CS 590 ETHICAL FALL 2024 HW1-EXPLAINABILITY DRASHTI SNEHALKUMAR PATEL

Homework Understanding :

The objective of this homework is that we have to train, test, and analyze three machine learning models: Decision Tree, Logistic Regression, and Neural Network (MLP) and focus on understanding the explainability of each by performing performance analysis .

Details mentioned :

The train and test files of data were given with 5 columns. We have to train ,test(over both train and test set) and explain three models (Decision Tree, Logistic Regression, and Neural Network) using default parameters. Each model uses the first 4 columns (salary, age, credit score, debt) as features and approved as the target, with separate functions to record performance and identify influential features.

ANALYSIS DONE

1.DECISION TREE MODEL [DT_Explain(train_file, test_file)]

a.The Decision Tree was trained on the credit_train.xlsx data using the DecisionTreeClassifier from scikit-learn with default parameters and tested on both the train and test data.

b. No feature normalization was required for this model.Unlike models such as Logistic Regression and Neural Networks, Decision Trees do not require feature normalization. Since Decision Trees split data based on threshold values rather than distance-based computations, normalizing the input features does not impact the model's performance.

c. I explored the plot_tree from scikitlearn to plot the tree. Also for clarity and readability , I saved the image as .png form with a resolution of 300 dpi.

Moving to explainability, I visualized the Decision Tree using plot_tree from scikit-learn as hinted in the homework document and it provides a clear visual representation of how to interpret features and helps in explainability.

Decision Trees are intrinsically interpretable models; that is, their decision-making process can be directly understood by tracing the splits at each node. Unlike black-box models like Neural Networks, Decision Trees allow us to see which features contribute to predictions and how the model arrives at a decision.

Some questions which came up in my mind while working on this were :

1.How did tree plot help in decisions?

We can always see the splitting and branching and analyse our results and decisions of important features . Thus , I interpret explainability by tree plot that the root node was the credit score, confirming again that it was the most relevant feature.This makes sense logically also as creditworth is often very important while financial decisions in the real world as well.

Next came debt, with very little reliance on salary and age, placed in much deeper nodes.
Colors also help interpret how classification was distributed at each node, hence facilitating more insight into the decisioning process.
By and large, plot_tree depicted feature prioritization and was easy to understand for me.

2.Is it overfitting the training data?

The deep tree structure suggested overfitting as we get 100% train accuracy and test accuracy being 97%.

LOGISTIC REGRESSION MODEL [LR_Explain(train_file, test_file)]

We trained the Logistic Regression model on the dataset after normalizing the features using L2 normalization and understanding the documentation given in the hw document.
Why was normalisation needed?
I printed the summary of our training data for understanding the need of normalisation using .describe() .We needed normalization because the salary and debt had much larger values, up to 250,000 and 149,000, respectively, while credit score and age were much smaller. This caused the feature influence to be out of balance. Since Logistic Regression is sensitive to feature scales, normalization made sure all features contributed proportionally and are not dominated by high-value ones.
For explainability : we use feature importance through model coefficients (coef_) to understand which features had the most impact on predictions.
Logistic Regression represents a linear model, whereby it relies on the weighted sum of the input features in making its decisions.
Coefficients (coef_) express the strength and direction of influence each feature has on the prediction.
The positive coefficients imply that this feature increases the probability of being granted the loan, while negative means decreasing the approval probability. The larger the value of the coefficient , the stronger its impact is on the models decision. As in our model,

Feature	value	impact
Credit score	(1.1641)	Highest positive factor
Debt	(-7.4074)	Highest negative factor
Salary	(-0.9819)	Little negative factor
Age	(0.0349)	Least impactful factor

NEURAL NETWORK MODEL[MLP_Explain(train_file, test_file)]

For training the Neural Network (MLP), I used the MLPClassifier from scikit-learn with two hidden layers (100 and 50 neurons) and trained it on the provided dataset. Since neural networks are sensitive to feature scaling, I applied L2 normalization using normalize() from scikit-learn to ensure that all features contributed proportionally to the learning process.

For explainability , I used:

a. Permutation Importance

I chose Permutation Importance because it provides a simple yet effective way to measure feature influence in neural networks, which lack built-in importance scores. By shuffling feature values and observing the drop in performance, this method highlights key contributors. It's especially useful for complex models like MLP, where direct interpretability is limited, and it complements SHAP’s more individualized explanations.

b.SHAP (SHapley Additive Explanations)

I used SHAP to enhance explainability by understanding how each feature influenced the individual predictions. Unlike Permutation Importance, which provides a global view, SHAP assigns a contribution value to each feature per instance, making it useful for interpreting complex models like MLP. Since neural networks act as black boxes, SHAP is helpful in breaking down the predictions by showing the impact of feature values that allow deep insights into model behavior.

From our results, debt had the highest effect on the prediction, followed by credit score. This also was the case in Permutation Importance, reinforcing that these two features played the most important roles in making these decisions.

FEATURE ANALYSIS

DECISION TREE

Most important feature: creditscore

Credit Score as it the root node with a split at credit score ≤ 680.5 and splitting into debt levels where (debt ≤ 3.5) increases approval chances of approvals .A higher credit score means better financial reliability, which strongly impacts loan approval decisions.Salary and age appear in deeper nodes, meaning they influence only borderline cases whereas age is of the least importance.

Evaluation metrics :

The model performed with very high accuracy of 97% showing correct classification of most loan approvals .

96.78% Precision and 97.51% recall are also high so wwe can conclude that our model avoids approval of unqualified applicants and rarely rejects actual approvals.

The 100% train accuracy clearly suggests overfitting, meaning the model memorized training data rather than learning general patterns. This could lead to weaker performance on unseen data. I think pruning the tree can help model generalise well .

LOGISTIC REGRESSION

Most important feature: creditscore

It had the highest importance score, directly indicating financial reliability.

As it is the most relevant feature to grant approval, while high debt strongly decreases its chances.I determined this by analyzing the feature coefficients from the trained model, where larger absolute values indicate stronger influence on predictions. Among all features, credit score had the highest positive coefficient of 1.1641, meaning it played the most critical role in determining approvals.

Evaluation metrics :

The Logistic Regression model was able to generalize fairly well, with a train accuracy of 81.65% and a test accuracy of 78.9%, which shows some performance drops on unseen data.

It also does an excellent job of approving qualified applicants with an accuracy of 81.87% in precision and a recall of 76.71%, though capturing most actual approvals with slightly more false negatives than Decision Tree.

The F1 score of 79.21% indicates that the model balances precision and recall well, though this model could be further improved by hyperparameter tuning or feature engineering.

NEURAL NETWORK

Most important feature: Debt

The most important feature in our neural network model is debt, as indicated by both Permutation Importance (0.3049) and SHAP analysis. This suggests that debt has the strongest influence on the model's predictions, meaning changes in debt levels significantly impact whether an applicant is approved or not. This makes sense, as higher debt levels typically correlate with higher financial risk, making it a key factor in loan approval decisions.

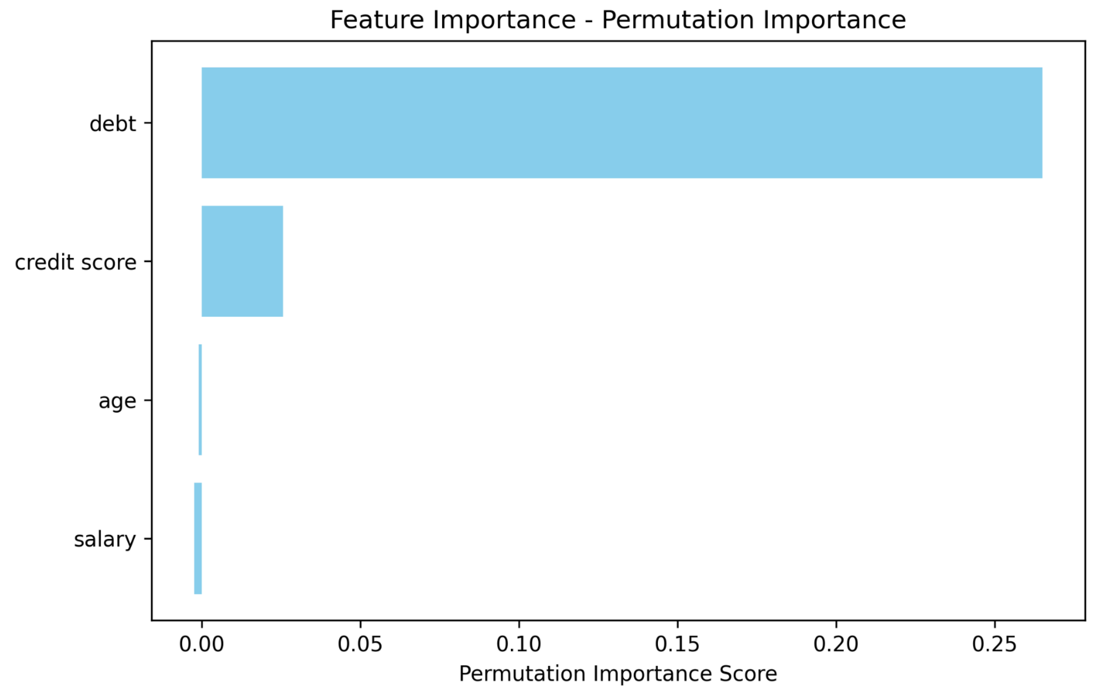
Evaluation metrics :

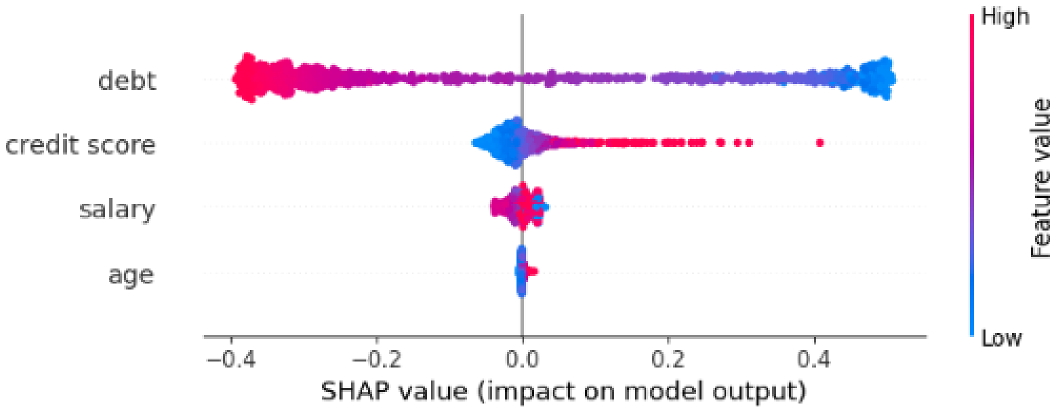
Generalizing well, the Neural Network model had a train accuracy of 86.7% and a test accuracy of 85% to show how well it could perform on data it had never seen before without overfitting.

With 94.31% precision, the model allows only qualified applications, which will result in minimal false approvals; however, having a recall of 75.95% makes it miss out on some actual approvals, therefore being conservative.

With the F1 score at 84.14%, this shows a pretty good balance between precision and recall, though better recall might enable the model to correctly identify even more applicants whose applications get approved. Such an optimization may come through hyperparameter tuning or tuning of the decision threshold.

Below is the output of Permutation importance and SHAP:





RESULTS

A. Result Table with accuracy

Model	Train_Accuracy	Test Accuracy
Decision Tree	1.00	0.97
Logistic Regression	0.8165	0.789
Neural Network	0.8670	0.85

The Decision Tree had perfect training accuracy (100%) but dropped to 97% on the test set, clearly showing overfitting. I could improve this by pruning or limiting depth to help it generalize better. Logistic Regression performed decently with 81.65% train accuracy and 78.9% test accuracy, meaning it generalizes better than the Decision Tree but still has room for improvement. Feature engineering or regularization tuning could help optimize it further. The Neural Network showed 86.7% train accuracy and 85% test accuracy, making it the most balanced model in terms of generalization. Since the gap between training and testing accuracy is small, fine-tuning hyperparameters like hidden layers or learning rates could further boost its performance.

B. Result Table with all evaluation metrics

Model	Train Accuracy	Test Accuracy	Precision	Recall	F1 Score
Decision Tree	1.00	0.97	0.9678	0.9751	0.9714
Logistic Regression	0.8165	0.789	0.8183	0.7671	0.7921
Neural Network	0.867	0.85	0.9431	0.7595	0.8414

For better understandability I decided to calculate other metrics also for evaluation . The Decision Tree demonstrated excellent precision, recall, and F1 score, showing its ability to effectively balance false positives and true positives. In comparison, both Logistic Regression and the Neural Network performed moderately, with Logistic Regression leaning slightly towards precision and the Neural Network achieving a better balance between precision and recall.

3. Helpful features identification and why? (Modelwise)

For the Decision Tree, I used `plot_tree` visualization and `feature_importances_` to analyze the splits. The feature_importances_ attribute showed that credit score was the most important feature, followed by debt. This makes sense because a high credit score depicts financial reliability, while debt may indicate financial risk. Salary and age were the least important, probably because their relationship with loan approvals is not as direct or consistent.

Also, considering the coefficients of Logistic Regression, credit score (1.1641) had the highest positive influence on approvals, indicating its importance in prediction. Debt(-7.4074) had a serious negative impact;

salary was moderate negativity, and probably because different applicants can have different thresholding; and finally, age, which contributes the least, hence it is not a strong indicator for this task.
I used **Permutation Importance** and **SHAP** for explainability. Unlike the other models, **debt** had the highest importance (0.3049), meaning the model detected a strong nonlinear relationship between debt levels and approvals. SHAP confirmed that both **debt and credit score** were major drivers, but the Neural Network captured deeper patterns.

4. Deciding Best features?

Decision Tree:

Best Feature: Credit Score as it the root node with a split at credit score ≤ 680.5 and debt levels(debt ≤ 3.5) increases approval chances of approvals .

Reason: It had the highest importance score, directly indicating financial reliability.

Logistic Regression:

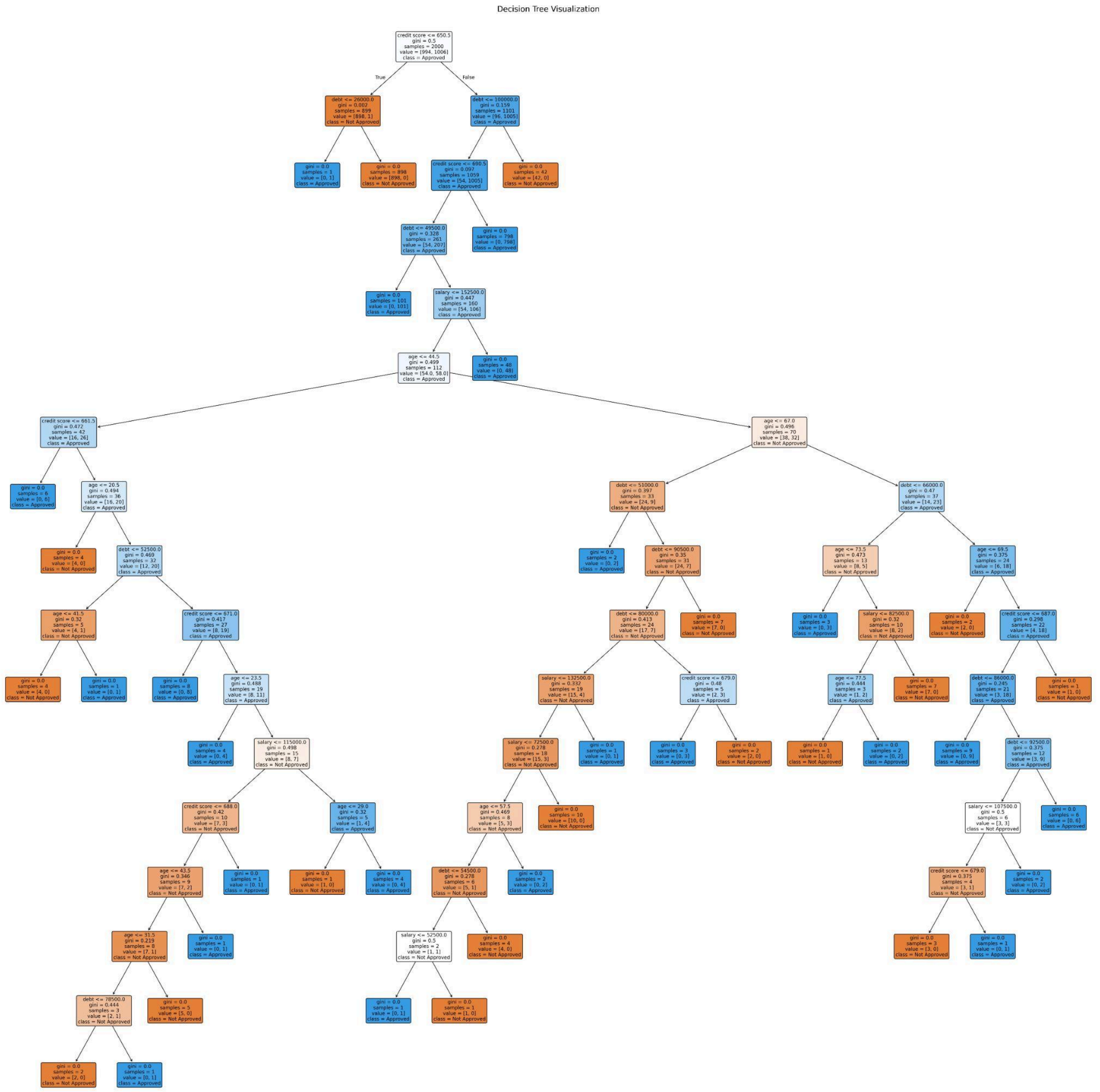
Best Feature: Credit Score

Reason: It had the strongest positive coefficient, showing its strong influence on predictions.

Neural Network:

Best Feature: Debt

6. Dtree diagram



ADDITIONAL DISCUSSION

The Decision Tree and Logistic Regression have both picked credit score as the most important feature, following a structured and linear approach. On one side, the Decision Tree overfitted, while on the other, Logistic Regression generalized better but lacked depth in capturing the relationships.

In contrast, debt was the most informative feature for the neural network, MLP, showing its capability to capture nonlinear patterns. It generalized best and struck a balance in performance to show deeper dependencies between financial factors. The contrast here shows that while simpler models rely on credit score, debt is recognized by more complex models such as MLP as a strong determinant in the approval of loans.

PROBLEMS AND CHALLENGES ENCOUNTERED

Since this was the first introductory assignment and ethical AI is a new field , it took a little time to understand the core concepts of explainability and strategies.

It took time to work with normalisation and in logistic regression and neural networks .

Finding the right number of hidden layers and neurons was challenging. Too few caused underfitting, while too many increased training time and overfitting risks.

CONCLUSION

This homework gave me a deeper understanding of how different models handle predictions and what makes each one unique , clearing concept of explainability. While some models, like the Decision Tree, were highly accurate, they tended to overfit, while others, like Logistic Regression, were more interpretable but missed some complexities. The Neural Network captured deeper patterns, prioritizing debt over credit score, which was an interesting contrast to the other models.Overall, this task showed me how feature scaling, model selection, and interpretability techniques play a huge role in building effective machine learning models.

Note: Leveraged AI tools like gpt for referencing , concept clarity and code understanding.