Part One: Group Work

(1.0) Introduction

This group project implemented and compared two supervised learning algorithms, Decision Tree (DT) and Logistic Regression (LR), on the KDD Cup 1999 intrusion detection dataset using Apache Spark's Mllib (UCI KDD Archive, 1999). The objective was to assess how well both models performed on high-dimensional and categorically-dense data in a distributed environment. Our primary intention in this experiment was to model a pragmatic pipeline to overcome some of the major challenges (e.g., scalability and reproducibility) that machine learning is often susceptible to in 'real world' and big data contexts. Both the DT and LR classifiers were run across ten random seeds for generalisability and evaluated against multiple classification metrics. In addition to implementing and benchmarking both models, we further validated our findings via statistical significance testing (paired t-Tests) and z-score standardisation. This approach aligned with the assignment's emphasis on utilising cross-domain expertise, like statistical modelling and Spark ML-tooling, to improve the experiment's general quality and rigour.

(1.a) Method

Decision Tree (DT) Pseudo-code

```
FOR EACH seed in seed-1 to seed-10:
    Load dataset
    Preprocess dataset
        • Convert all categorical variables into numerical form
        • Merge all feature columns into a single feature representation
        • Convert target labels into numerical format
    Split dataset into train/test sets on current seed
    Initialise a DT classifier
    Train DT classifier on training set
    Test the trained model's predictions on the testing set
    Evaluate the Acc./F1/Precision/Recall Predictions
        • Acc = (TP+TN) / (TP+TN+FP+FN)
        • P = TP / (TP+FP)
        • R = TP / (TP+FN)
        • F1 = 2*(P*R) / (P+R)
        • AUC = Area under ROC curve
    Record run results and time of current seed
END FOR
```

Logistic Regression (LR) Pseudo-code

```
FOR EACH seed in seed-1 to seed-10:
    Load dataset
    Preprocess dataset
        • Convert all categorical variables into numerical form
        • Merge all feature columns into a single feature representation
        • Convert target labels into numerical format
    Split dataset into train/test sets on current seed
    Initialise a Logistic Regression model
    Train Log-Reg model on training set
    Test the trained model's predictions on the testing set
    Evaluate the Acc./F1/Precision/Recall/AUC Predictions
        • Acc = (TP+TN) / (TP+TN+FP+FN)
        • P = TP / (TP+FP)
        • R = TP / (TP+FN)
        • F1 = 2*(P*R) / (P+R)
        • AUC = Area under ROC curve
    Record run results and time of current seed
END FOR
```

(1.b) README.txt:

Please see README for install & run instructions.

(1.c) Reported Raw Results: DT/LR Output (10 Seeds)

The results of running the DT and LR models across ten seeds can be found in *Tables 4-8*, below. In looking solely at the reported raw results, we observed that DT performed better on average than LR for both training and testing data (e.g., Accuracy ~95.2% vs. ~93.0%, *Tables 1-2*) and executed in slightly less time (5.055s vs. 5.929s, *Table 3*).

Logistic Regression

Logistic Regression (LR) : TRAIN RESULTS			
Train Accuracy		**Train ROC AUC**	
MIN = 0.928	**AVG = 0.930**	MIN = 0.980	**AVG = 0.981**
MAX = 0.931	STD = 0.001	MAX = 0.981	STD = 0.000
Train Precision		**Train Recall**	
MIN = 0.929	**AVG = 0.931**	MIN = 0.928	**AVG = 0.930**
MAX = 0.931	STD = 0.001	MAX = 0.931	STD = 0.001
F1 Score			
MIN = 0.928	**AVG = 0.930**		
MAX = 0.931	STD = 0.001		
Logistic Regression (LR): TEST RESULTS			
Test Accuracy		**Test ROC AUC**	
MIN = 0.928	**AVG = 0.930**	MIN = 0.979	**AVG = 0.981**
MAX = 0.932	STD = 0.001	MAX = 0.982	STD = 0.001
Test Precision		**Test Recall**	
MIN = 0.928	**AVG = 0.930**	MIN = 0.928	**AVG = 0.930**
MAX = 0.933	STD = 0.001	MAX = 0.932	STD = 0.001
F1 Score			
MIN = 0.928	**AVG = 0.930**		
MAX = 0.932	STD = 0.001		

Table 1: Logistic Regression Train/Test Outputs

Decision Tree

Decision Tree (DT) : TRAIN RESULTS			
Train Accuracy		**Train ROC AUC**	
MIN = 0.951	**AVG = 0.952**	MIN = 0.949	**AVG = 0.952**
MAX = 0.954	STD = 0.001	MAX = 0.955	STD = 0.002
Train Precision		**Train Recall**	
MIN = 0.952	**AVG = 0.952**	MIN = 0.951	**AVG = 0.952**
MAX = 0.954	STD = 0.001	MAX = 0.954	STD = 0.001
F1 Score			
MIN = 0.951	**AVG = 0.952**		
MAX = 0.954	STD = 0.001		
Decision Tree (DT) : TEST RESULTS			

Test Accuracy		**Test ROC AUC**	
MIN = 0.949	**AVG = 0.952**	MIN = 0.949	**AVG = 0.952**
MAX = 0.954	STD = 0.001	MAX = 0.955	STD = 0.001
Test Precision		**Test Recall**	
MIN = 0.949	**AVG = 0.952**	MIN = 0.949	**AVG = 0.952**
MAX = 0.954	STD = 0.001	MAX = 0.954	STD = 0.001
F1 Score			
MIN = 0.949	**AVG = 0.952**		
MAX = 0.954	STD = 0.001		

Table 2: Decision Tree Train/Test Outputs

Model Runtime

Runtime Results: DT/LR			
**All model runtimes reported individually in 'AllResultsFinal' (i.e., raw Spark output).*			
DT: Runtime (Sec)		**LR: Runtime (Sec)**	
MIN = 4.864	**AVG = 5.055**	MIN = 5.307	**AVG = 5.929**
MAX = 5.744	STD = 0.240	MAX = 9.238	STD = 1.112

Table 3: Decision Tree/Logistic Regression Runtime Summary

(1.d) Model Discussion

After running the models, we performed additional post-processing on the Spark output in Jupyter Lab for a more rigorous analysis of these results. Upon closer inspection, we found that across all standard classification metrics (accuracy, precision, recall, F1-score, and ROC AUC), consistent and statistically significant differences were observed between DT and LR classifiers *(Table 4 – Table 8)*.

ACCURACY (Paired t-Test / Wilcoxon Test)				**Significance**	
Dataset	**DT**	**LR**	**Paired t-Test**	**p-value**	**Wilcoxon p-value**
TRAIN	μ = 0.9523	μ = 0.9301	63.9096	***0.000**	***0.020**
	σ = 0.0007	σ = 0.0010			
TEST	μ = 0.9519	μ = 0.9297	33.9416	***0.000**	***0.020**
	σ = 0.0015	σ = 0.0016			

Table 4: Paired Statistical Comparison of Model Accuracy Across Seeds

PRECISION (Paired t-Test / Wilcoxon Test)				**Significance**	
Dataset	**DT**	**LR**	**Paired t-Test**	**p-value**	**Wilcoxon p-value**
TRAIN	μ = 0.9524	μ = 0.9307	62.7084	***0.000**	***0.020**
	σ = 0.0007	σ = 0.0010			
TEST	μ = 0.9520	μ = 0.9304	32.9664	***0.000**	***0.020**
	σ = 0.0015	σ = 0.0015			

Table 5: Paired Statistical Comparison of Model Precision Across Seeds

RECALL (Paired t-Test / Wilcoxon Test)				**Significance**	
Dataset	**DT**	**LR**	**Paired t-Test**	**p-value**	**Wilcoxon p-value**
TRAIN	μ = 0.9523	μ = 0.9301	63.9096	***0.000**	***0.020**
	σ = 0.0007	σ = 0.0010			
TEST	μ = 0.9519	μ = 0.9297	33.9416	***0.000**	***0.020**
	σ = 0.0015	σ = 0.0016			

Table 6: Paired Statistical Comparison of Model Recall Across Seeds

F1 SCORE (Paired t-Test / Wilcoxon Test)				**Significance**	
Dataset	**DT**	**LR**	**Paired t-Test**	**p-value**	**Wilcoxon p-value**
TRAIN	μ = 0.9523	μ = 0.9301	63.9096	***0.000**	***0.020**
	σ = 0.0007	σ = 0.0010			
TEST	μ = 0.9519	μ = 0.9297	33.8797	***0.000**	***0.020**
	σ = 0.0015	σ = 0.0016			

Table 7: Paired Statistical Comparison of Model F1 Score Across Seeds

ROC AUC (Paired t-Test / Wilcoxon Test)				**Significance**	
Dataset	**DT**	**LR**	**Paired t-Test**	**p-value**	**Wilcoxon p-value**
TRAIN	μ = 0.9521	**μ = 0.9806**	-49.8908	***0.000**	***0.020**
	σ = 0.0020	σ = 0.0003			
TEST	μ = 0.9518	**μ = 0.9806**	-45.3142	***0.000**	***0.020**
	σ = 0.0015	σ = 0.0008			

Table 8: Paired Statistical Comparison of Model ROC AUC Across Seeds

MEAN Z-SCORES (TEST SET ONLY)				**OVERALL MEAN Z-SCORE PER MODEL**	
Metric	**DT**	**LR**		**DT**	**LR**
ACCURACY	0.9913	-0.9913		**0.5937**	**-0.5937**
PRECISION	0.9914	-0.9914			
RECALL	0.9913	-0.9913			
F1 SCORE	0.9913	-0.9913			
ROC AUC	**-0.9968**	**0.9968**			

Table 9: Mean Z-Scores by Metric and Model and Overall Mean Z-Score per Model

The first critical observation was that all metric differences were statistically significant, as indicated by the extremely low p-values of the paired t-tests (< 0.000001). Supplementary Wilcoxon signed-rank tests corroborated this (p = 0.0020 across all metrics), further reinforcing the result's robustness even under non-parametric assumptions.

We also noted that DT outperformed LR in all metrics except for ROC AUC, in which LR exhibited a substantially higher average (0.9806 vs. 0.9518). However, this was expected behaviour because the probabilistic nature of LR models makes them well-suited to metrics like ROC AUC, which assesses a ranking of predicted probabilities (as opposed to raw accuracy).

As Murphy (2012) explains, *"ROC curves and the AUC metric are commonly used to evaluate classifiers that output scores or probabilities. Probabilistic models, such as logistic regression, are particularly suitable for this kind of evaluation" (Murphy, 2012)*. The radar charts for train/test results (which are more descriptive than inferential) **also emphasised DT's clear ability to outperform LR on four out of the five metrics** *(Figure 1)*.

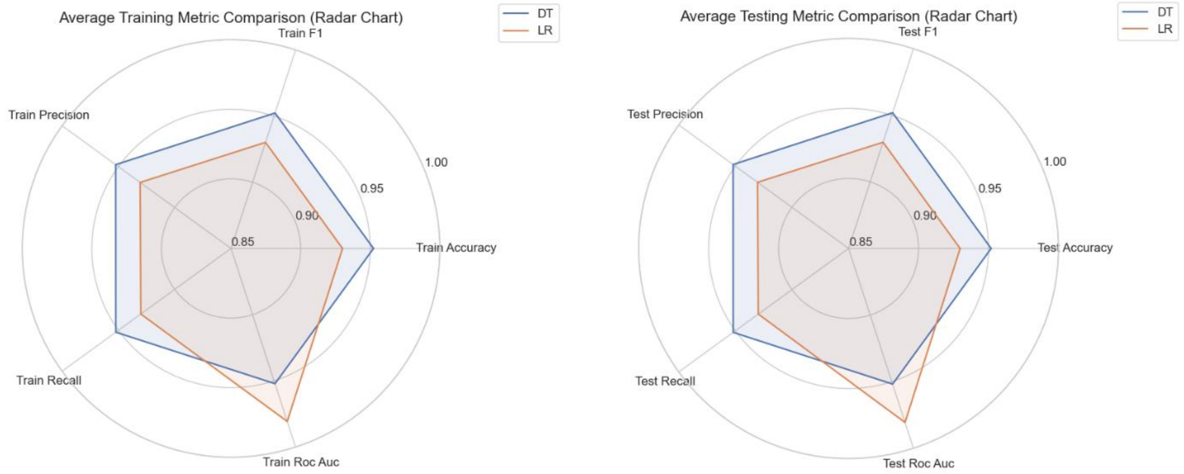


Figure 1: Train (Left) and Test (Right) Average Metric Comparison Radar Charts (Blue=DT; Orange=LR).

Next, we noted that both models exhibited low standard deviations across all evaluation metrics, indicating stable and consistent performance across different random train/test splits and seed initialisations. This suggests that model evaluations are robust and not overly sensitive to sampling variability. Based on our discussion so far, it is also perhaps expected to find that DT showed slightly greater consistency in accuracy-based metrics, while LR demonstrated more stable probabilistic ranking (i.e., lower variance in ROC AUC). We also visually confirmed the variation between metrics and model runs using box plots and z-score violin plots *(Appendix Figures A-B and Figure C, respectively)*.

Z-score normalisation allows all metrics to be interpreted on a common, unitless scale relative to their distribution. Hence, in our final post-processing step we applied Z-score normalisation to standardise the metric values across models and evaluation seeds. This ensured that differences in metric scales (e.g., accuracy vs. ROC AUC) did not bias our comparisons. This approach was particularly useful in this experiment because it enabled us to directly compare model performance across several metrics whilst communicating models' overall and relative performance consistencies (as opposed to their mere absolute scores).

The mean Z-scores per metric *(Table 9)* demonstrated that DT definitively performed above the mean across accuracy, precision, recall, and F1 (Z ≈ +0.99), which indicates strong and stable classification performance. Relative to DT, LR underperformed on these metrics (Z ≈ -0.99) but showed a high Z-score on ROC AUC (+0.9968). This divergence shows that DT is better suited to discrete classification tasks, while LR produces well-calibrated probability estimates that are favored by ranking-based metrics like ROC AUC (as previously discussed).

Finally, the overall mean Z-score across metrics was +0.5937 for DT and -0.5937 for LR, reinforcing the conclusion that **DT delivered more consistent and balanced performance in this experiment on the KDD dataset.**

As for the observation that the individual metric and overall Z-scores for DT and LR are exactly inverse, this is because Z-score normalisation was applied within each metric across both models, and the (metric) values themselves are symmetrical and balanced across seeds. This numeric symmetry *(Table 9)* is simply a byproduct of normalising two groups whose sizes and variances are equal (or very close), and causes the Z-scores to distribute in mirror image across DT and LR. Mathematically, Z-score normalisation is defined as:

$$z_i = \frac{x_i - \mu}{\sigma} \qquad \textbf{(1)}$$

In which x_i is the raw score, μ is the mean of all scores in that group, and σ is the standard deviation. For each metric, we pool DT and LR values together. If DT scores are consistently higher and LR scores are consistently lower (or vice versa) and both groups are equally sized with mirrored spread, then DT values will fall equally above the mean (+Z) while LR values fall equally below the mean (-Z), and the mean Z-score of DT will exactly negate the mean Z-score of LR (as in our example).

That is to say, the mean Z-score nomalisation values being mirrored across DT and LR is not an error. Rather, it informs us that each metric has exactly two groups (DT/LR) of equal size and that DT and LR performances are highly separable and directionally consistent across seeds for each metric (and overall).

(1.e) Conclusion

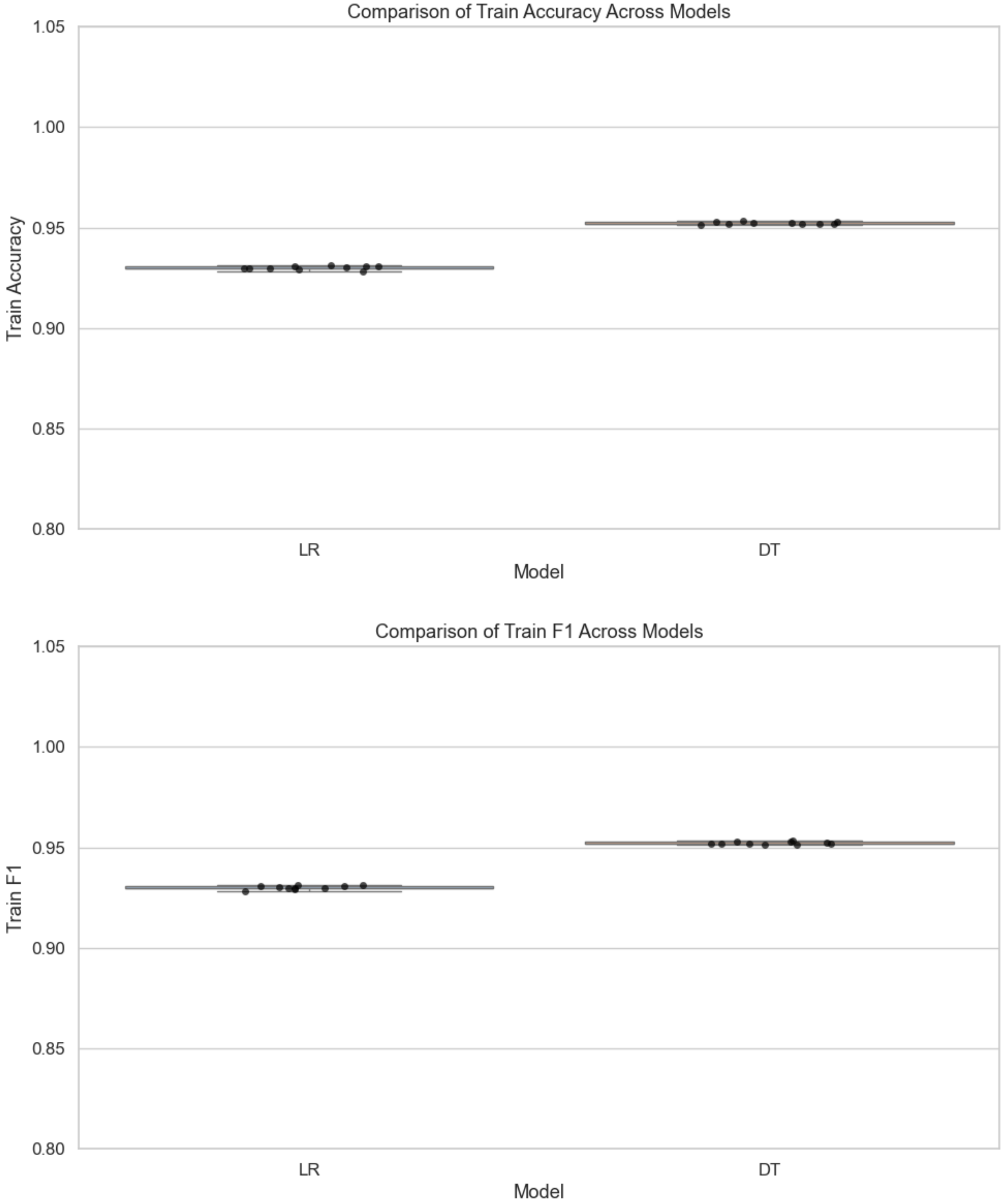
Overall, DT consistently outperformed LR across standard classification metrics (accuracy, precision, recall, F1), both in raw scores and in z-score normalised comparisons. In contrast, LR excelled on ROC AUC [reflective of its strength in probabilistic ranking] but performed worse than DT on all other evaluation metrics. Low standard deviations across these metrics confirmed the stability of both classifiers over repeated trials and statistical significance testing (paired t-Tests) further reinforced the reliability of these differences across all seeds.

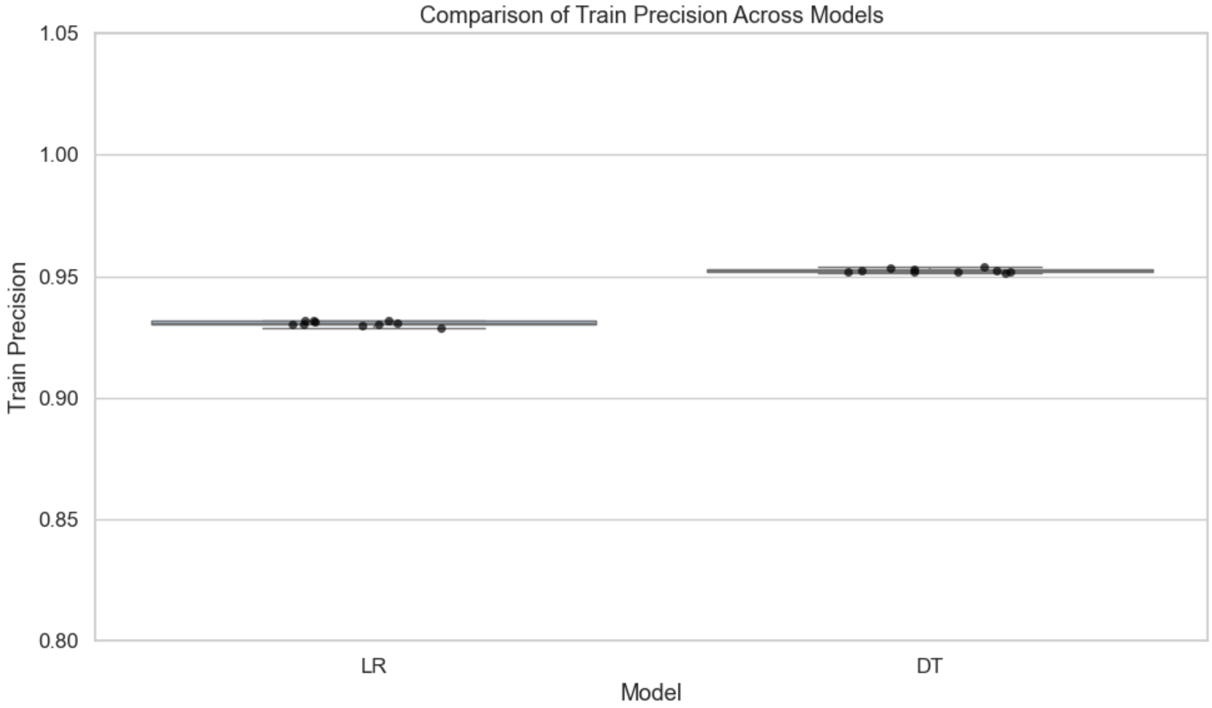
The Z-score analysis demonstrated that DT consistently performed above-average on discrete metrics, while LR led on ROC AUC. The runtime analysis *(Table 3)* also showed that DT executed more efficiently on average. These results suggest that DT is very likely to be the more effective choice for classification tasks prioritising discrete label accuracy on categorically-dense datasets, while LR is more suited to scenarios that require calibrated probability estimates.

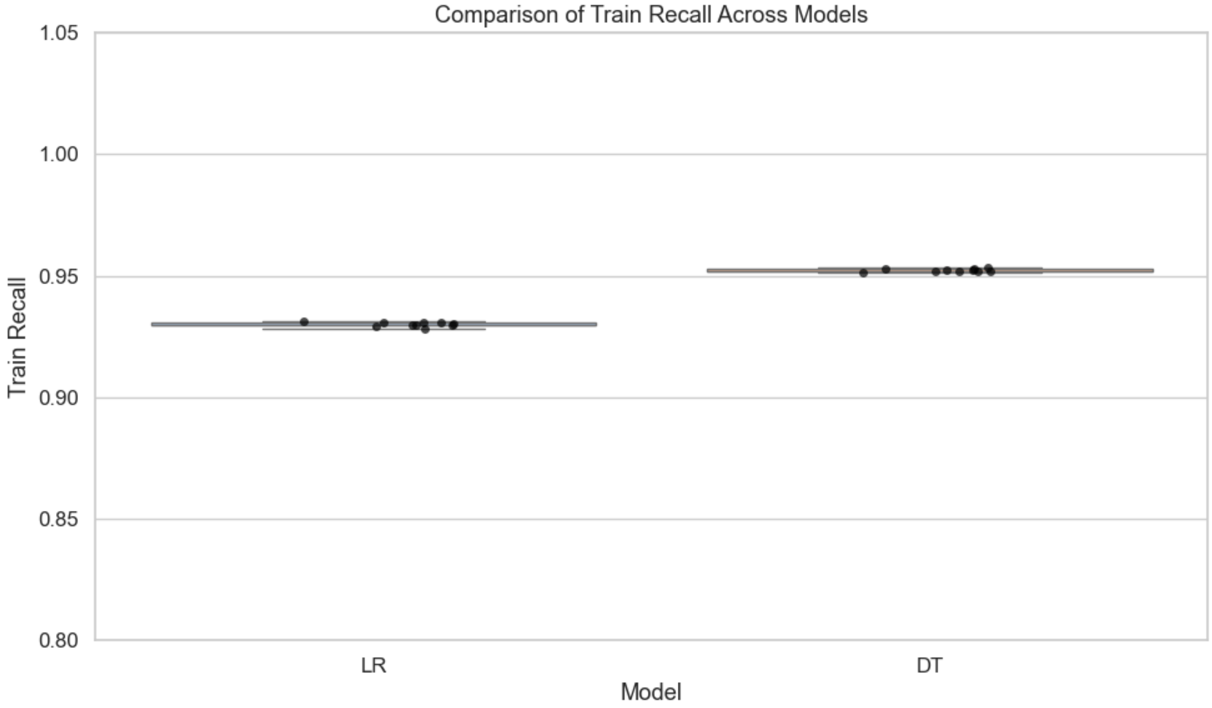
Finally, this experiment effectively applied interdisciplinary principles from data science, statistics, and machine learning (mixed disciplines) to implement the models in a distributed (Apache Spark) environment, evaluate the models, and draw robust comparisons between them.

Figure Appendix

***Figure A**: DT/LR Training Results Over 10 Seeds*




Comparison of Train Precision Across Models
Train Precision
1.05
1.00
0.95
0.90
0.85
0.80
LR
DT
Model


Comparison of Train Recall Across Models
Train Recall
1.05
1.00
0.95
0.90
0.85
0.80
LR
DT
Model

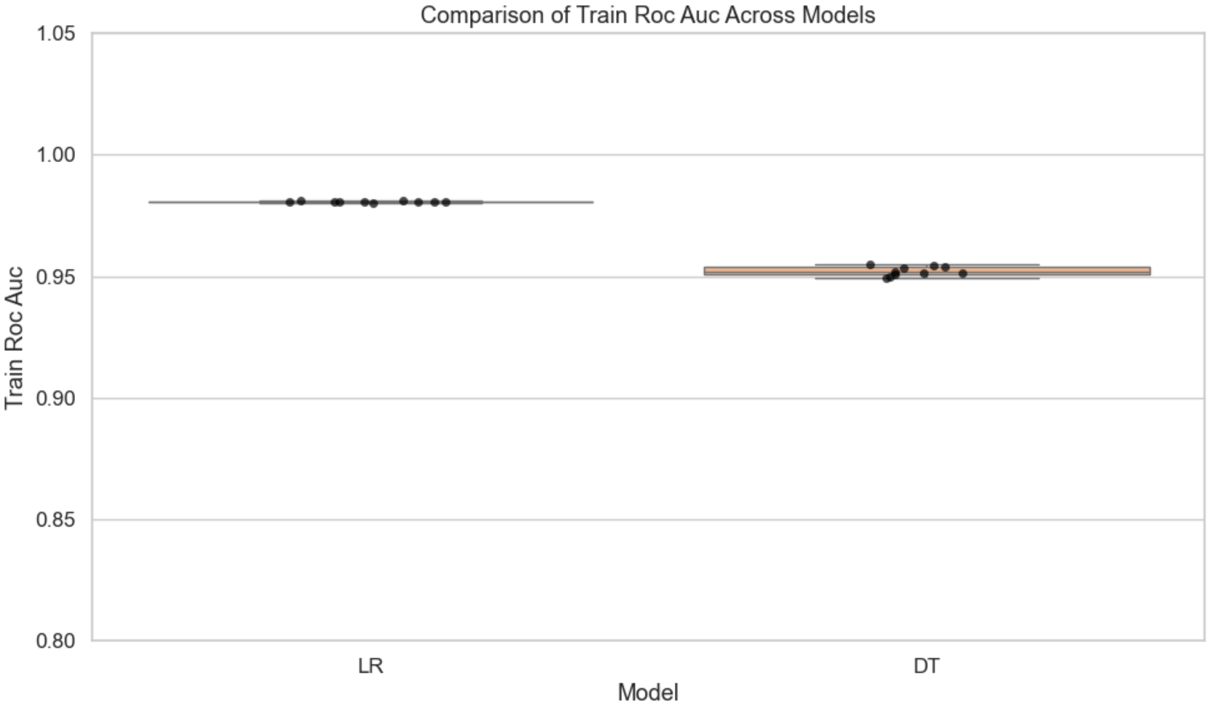
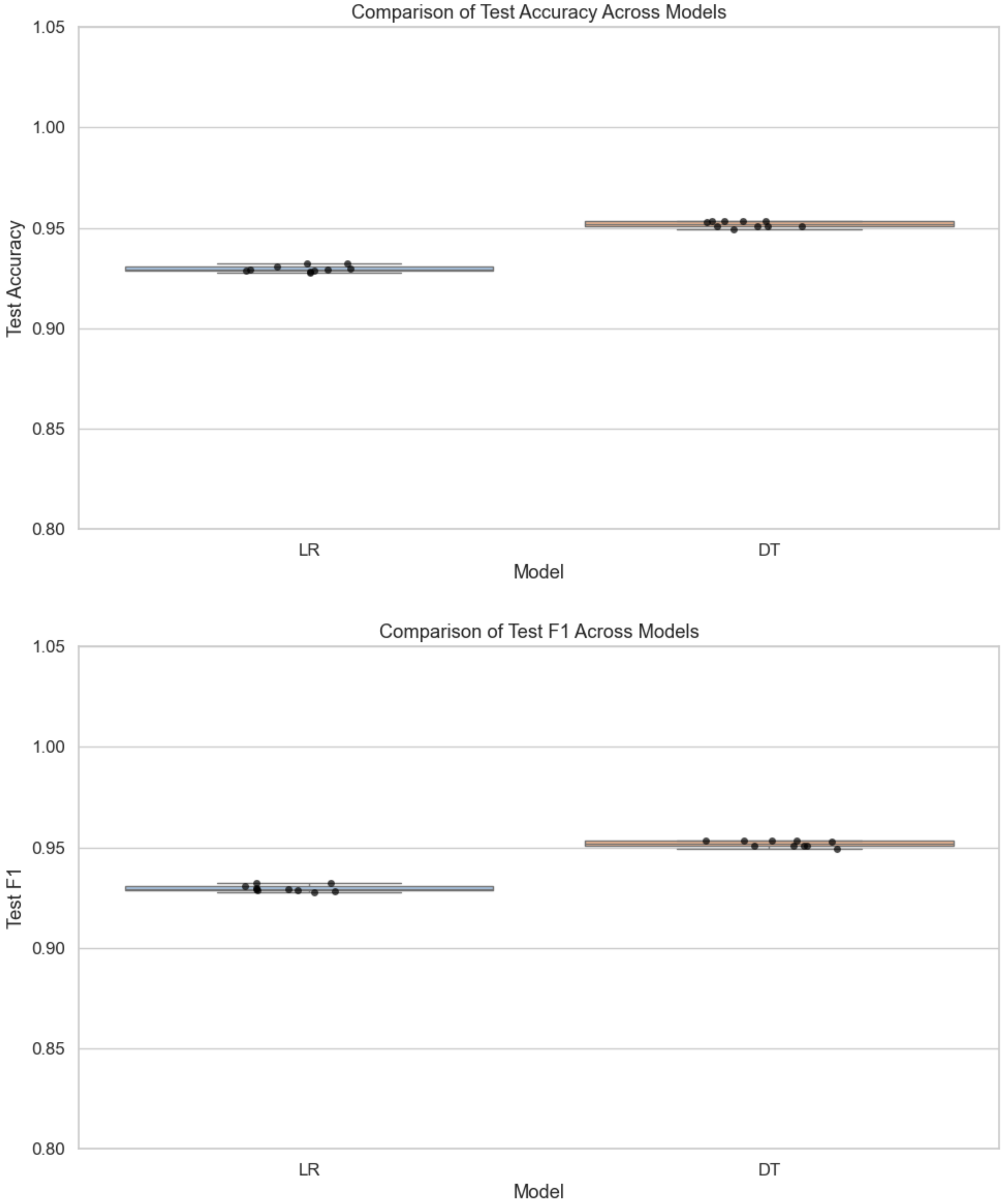
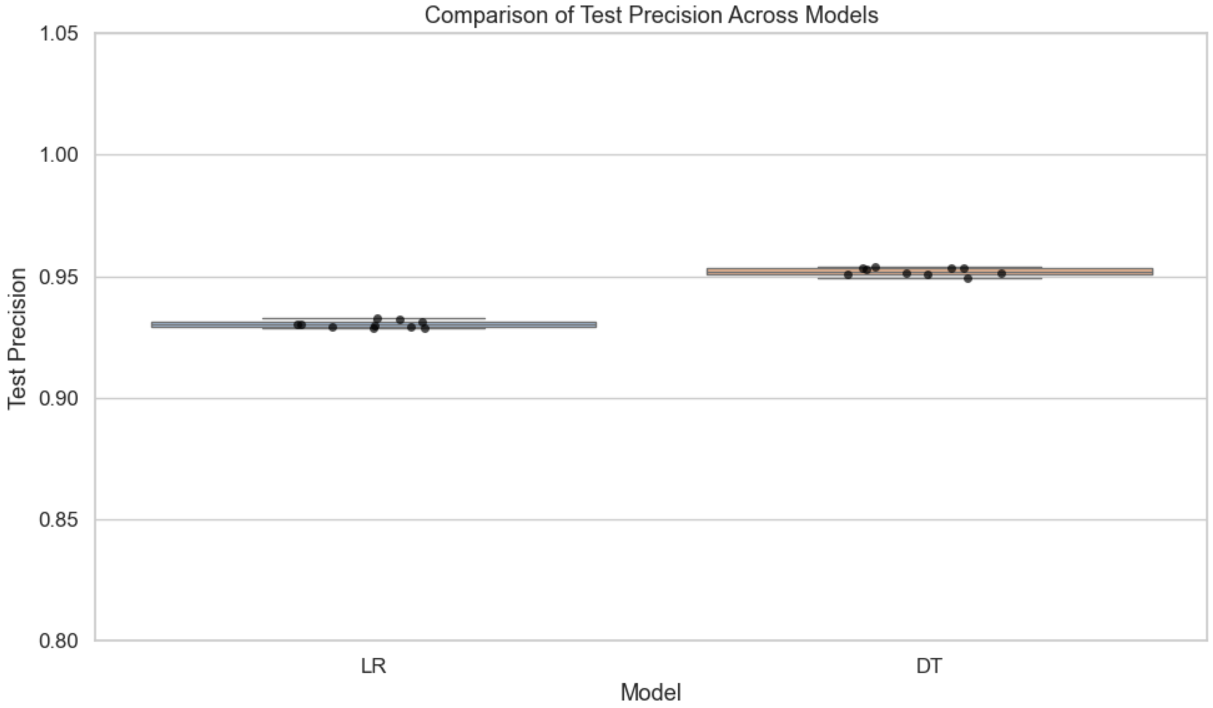
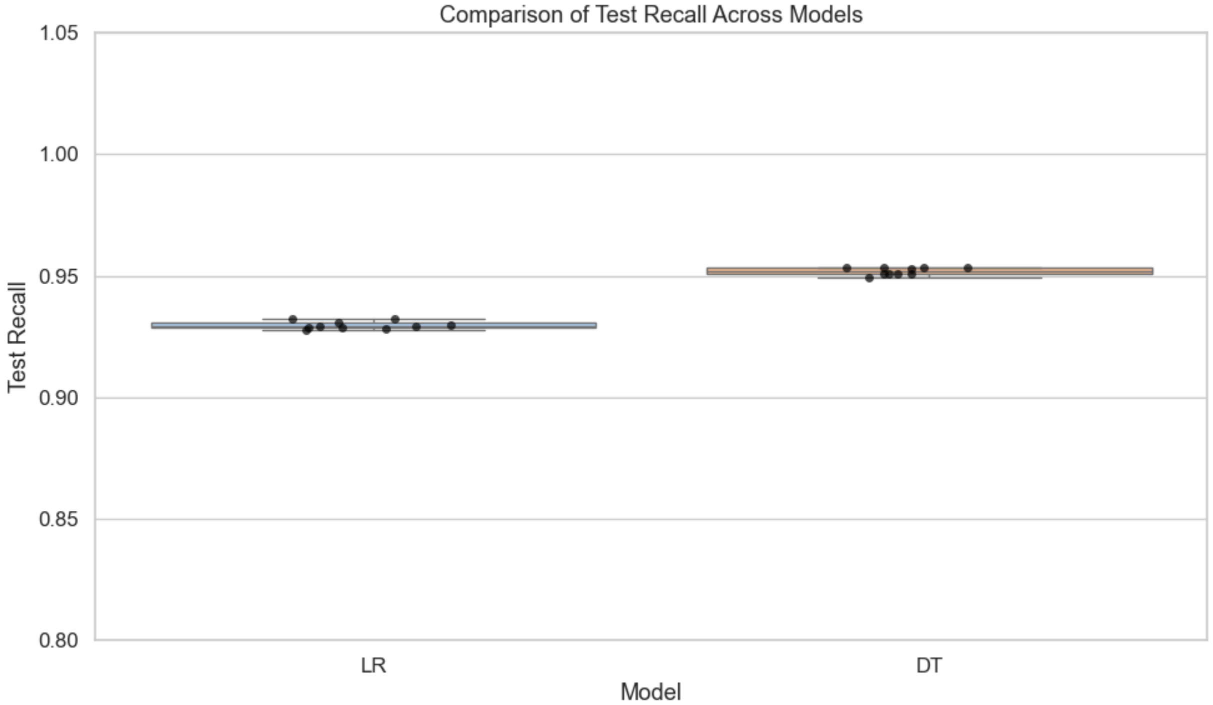

Comparison of Train Roc Auc Across Models
1.05
1.00
0.95
0.90
0.85
0.80
Train Roc Auc
LR
DT
Model

Figure B: DT/LR Testing Results Over 10 Seeds




Comparison of Test Precision Across Models
Test Precision
1.05
1.00
0.95
0.90
0.85
0.80
LR
DT
Model


Comparison of Test Recall Across Models
Test Recall
1.05
1.00
0.95
0.90
0.85
0.80
LR
DT
Model

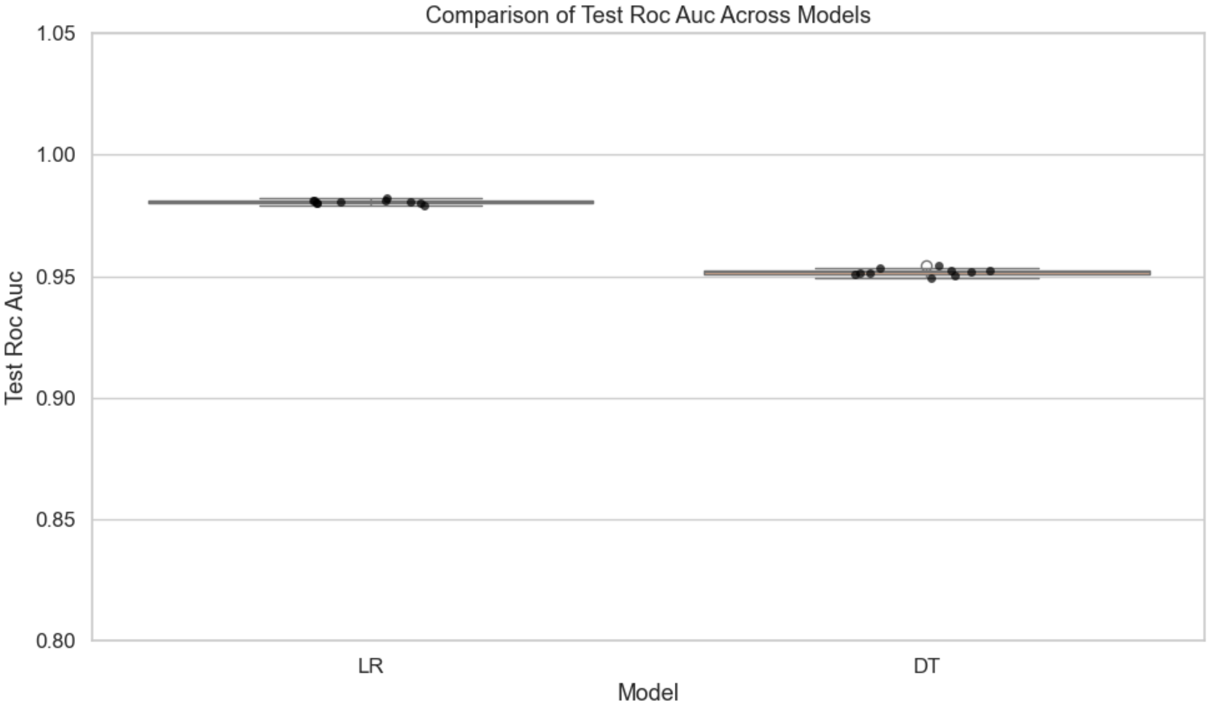
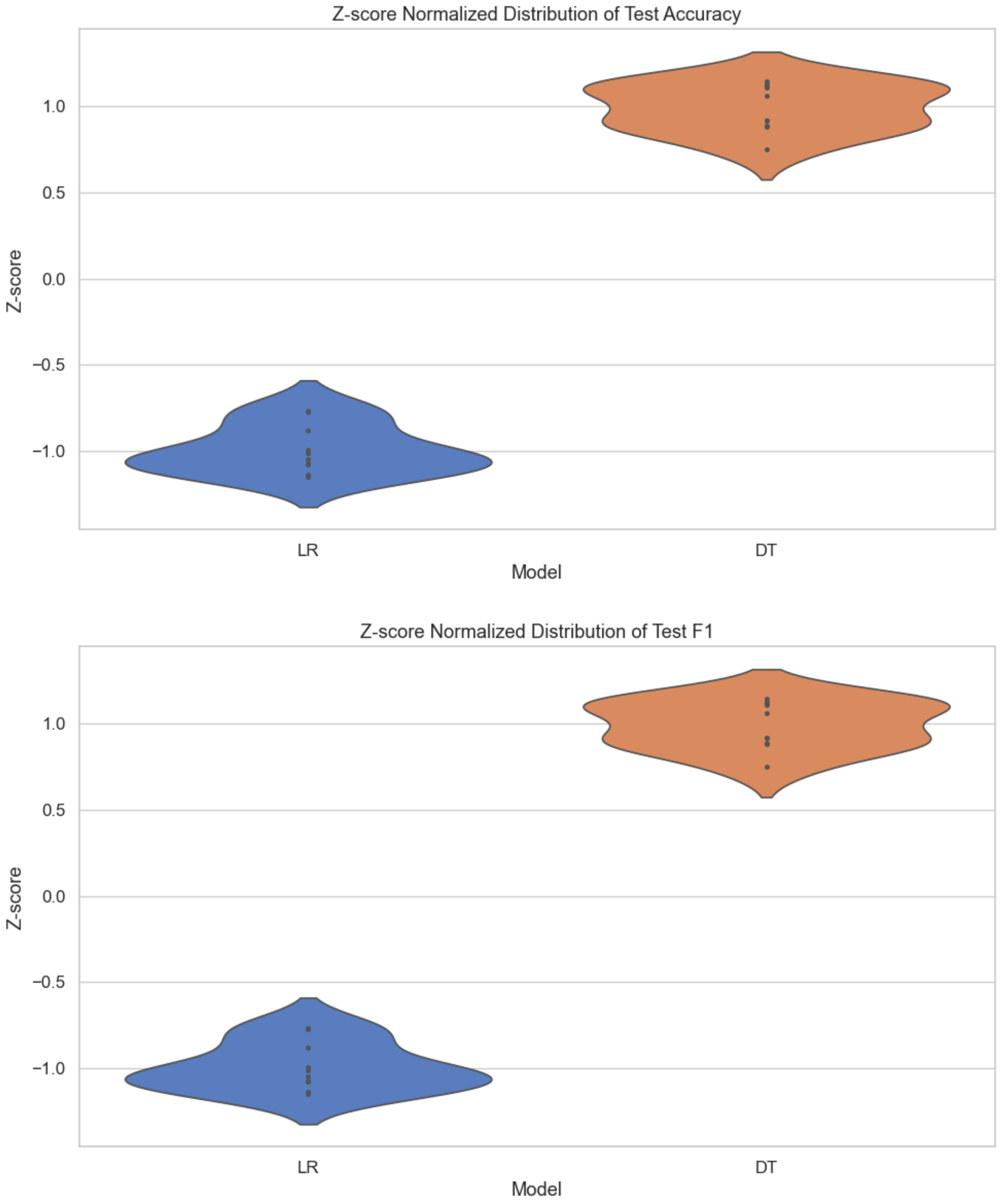
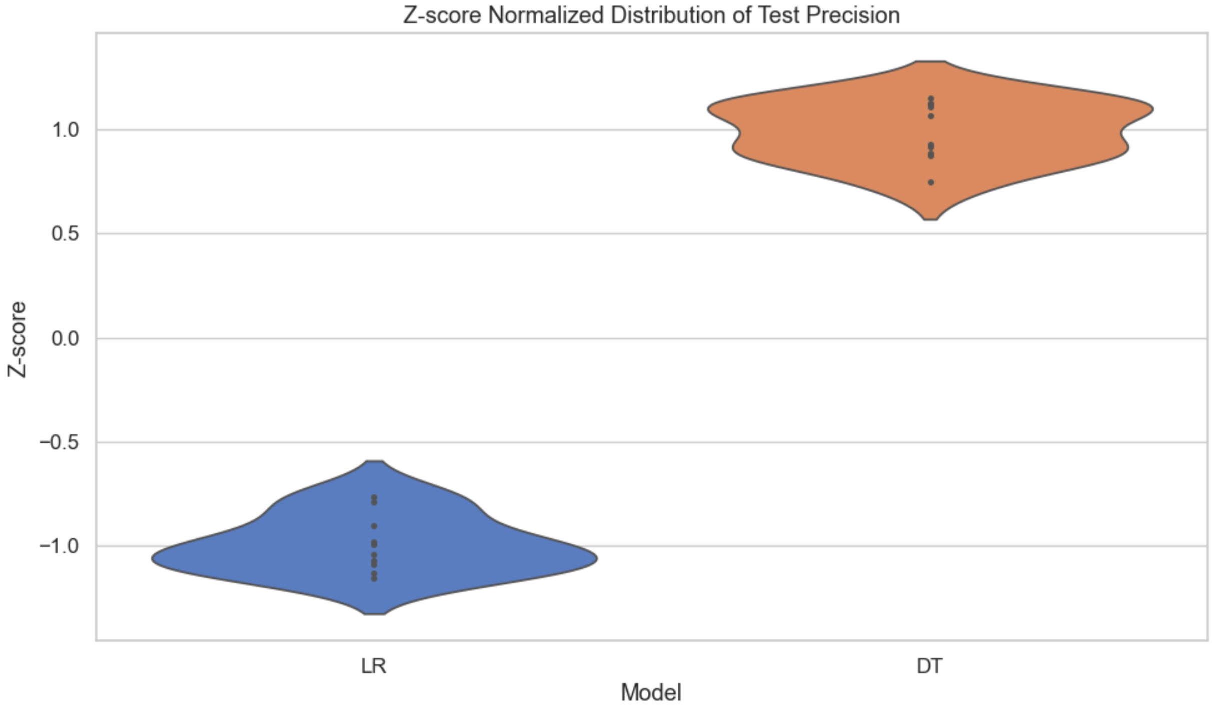
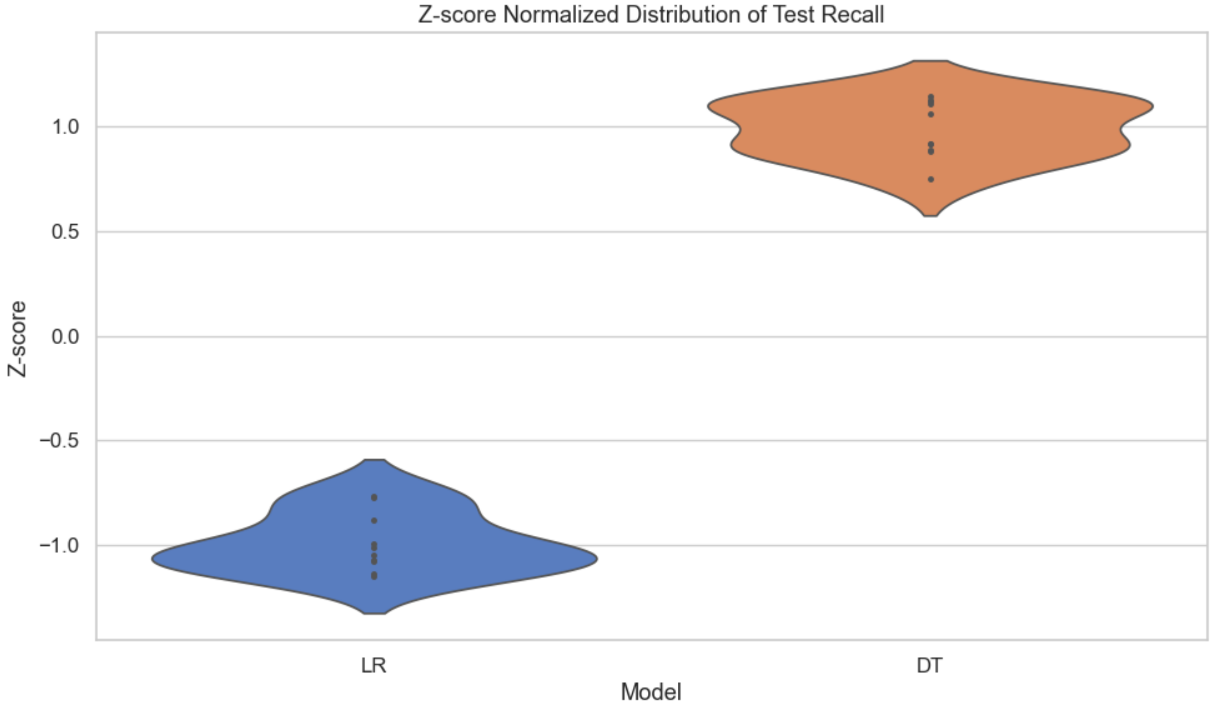
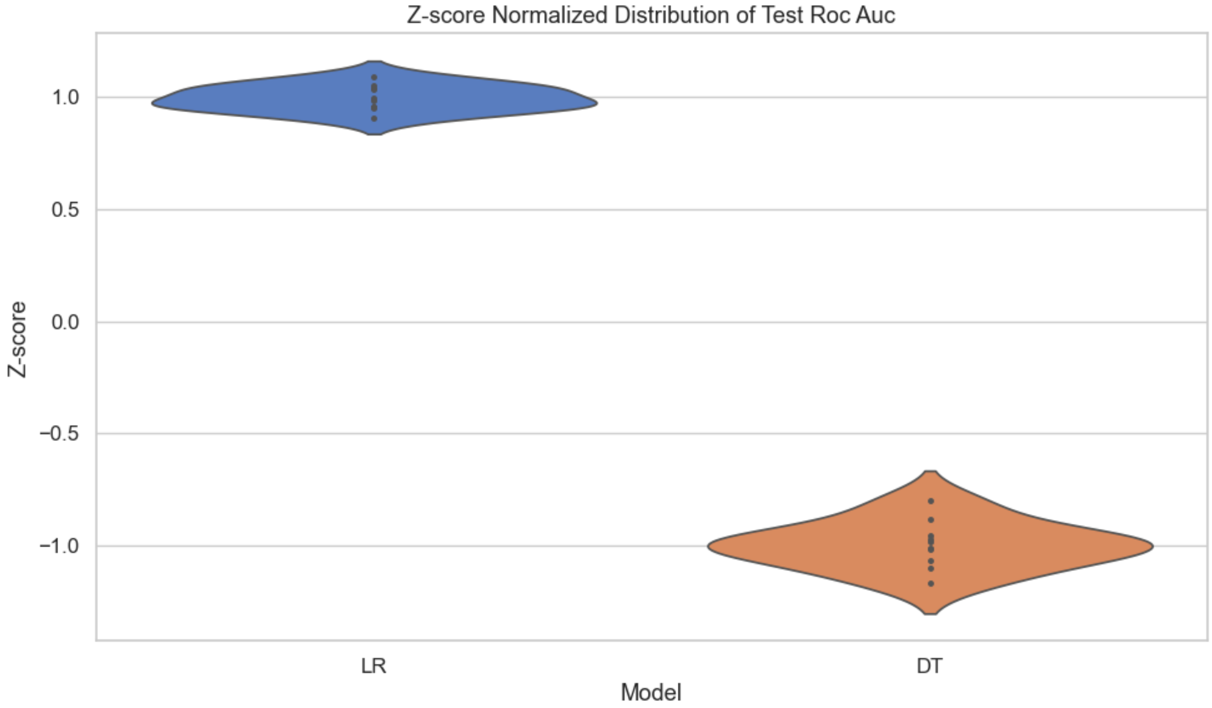

Comparison of Test Roc Auc Across Models
Test Roc Auc
1.05
1.00
0.95
0.90
0.85
0.80
LR
DT
Model

Figure C: Z-score (Test Results) Over 10 Seeds




Z-score Normalized Distribution of Test Precision
Z-score
1.0
0.5
0.0
−0.5
−1.0
LR
DT
Model


Z-score Normalized Distribution of Test Recall
Z-score
1.0
0.5
0.0
−0.5
−1.0
LR
DT
Model



References

Murphy, K. P. (2012). Logistic Regression. In *Machine Learning: A Probabilistic Perspective* (pp. 245–271). MIT Press.

UCI KDD Archive. (1999). KDD Cup 1999 Data. In *Information and Computer Science*. University of California. https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html