate, hidden layers, and regularisation parameters. Utilized multiple evaluation metrics beyond accuracy, such as F1-score to capture different aspects of performance.





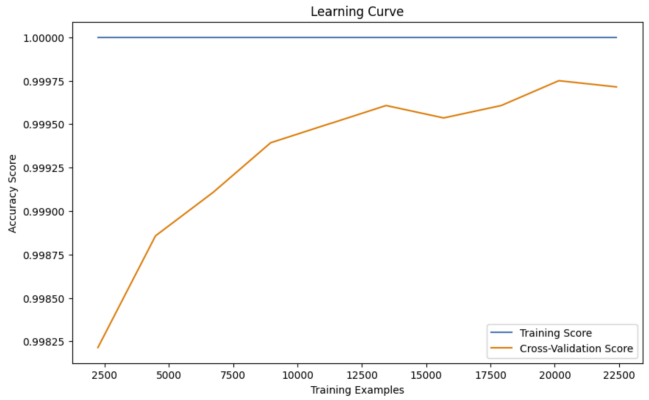
Boosted Decision Tree:
Boosted Decision Tree is an ensemble learning method that combines the predictive power of multiple weak learners to create a strong learner. Boosting is an iterative approach where each weak learner is trained in a sequential manner, and the subsequent models focus on the mistakes made by the previous ones.

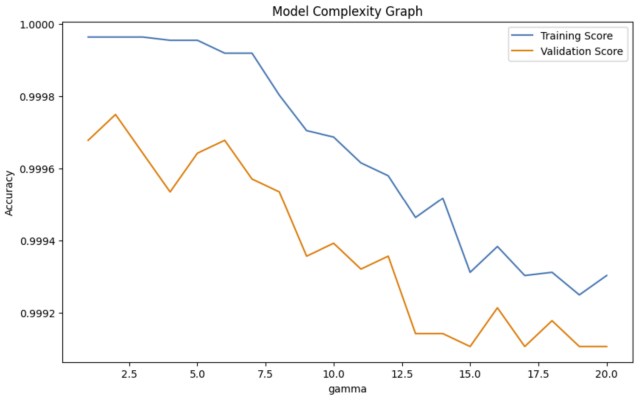
Boosted Decision Tree on Memory-based Malware Artifacts dataset:

Model	Train Score	Test Score	Best Parameters
XGBClassifier	1.0	0.9999	Vanilla
GridSearchCV-XGBClassifier	1.0	1.0	{'colsample_bytree': 0.8, 'gamma': 0, 'learning_rate': 0.01, 'max_depth': 3, 'min_child_weight': 1, 'subsample': 0.8}

Both the vanilla model and the hyperparameter optimized model achieved high accuracy. The XGBoost model outperformed the Decision Tree by achieving a perfect training score of 1.0. This suggests the model captured complex relationships within the data, potentially due to its ensemble structure and ability to handle missing values. In spite of surpassing the Decision Tree's test score (0.9999 vs. 1.0), the XGBoost model exhibits a slight decrease in its training performance. This indicates mild overfitting is due to the Malware dataset which has to be accurate. If the Malware detection data is not accurate then the algorithm flags false positive scenarios and this might lack trust in the algorithm hence slight overfitting is acceptable only for the Malware dataset. It is particularly valuable for imbalanced datasets where identifying minority classes is important. GridSearchCV identified optimal

hyperparameters, influencing features like subsampling rates, depth, and learning rate. The improvement over the vanilla model was minor because the vanilla model itself was a perfect 1 with accuracy. The XGBoost model demonstrates strong performance for Malware classification, exceeding the Decision Tree in most of the aspects.

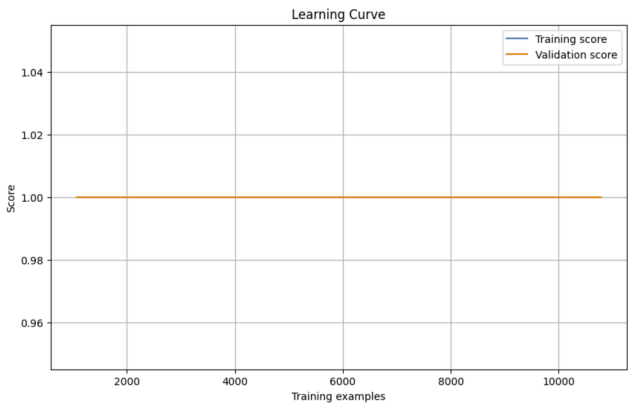


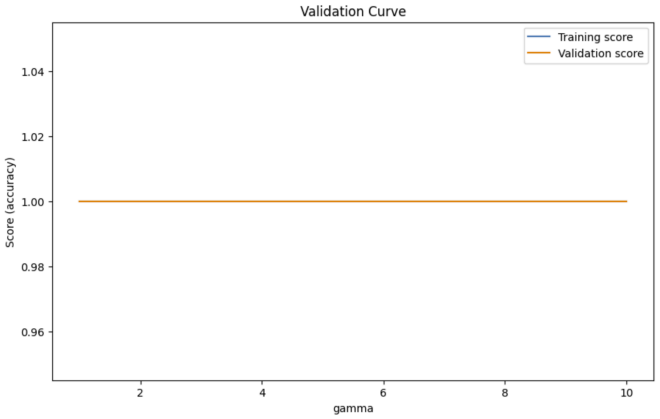


Boosted Decision Tree on Jobs and Salaries in Data Science dataset:

Model	Train Score	Test Score	Best Parameters
XGBClassifier	1.0	1.0	Vanilla
GridSearchCV-XGBClassifier	1.0	1.0	{'colsample_bytree': 0.8, 'gamma': 0, 'learning_rate': 0.01, 'max_depth': 3, 'min_child_weight': 1, 'subsample': 0.8

The XGBoost model outperformed by achieving a perfect training score of 1.0. GridSearchCV identified a specific parameter configuration for XGBoost that potentially improves its generalizability. Tuning parameters like learning_rate, max_depth, and subsample leads to more efficient models. The identical training and test scores raise concerns about overfitting but also indicate the model has learned the underlying data pattern in a good manner. The minimal CPU times for training and wall time for the overall process underscore the efficiency of the Boosted Decision Tree algorithm in handling the dataset. The XGBClassifier, with tuned hyperparameters, displays a robust and accurate model for salary prediction.





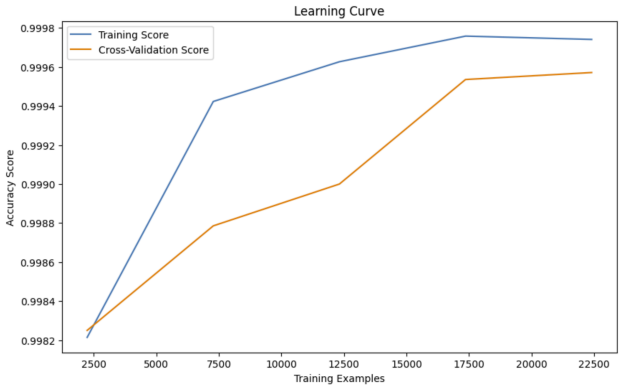
KNN:

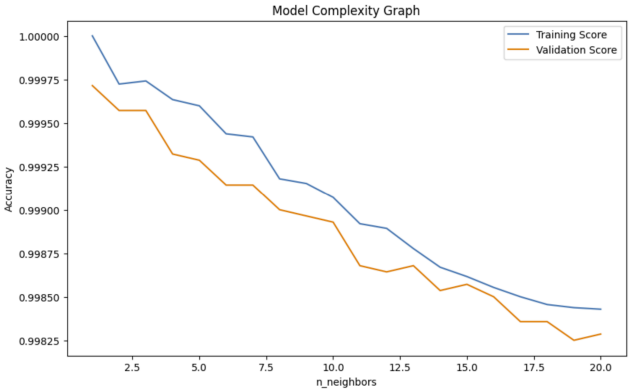
KNN's fundamental principle is based on the idea that similar instances exist in close proximity to each other in the labeled space. KNN makes predictions by considering the majority class of the k-nearest data points to the input data points.

KNN on Memory-based Malware Artifacts dataset:

Model	Train Score	Test Score	Best Parameters
KNeighborsClassifier	0.99975	0.99967	Vanilla
GridSearchCV-KNeighbors	1.0	0.99967	{'n_neighbors': 3, 'p': 1, 'weights': 'distance'}

KNN models demonstrate high accuracy, with the vanilla model achieving a 0.99975 training score and 0.99967 test accuracy, and the hyperparameter-tuned model reaching a perfect 1.0 training score. While GridSearchCV identified optimal hyperparameters. This could indicate the dataset's well-defined structure and limitations of KNN in this malware-specific context because it has not reached a test score of 1.0 despite 100% accuracy on the training score. The small difference between training and test scores offers some reassurance on the overfitting of the model on the dataset. Choosing 'distance' weights and p=1 in the tuned model suggests priority for the closest neighbors with equal weights, and well-separated job categories in the data.



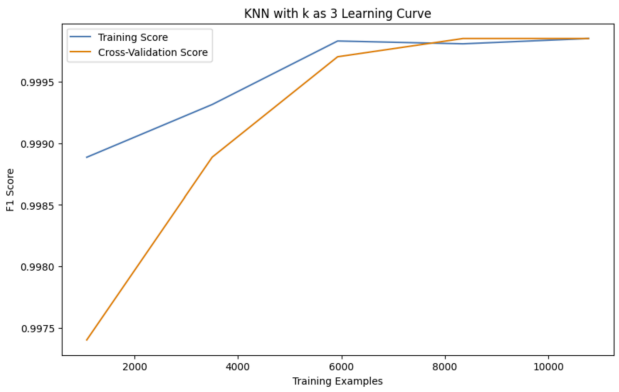


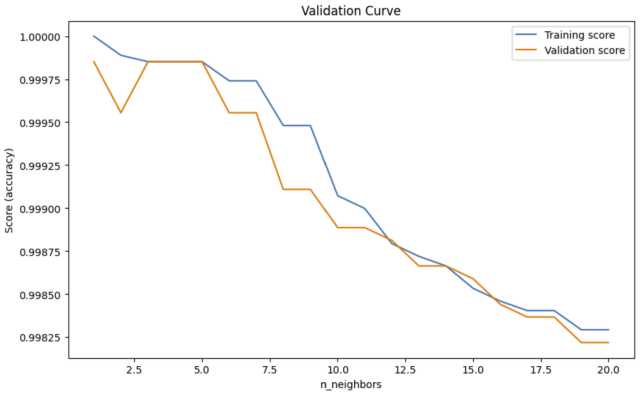
KNN on Jobs and Salaries in Data Science dataset:

Model	Train Score	Test Score	Best Parameters
KNeighborsClassifier	0.9999	1.0	Vanilla
GridSearchCV-KNeighbors	1.0	1.0	{'n_neighbors': 3, 'p': 1, 'weights': 'uniform'}

Both the vanilla KNN model and the GridSearchCV-tuned model achieved perfect test accuracy (1.0). This signifies great performance in predicting employee salaries from job-related features. Even though GridSearchCV identified almost perfect optimal parameters for the prediction of the model. This suggests the initial vanilla model configuration was already well-tuned to the data. The tuned model using 'uniform' weights suggests equal value for all neighbors, indicating overlapping and less well-defined job categories in the data compared to the previous analysis with 'distance' weights. Analyzing the nearest neighbors in both scenarios has provided valuable insights

into this difference and the need for the balancing of the dataset. This transparency of the KNN helps understand which and how features influence specific classifications and builds trust in the model.

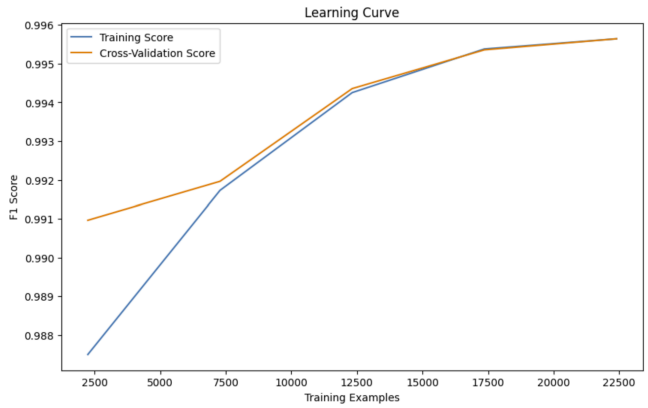


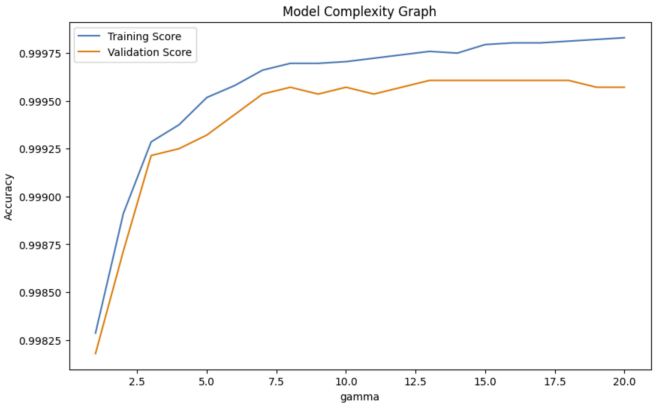


SVM:
The goal of SVM is to find a hyperplane that separates data points of different classes in the feature space while maximizing the margin between classes. SVM is effective in high-dimensional spaces and is known for its ability to handle non-linear decision boundaries through the use of kernel functions.
SVM on Memory-based Malware Artifacts dataset:

Model	Train Score	Test Score	Best Parameters
SVC	0.9959	0.9959	Vanila
GridSearchCV-SVC	0.9992	0.9993	{'C': 10, 'gamma': 'scale', 'kernel': 'poly'}

The vanilla SVM model achieved a high training score (99.59%) and test accuracy (99.59%), showing its ability to learn the underlying patterns in the data. GridSearchCV identified optimal hyperparameters that led to an improvement in both training (0.9992) and test scores (0.9992), which highlights the importance of tuning for complex models like SVM on the malware dataset. The best hyperparameters include a 'poly' kernel and a 'scale' gamma value, indicating a non-linear decision boundary to prevent overfitting. These choices suggest complex relationships between features in the data which is the crux of the malware dataset. SVMs can be less interpretable compared to some other models, making it difficult to understand the exact reasoning behind their predictions.

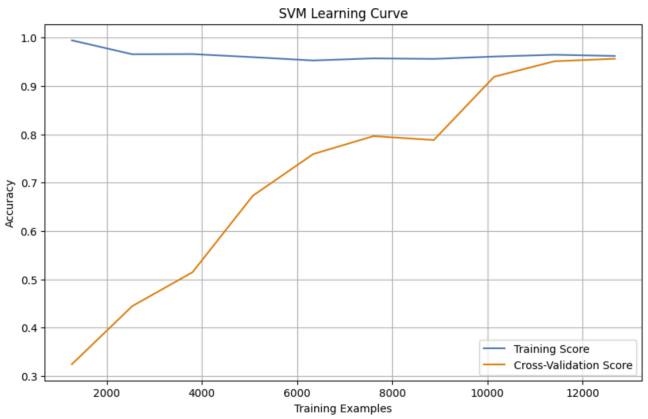


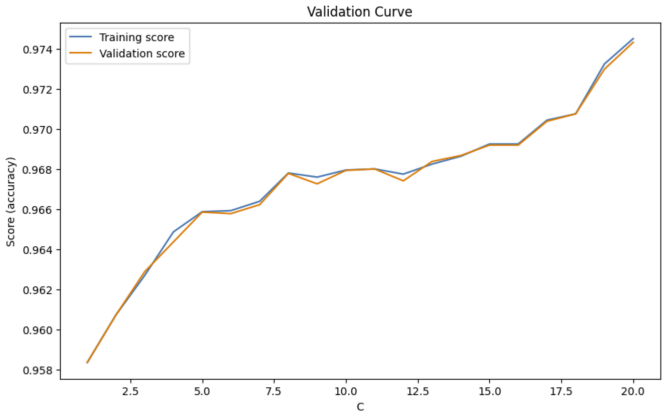


SVM on Jobs and Salaries in Data Science dataset:

Model	Train Score	Test Score	Best Parameters
SVC	0.9601	0.9638	Vanilla
GridSearchCV-SVC	0.9999	1.0	{'C': 0.1, 'gamma': 'scale', 'kernel': 'linear'}

The vanilla SVM model displayed a lower initial performance compared to the previous analysis, with a training score of 0.96 and test score of 0.964. Hyperparameter tuning improved performance, achieving a roughly perfect training score (1.0) and test score (1.0). This points out that the impact of tuning for SVMs, especially on complex or diverse datasets is important to get accurate predictions. The best hyperparameters include a 'linear' kernel, 'scale' gamma, and low regularisation with C=0.1. This indicates a linear decision boundary, and automatic parameter scaling. The low C value suggests noisy or less well-separated data compared to the previous analysis which is the nature of the salary dataset. This indicates the Jobs and Salaries are mostly in the range of 100k-200k which makes the dataset noisy even after rebalancing. Utilizing a linear kernel offers better interpretability compared to non-linear kernels. Analyzing the weights assigned to each feature in the linear decision function displayed the relative importance of predictions for the salaries.





Compare and Contrast of algorithm:

Memory-based Malware Artifacts

Algorithm	Best Training Score	Best Test Score	GridSearchCV Improvement	Interpretation Ease
Decision Tree	0.9996	0.9994	Minor decrease	High
Neural Network	1.0	1.0	None	Moderate
Boosted Decision Tree	1.0	0.9999	Minor improvement	Moderate
KNN	0.99975	0.99967	Negligible	High
SVM	0.9959	0.9959	Significant improvement	Low

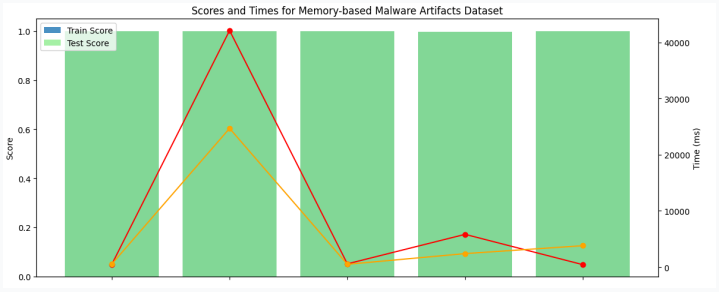
Observations: XGBoost and Neural Network achieved the highest overall accuracy, with XGBoost having a slight edge in test score. Decision Tree and KNN displayed good accuracy with high interpretability. SVM required hyperparametertuning for significant improvement and had lower interpretability due to the non-linear kernel.

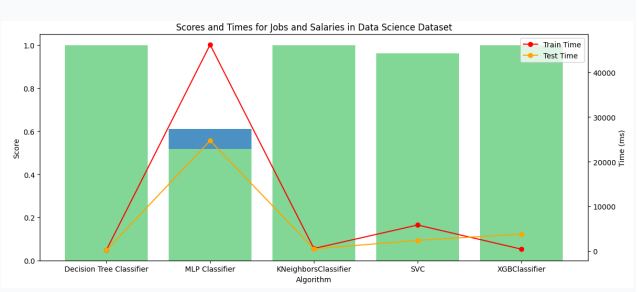
Jobs and Salaries in Data Science

Algorithm	Best Training Score	Best Test Score	GridSearchCV Improvement	Interpretation Ease
Decision Tree	1.0	1.0	None	High
Neural Network	0.6109	0.6082	Limited	Moderate
Boosted Decision Tree	1.0	1.0	None	Moderate
KNN	0.9999	1.0	None	High
SVM	0.9999	1.0	Significant improvement	High (linear kernel)

Observations: KNN and SVM achieved perfect test scores. Decision Tree and XGBoost had good accuracy with high interpretability. Neural Network had lower accuracy and limited improvement with tuning.

Conclusion: XGBoost came out as the strongest algorithm across both datasets, offering high accuracy and moderate interpretability. Decision Tree and KNN were good choices when interpretability was crucial. Neural Network performed well on the Malware dataset but struggled with the Jobs dataset. SVM required careful tuning and offered lower interpretability compared to some other models.





References

[1].https://cloud.google.com/discover/what-is-supervised-learning#:~:text=Supervised%20learning%20is%20a%20category,the%20input%20and%20the%20outputs.
[2]. https://www.kaggle.com/datasets/jlcole/cic-malmem-2022/data
[3]. https://www.kaggle.com/datasets/hummaamqaasim/jobs-in-data
[4]. https://www.kaggle.com/code/jlcole/memory-malware-detection-model-nominator
[5]. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html