CS482 Final Project Report
Course: Introduction to AI
Project Title: Comparative Analysis of Five AI Models: Mushroom Poison Classification
Submitted by: Ivan Ching, Diego Borne
Date: 12/12/2024
https://github.com/borne-diego/CS-482-Final-Project

Introduction and Objective

Objective:
The objective of this project is to create, train, and evaluate multiple machine learning models and conduct a comparative analysis on a dataset of our choice to demonstrate a comprehensive understanding of key concepts, technique and algorithm in Artificial Intelligence.

Problem Statement:
This project uses data from a mushroom dataset to design, implement, and evaluate machine learning models that help indicate the accuracy of how edible or poisonous a mushroom can be. The primary objective is to analyze the features and compare the performance of different models in terms of accuracy, precision, recall, and F1-score.

Model Number	Model Name	Purpose
1	XGBoost	To make highly precise predictions on complex data by combining the predictions of multiple decision trees, optimizing for speed and accuracy.
2	Neural Networks	To learn from data and make decisions based on complex non linear relationships between inputs and output information.
3	Linear Regression	To predict the value of a variable based on the values of another value.
4	Support Vector Machine	To classify data by finding the optimal bound that maximizes the distance between different classes, separating them with the largest possible margin.
5	K-Nearest Neighbors	To make classifications and/or

		predictions about the grouping of an individual datapoint.

Justifications for Model selection

Model Name	**Justifications**
XGBoost	-Handles large data sets efficiently and deals with categorical data - Excels in high-dimensional feature spaces by automatically determining feature importance and interactions
Neural Networks	- Suitable for large datasets with complex relationships between features. - Capable of handling one-hot encoded categorical variables
Linear Regression	-Predicts the relationship between dependent and independent variables.
Support Vector Machine	- Works well for binary classification problems - Effective in high-dimensional spaces and handles non-linear boundaries
K-Nearest Neighbors	-Intuitive for binary classification problems.

Model Description

Model Number	**Model Name**	**Architecture Detail**	**Key Features**
1	XGBoost	-Decision Trees trained sequentially to minimize errors from prior trees.	-Efficient handling of missing and categorical data. -Fast and scalable
2	Neural Networks	-Multilayer Perceptron(MLP) for classification. -Includes input, hidden and output layers. -Activation function (ReLU, Sigmoid, etc.).	-Can model complex nonlinear relationships

		-Trained using backpropagation.	
3	Linear Regression	-Linear equation that models the relationship between independent variables and the target.	-Simple and interpretable
4	Support Vector Machine	- Maximizes the margin between data points of different classes using support vectors. -Linear and nonlinear boundaries using kernel functions.	-Effective for binary classification tasks -Handles both linear and non-linear boundaries
5	K-Nearest Neighbors	-- Lazy learning algorithm that classifies based on the majority vote of *k* k-nearest data points in feature space.	-Simple and intuitive.

Dataset Description

Dataset Attribute	**Description**
Name	Mushroom Dataset
Source	https://www.kaggle.com/datasets/prishasawhney/mushroom-dataset
Size	9 columns including the target class 54,034 rows
Class Distribution	2 classes (Edible or Poisonous) With 29,675 poisonous and 24,360 edible.
Categories	Cap Diameter Cap Shape Gill Attachment Gill Color Stem Height Stem Width Stem Color

	Season Target Class - Edible or Poisonous? (0 or 1)
Preprocessing Steps	-Convert categorical attributes using one-hot encoding -Check and handle missing values -Normalize or scale features for models -Divide the dataset into training and testing subsets -Encode the target class

Dataset Justification:
The dataset chosen is well suited for binary classification tasks and models between its target class. The dataset contains nine columns of classes with 54,034 total samples which is large enough to train and generate machine learning models effectively while remaining manageable. The dataset contains categorical features, making it ideal for models such as XGBoost and Neural Network models. In this dataset specifically, the data is clean with no missing values that would need to go through preprocessing.

Experimental Setup

Metric to obtain
Accuracy
Precision
Recall
F1 Score

Model Name	**Hyperparameter 1**	**Hyperparameter 2**	**Hyperparameter 3**	**Hyperparameter 4**	**Hyperparameter 5**
XGBoost	Learning Rate	Max Depth	Regulaizion (L1/L2)	N/A	N/A
Neural Networks	Learning Rate	# of hidden layers	# of units per layer	Activation Functions	Batch size
Linear Regression	Regularization	Optimization Algorithm	Fit intercept	N/A	N/A
Support Vector Machine	Degree	C	Gamma	Kernels	N/A

K-Nearest Neighbors	# of Neighbors	Distance	Weights	N/A	N/A

Environment Setup

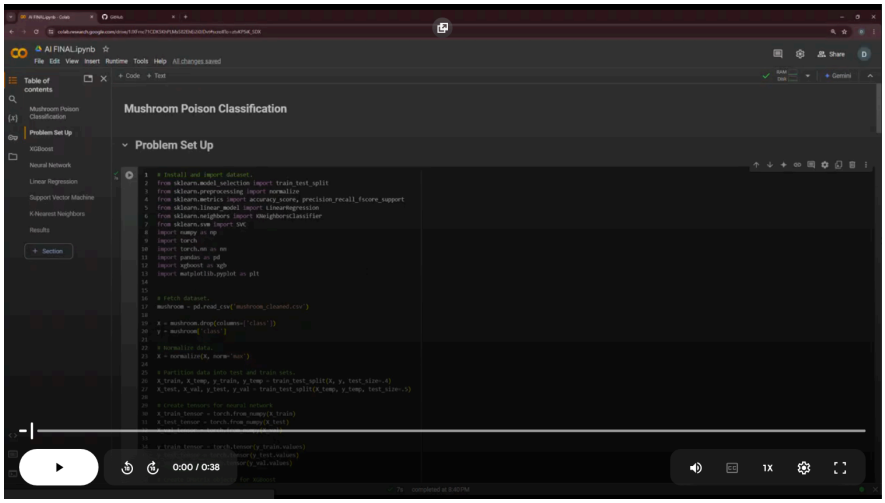
Component	**Specifications**
Operating System	Windows 10/11
Software Version	All code was run via Google Colab. Compatible with other similar Jupyter Notebook environments.
Hardware	Google Colab provided CPUs.
Code Base	Colab:https://colab.research.google.com/drive/1JXFrnc71CDK5KhPLMsS82EhEi2i0JDvt?usp=sharing GitHub:https://github.com/borne-diego/CS-482-Final-Project

Result and Analysis

Model Name	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1 Score (%)**
XGBoost	96.87	96.86	96.82	96.84
Neural Networks	83.05	83.66	82.24	82.58
Linear Regression	59.09	58.55	58.46	58.48
Support Vector Machine	60.06	61.25	61.03	60.01
K-Nearest Neighbors	93.99	93.92	93.96	93.94

CS482 Final Project Report

https://youtu.be/TqPUETFKyXw



Comparative Analysis:

XGBoost emerged as the top performing model with the highest accuracy, precision, recall, and F1-Score at 96.87%, 96.86%, 96.82%, 96.84%, respectively. This was the expected result as XGBoost models handle categorical data exceptionally well, reducing overfitting and bias, which would result in an overall higher score compared to the rest.

K-Nearest Neighbors (KNN) follows closely behind the XGBoost Model with an accuracy of 93.99% and similar scores between 93.93 to 93.96 for precision, recall, and F1-score.

Neural Networks model results in a model with 83.05% accuracy, 83.66% precision, 82.24% recall, and 82.58% f1-score. This lower performance compared to the XGBoost and the K-Nearest Neighbor model is likely due to hyperparameter tuning or overfitting on the training data.

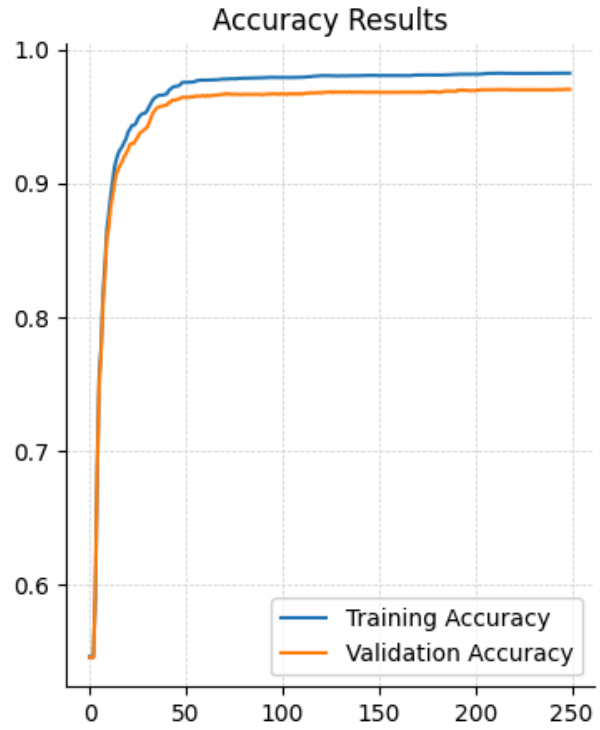
Both **Support Vector Machine** and **Linear Regression** models both underperformed on this dataset. The **Support Vector Machine** achieved slightly better results with 60.06% accuracy, 61.25% precision, 61.03% recall, and 60.01 f1-score compared to the results from the **Linear Regression** model of 59.09% accuracy, 58.55% precision, 58.46% recall, 58.48% f1-score. These results from these models are the result of their limitations as machine learning models on this dataset.

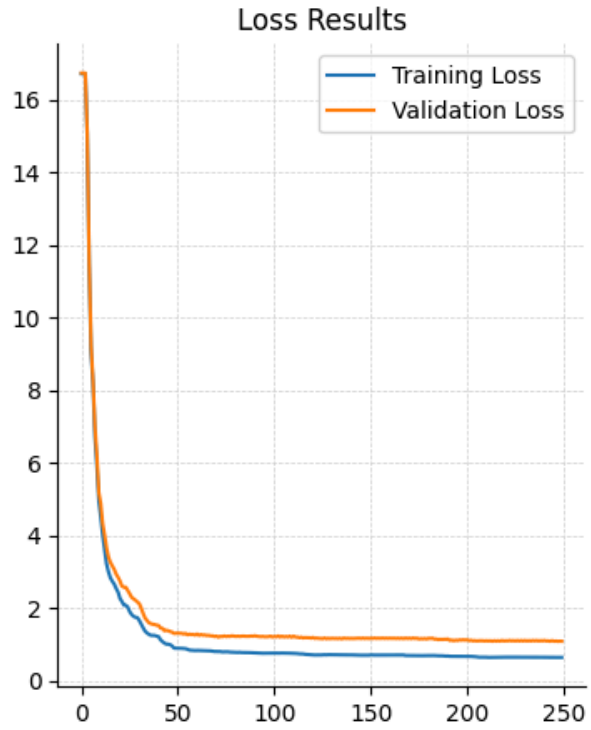
Error Analysis:

Sources of error in this model could be from the high dimensionality of the dataset or hyperparameter tuning of our models. Models like Neural Networks and SVM may have achieve higher results with more hyperparameter optimization. Some specific errors for each model that might have occurred would be KNN's reliance on distance metrics, likely reduced its performance consistency. There can be over fitting in the Neural Networks model.

CS482 Final Project Report

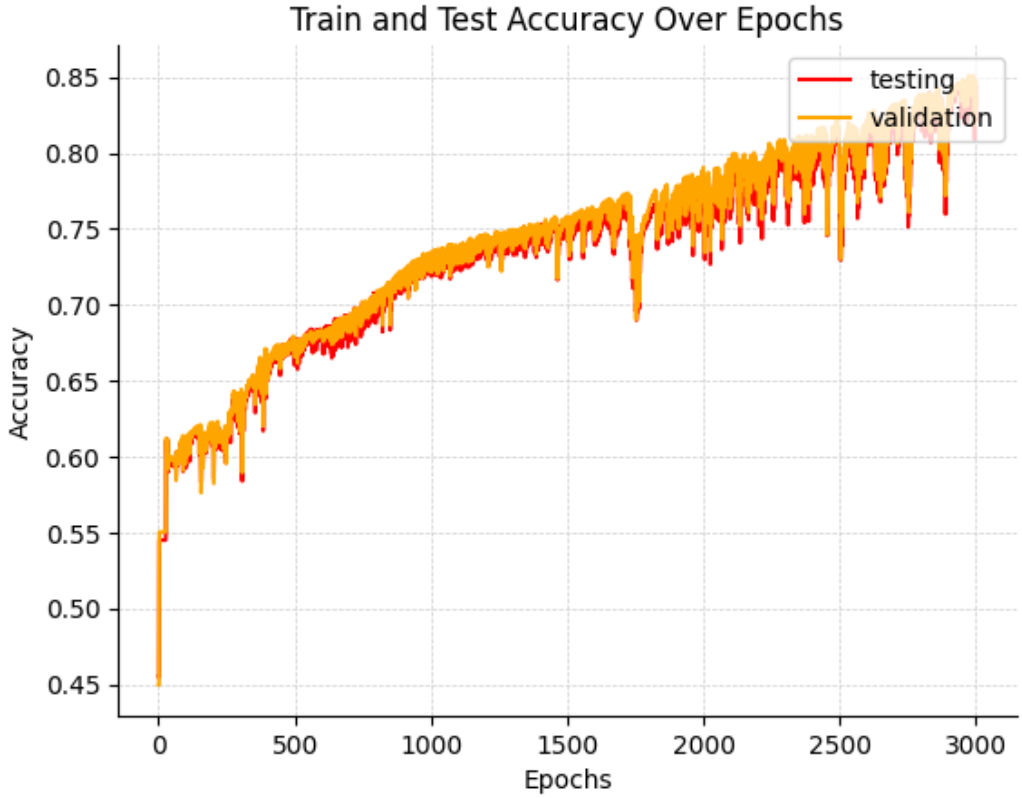
XGBoost Accuracy and Loss

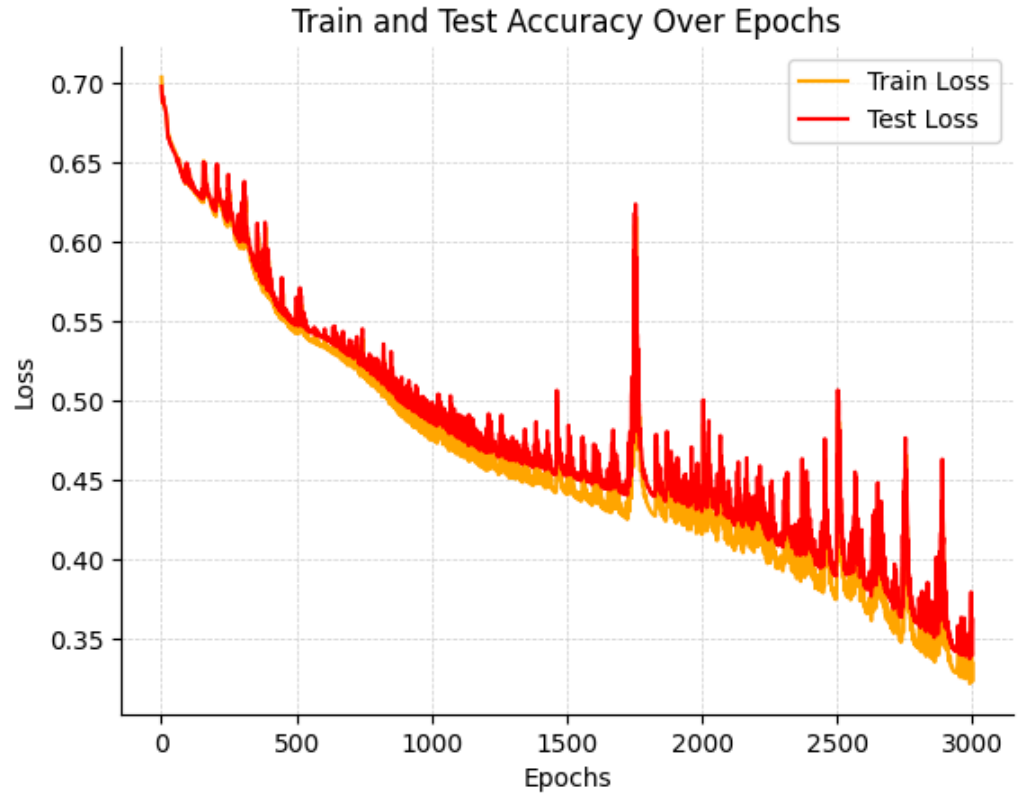




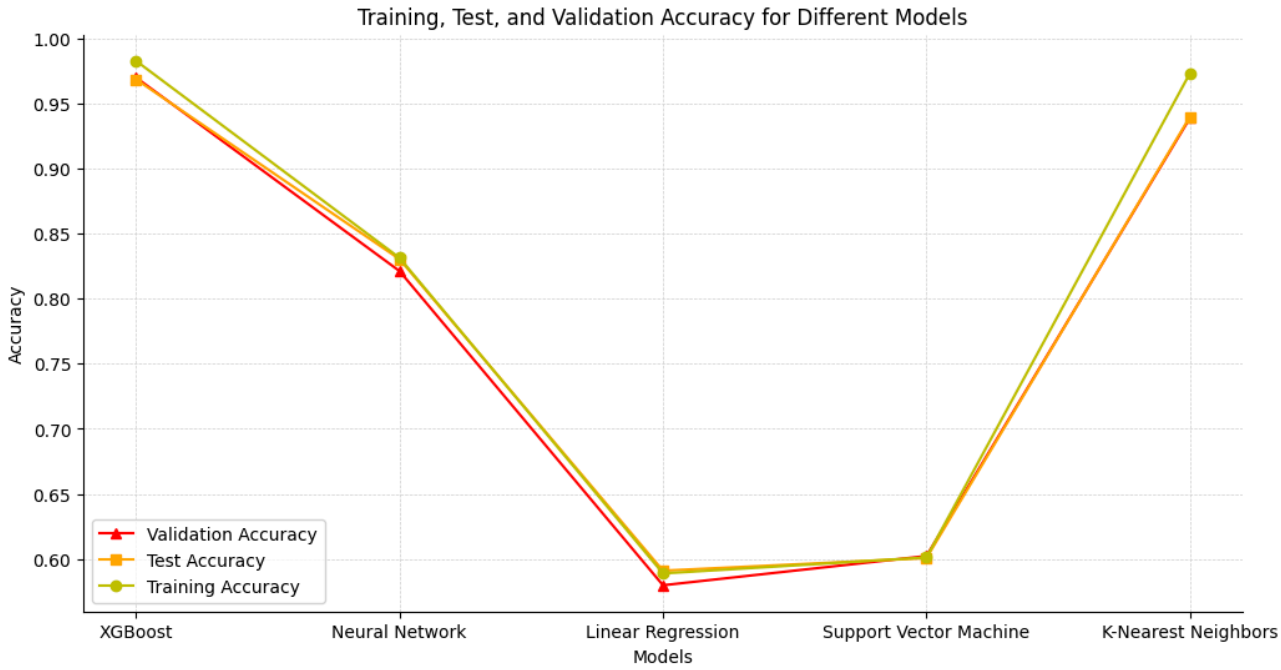
CS482 Final Project Report

Neural Network Accuracy and Loss





Accuracy Results



Discussion and Insight

Results Interpretation:

XGBoost achieved the highest performance across all metrics, with its ability to handle categorical features to provide impressive predictions, making it the best choice for this dataset.
Neural Networks delivered reasonable performance but was outperformed by XGBoost and K-Nearest Neighbors.
Linear Regression performed poorly in this dataset.
Support Vector Machine delivered moderate performance. This can be changed with further hyperparameter testing to yield better results.
K-Nearest Neighbors performed exceptionally well, with results almost matching the results from the XGBoost model.

Limitations:

The dataset is categorical, requiting different types of encoding which can negatively affect the distance-based models. Insufficient amount of hyperparameter tuning can negatively affect the performance of the neural network and support vector machine models. Computational constraints such as scalability might have hindered the ability of some models results compared to others.

Future Directions:

Improvement to this project in the future are optimizations on the dataset and further testing of hyperparameter optimization. This can improve the results on the different models we used such as neural networks. There can be cases where the project would include additional models to experiment and get results from. Lastly, optimization for models for scalability to improve applications can yield different results.

Conclusion

This Artificial Intelligence project demonstrates the applications of multiple machine learning models to classify the mushrooms in the dataset as either edible or poisonous based on their features. This project revealed the strengths and weaknesses of each model and their suitability in relation to this dataset and other similar datasets. XGBoost emerged as the top performing model with the highest accuracy, precision, recall, and F1-Score with an average of 96%. K-Nearest Neighbors performed exceptionally well, with results almost matching the results from the XGBoost model. Neural Networks delivered reasonable performance but was outperformed by XGBoost and K-Nearest Neighbors. Both Support Vector Machine and Linear Regression models both underperformed on this dataset. In conclusion, these insights highlight the importance of model selection and trade-offs of selected models in machine learning applications.

For this project, we decided to divide the workload evenly to ensure efficiency. Diego took responsibility for implementing three machine learning models which were XGBoost, K-Nearest Neighbors, and Linear Regression. Diego visualized the results and documented his findings. Ivan managed the implementation and analysis of two machine learning models which were Neural Networks and Support Vector Machine (SVM). Ivan also visualized the results and documented his findings. For the project's documentation, the workload was also divided into two sections. Diego was tasked with the first part of the documentation which included the Introduction and Objective, Justifications for Model selection, Model Descriptions, Dataset Description. Ivan was tasked with the second half of the documentations which includes Experimental Setup, Results and Analysis, Discussion and Insight, and Conclusion. This division of workload allows the team members to contribute equally.