UNIVERSITÀ
DEGLI STUDI
DELL'AQUILA

disim

Introduction to Statistical Learning

Universita' degli studi dell'Aquila

Department of Information Engineering and Computer Science and Mathematics

Muhammad Ahmed - 287068

Masters Applied Data Science

Contents

1 Introduction

This report presents a comprehensive analysis of two statistical learning problems using real and simulated datasets. The first exercise focuses on modeling flight delay data, specifically flights from Washington DC to New York City in January 2004. The primary objective is to build a classification model to predict whether a flight is delayed by more than 15 minutes, using various temporal, operational, and environmental features.

The process begins with exploratory data analysis (EDA) to uncover key patterns and correlations, followed by feature engineering and selection using correlation thresholds, subset selection, and LASSO regression. Several classification models—Logistic Regression, LDA, Decision Trees, Naive Bayes, Bagging, and Random Forests—are trained and evaluated using accuracy, sensitivity, specificity, and AUC metrics. The logistic regression model is ultimately chosen for its balanced performance.

The second exercise involves synthetic data generation for regression analysis. Various linear models—LASSO, Ridge, Principal Component Regression (PCR), and Forward Subset Selection—are compared under different scenarios using Monte Carlo simulations. The performance is assessed using test Mean Squared Error (MSE), with the goal of identifying the most robust model under different data complexities.

All analyses were conducted in Python using statistical libraries. The entire workflow emphasizes robust modeling, variable reduction, and the impact of sample size and sparsity on model performance.

2 Exercise 1

We will start by exploring the dataset. The flightdelays.csv dataset tracks flights from Washington DC to NYC in January 2004, with a focus on predicting delays over 15 minutes.

```
RangeIndex:  1981 entries, 0 to 1980
Data columns (total 79 columns)
```

Table 1 shows that our dataset has a total of 1981 entries with 79 columns. Lets analyse these columns

Column	Non-Null	Count	Dtype
var600	not-null	1981	int64
var630	not-null	1981	int64
...	...	...	...
Y	not-null	1981	int64

Table 1: Dataset Information

Columns	Description	Examples	DataType	Count
Scheduled Time	Binary indicator for scheduled departure hour	var600 - var2130	Binary	59
Carrier	Airline codes: CO, DH, DL, MQ, OH, RU, UA, US	varCO, varDH, ..., varUS	Binary	8
Departure Airport	Departure from: DCA, IAD, BWI	varDCA, varBWI, varIAD	Binary	3
Arrival Airport	Arrival to: JFK, LGA, EWR	varJFK, varEWR, varLGA	Binary	3
Day of the Week	1 for Sunday and Monday, 0 otherwise	dayweek	Binary	1
Weather	1 for bad, 0 for good	weather	Binary	1
Actual Departure Time	True time of takeoff (e.g., 1030 = 10:30 am)	deptime	Integer	1
Flight Distance	Distance covered in flight	distance	Integer	1
Flight Number	Flight identifier	flightnumber	Integer	1
Delayed	1 if delayed >15 mins, 0 otherwise	Y	Binary	1

Table 2: Key Feature Groups

The Table 2 outlines the key variables such as flight status (Delayed), departure and arrival airports, airline carriers, scheduled departure time, etc. These features are used to analyze factors contributing to delays.

2.1 Exploratory Data Analysis

The initial step is to do a thorough analysis of the dataset.

The exploratory analysis in Figure 1 revealed that out of 1,981 total observations, only 384

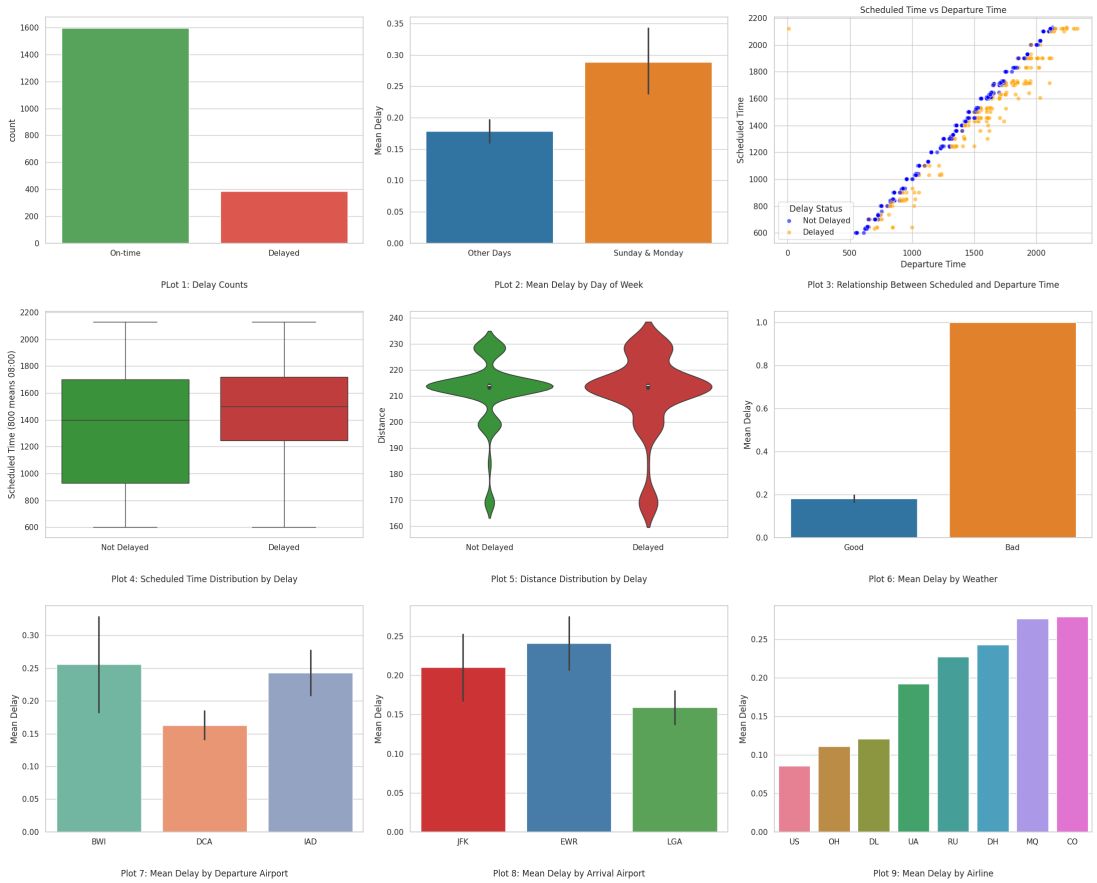

Figure 1: Exploratory Data Analysis plots

flights (19.4%) experienced delays, while the majority (1,597 flights) departed on time. Delays were more pronounced on Sundays and Mondays, suggesting a potential link to higher travel volumes or operational bottlenecks at the start and end of the week. A strong correlation between scheduled and actual departure times allowed us to simplify the model by retaining only scheduled time, avoiding redundancy (Section 2.2). Interestingly, flights scheduled around 1:00 PM and 5:00 PM showed a higher propensity for delays, likely due to peak traffic periods. Violin plots highlighted that delays affected both short and long-haul flights, but those exceeding 235 km were almost guaranteed to be delayed, possibly due to stricter scheduling constraints or longer turnaround times. Weather emerged as a key factor, with adverse conditions significantly increasing delay frequencies. Among airlines, Continental (varCO) had the worst on-time performance, whereas US Airways (varUS) operated most efficiently. These insights underscore the multifaceted nature of flight delays, influenced by temporal, operational, and environmental factors.

2.2 Variable Selection

The dataset includes several explanatory variables, such as categorical features related to flight origin, destination, and carriers, as well as numerical variables like distance and flight number, which together total 20 variables. Additionally, I applied inverse one-hot encoding to the scheduled time variable, creating a new variable named "Scheduled Time," which brings the total number of features to 21.

After splitting the data into 70% training and 30% testing sets, I began by removing highly correlated variables. Based on an initial correlation analysis figure 2, I identified pairs with high correlations, such as departure time and scheduled time (correlation of 0.99), as well as other pairs with correlations exceeding 0.8 (e.g., varDH with varIAD, varBWI, flight number, etc.), and similarly, variables with correlations above 0.5. I removed these redundant features to reduce multicollinearity and mitigate potential overfitting.

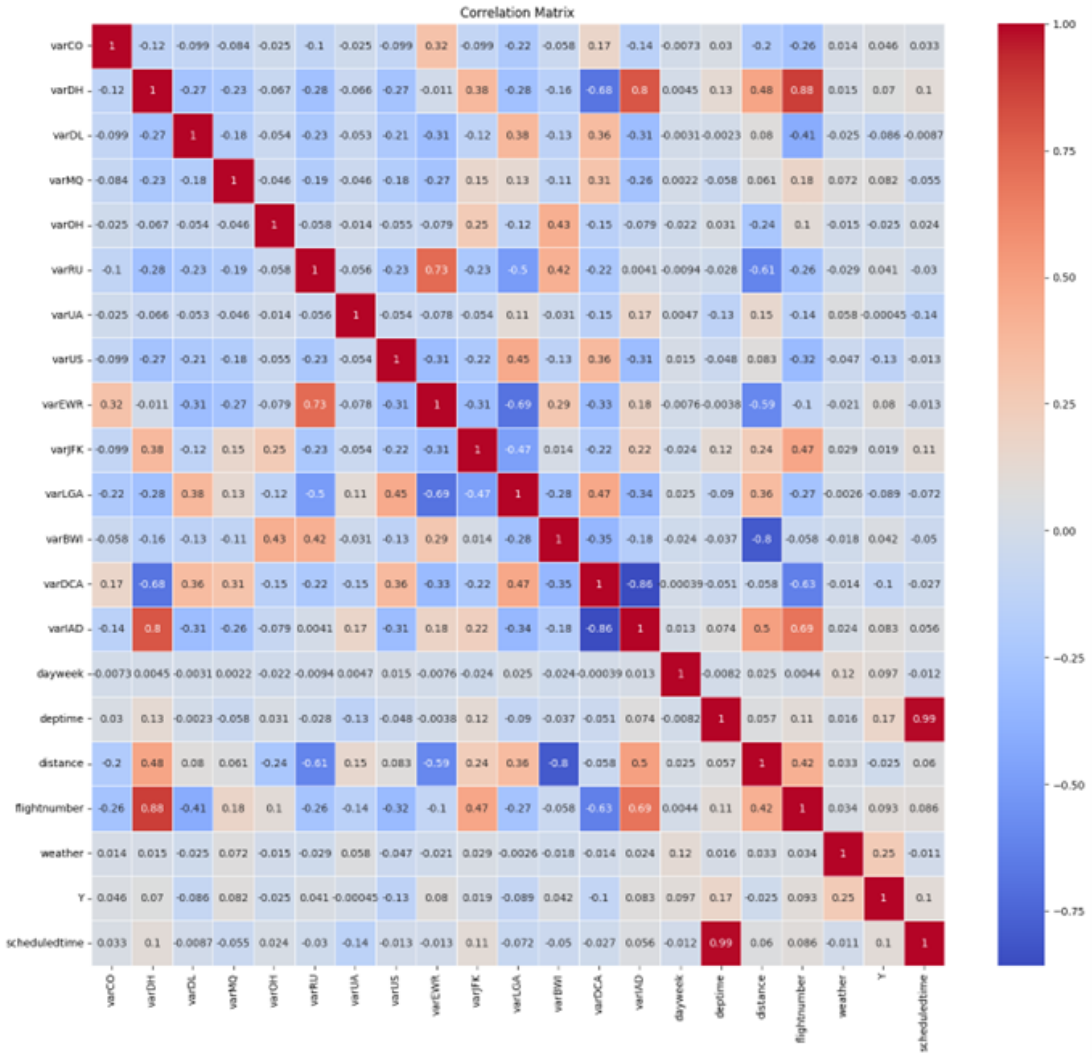

Figure 2: Correlation Matrix 21 variables in my dataset

For feature selection, I tested several approaches. Working with the training data, I fitted a logistic regression model to identify important variables based on their p-values. I experimented with different correlation thresholds (0.5, 0.8, and 0.9) to identify which features were most relevant and non-redundant. This process provided useful insights, as some variables with low or NaN p-values were removed. Finally, for the 0.5+ threshold, I selected 10 features. For the 0.8+ threshold, 9 features were chosen based on their relevance and performance; similarly, for the 0.9+ threshold, 13 features were selected, as shown in Table 3.

Correlation Threshold	**Selected Features**
0.5 +	['varDL', 'varLGA', 'distance', 'varCO', 'varJFK', 'varUA', 'varMQ', 'varOH', 'varDCA', 'weather']
0.8 +	['varDL', 'varUA', 'varCO', 'varMQ', 'varOH', 'varDCA', 'varRU', 'distance', 'weather']
0.9 +	['varDL', 'varUA', 'varCO', 'varUS', 'varMQ', 'varOH', 'varRU', 'varDH', 'varLGA', 'varJFK', 'varEWR', 'distance', 'weather']

Table 3: Selected features based on correlation thresholds

Subsequently, I decided to apply subset selection to the features selected by each correlation threshold. The Forward and Backwards subset models yielded higher MSE values compared to the Best Subset Selection for each correlation threshold. To further refine the feature selection process, I employed LASSO (Least Absolute Shrinkage and Selection Operator), which automatically shrinks some coefficients to zero. After fitting the model on the training data, I selected the optimal lambda (regularization parameter) using cross-validation. LASSO selected different features for each threshold, resulting in a slightly higher MSE compared to the Best Subset model, as shown in Figure 3.

Hence, the lowest MSE recorded among all the Best Subset and LASSO models across all thresholds is from the Best Subset with a 0.5+ correlation threshold, which has an MSE of 0.1418 (highlighted) in Figure 3.

Finally the selected Model has equation 1:

$$Y \sim \text{varUA} + \text{varCO} + \text{varMQ} + \text{varOH} + \text{varDCA} + \text{distance} + \text{weather} \tag{1}$$

Correlation Threshold: **0.5+**

Best Subset Selected Features: distance, varCO, varUA, varMQ, varOH, varDCA, weather

Best Subset Test MSE: 0.1418

Best inverse lambda (C): 0.35938

LASSO Logistic Regression Test MSE: 0.1425

LASSO Selected Features: varLGA, distance, varCO, varJFK, varUA, varMQ, varOH, varDCA, weather

Correlation Threshold: **0.8+**

Best Subset Selected Features: varDL, varUA, varCO, varMQ, varOH, varDCA, distance, weather

Best Subset Test MSE: 0.1419

Best inverse lambda (C): 0.35938

LASSO Logistic Regression Test MSE: 0.1420

LASSO Selected Features: varUA, varCO, varMQ, varOH, varDCA, varRU, weather

Correlation Threshold: **0.9+**

Best Subset Selected Features: varDL, varUS, varMQ, varOH, varDH, distance, weather **Best Subset Test MSE:** 0.1422

Best inverse lambda (C): 0.35938

LASSO Logistic Regression Test MSE: 0.1433

LASSO Selected Features: varDL, varUA, varCO, varUS, varMQ, varOH, varDH, varJFK, varEWR, weather

Figure 3: Results on different Correlation Threshold levels using subset selection and LASSO

2.3 Model Selection

After performing logistic regression, I fit the training data to a Linear Discriminant Analysis (LDA) model. Both models produced nearly identical AUC scores (0.684), though their confusion matrices differed slightly. Notably, LDA resulted in a perfect specificity of 100% but had a very low sensitivity (6.09%), indicating it was heavily biased toward predicting the majority class (no delay) as shown in the Table 4.

I also trained a classification tree using DecisionTreeClassifier from scikit-learn in python. Unlike logistic regression and LDA, the decision tree demonstrated a higher ability to detect delays (sensitivity = 37.39%) but at the cost of reduced overall accuracy (69.58%) and specificity (77.29%).

Across these models, analysis revealed that most splits were dominated by a few variables, such as weather and scheduled distance. This made the models good at predicting no-delay outcomes (class 0), but poor at detecting actual delays (class 1), which were underrepresented in the data. The class imbalance (Pr(Y=0) = 0.81, Pr(Y=1) = 0.19) likely contributed to this, especially when using a threshold of 0.5 to classify observations.

To mitigate this, I experimented with ensemble methods, namely Bagging and Random Forests, which allow consideration of more variables at each split and may offer robustness against overfitting. The bagging model, using all 7 predictors and 100 base estimators, showed a slight improvement in delay detection (sensitivity = 13.91

I then trained two Random Forest models: one using $m = 3$ (i.e., $\sqrt{7} \approx 3$) and another using $m = 6$. Both models had the same confusion matrix as bagging, but slightly lower out-of-bag error during training, suggesting better generalization. However, none of the ensemble models surpassed the logistic model in AUC.

Additionally, I included Naive Bayes in my comparison. While its performance was very similar to LDA (same confusion matrix, identical specificity and sensitivity), it slightly outperformed all others in terms of AUC (0.686), making it the top scorer by that metric.

Below is a summary of the models and their evaluation metrics:

Model	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC
Naive Bayes	81.85	6.09	100.00	0.686
LDA	81.85	6.09	100.00	0.684
Logistic	78.66	11.30	94.79	0.684
Decision Tree	69.58	37.39	77.29	0.660
Bagging/RF (m =3 & 6)	77.14	13.91	92.29	0.657

Table 4: Summary of Performance Metrics

Despite having lower accuracy than LDA and Naive Bayes, I selected the logistic regression model as the final model due to its competitive AUC and balanced performance. It demonstrated a better trade-off between true positive and false positive rates. If maximizing the detection of delays was the sole priority, the decision tree or bagging methods would have been preferable due to their higher sensitivity. The ROC curves are represented on Figure 4.

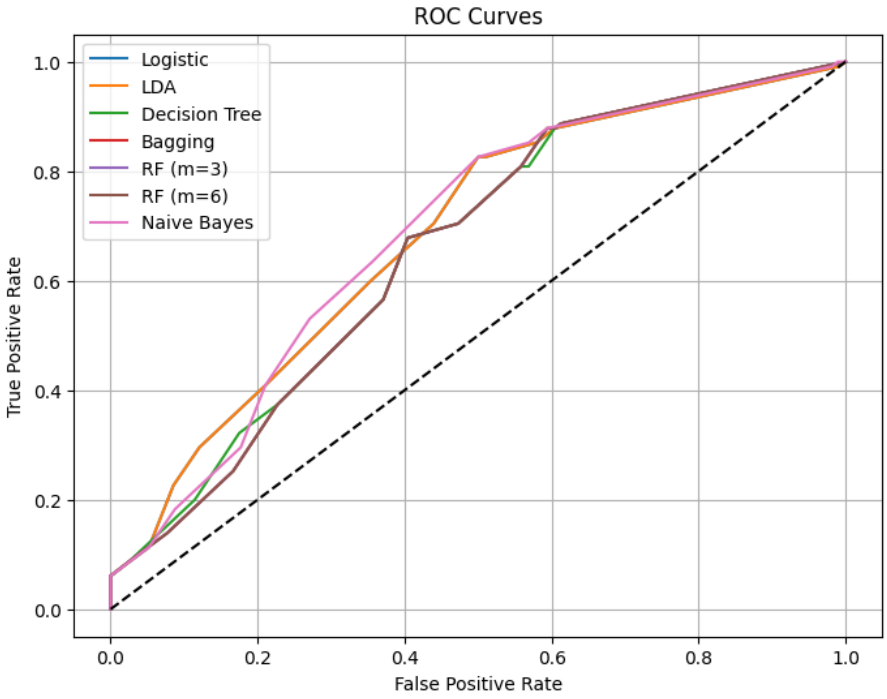

Figure 4: Plot of ROC curves

2.4 Prediction

Finally, I used the selected model to generate predictions on the test dataset with a classification threshold of 0.3, which helps to improve sensitivity for detecting delays in this imbalanced setting. The predictions were saved to `final_predictions_submission.csv` for evaluation.

3 Exercise 2

I randomly generated a data set with $p = 8$ features, $N = 100$ observations, and an associated quantitative response vector according to the model:

$$y_i = \sum_{j=1}^{p} \beta_j X_{ij} + \varepsilon_i$$

The matrix $\mathbf{X}$ had a normal distribution with mean $\mathbf{0}$ and variance Σ_X

To evaluate model performance under varying conditions, the analysis focused on the distribution of test Mean Squared Error (MSE) across different values of the sample size N and coefficient vector $\boldsymbol{\beta}$. The error term ε was assumed to follow a normal distribution with mean 0 and standard deviation 3. For each configuration, the dataset was randomly generated and partitioned into 70% training and 30% test data.

Four modeling approaches were assessed: LASSO, Ridge Regression, Principal Components Regression (PCR), and Forward Subset Selection. The best-performing model in each scenario was determined by the one that minimized test MSE. A total of six experimental settings were explored for each of the two cases of $\boldsymbol{\beta}$.

Case 1:

- $\mathbf{b} = (1, 1, 1, 1, 1, 1, 1, 1); \quad N = 100$
- $\mathbf{b} = (1, 1, 1, 1, 1, 1, 1, 1); \quad N = 500$
- $\mathbf{b} = (1, 1, 1, 1, 1, 1, 1, 1); \quad N = 1000$

Case 2:

- $\mathbf{b} = (1, 1, 1, 1, 0, 0, 0, 0); \quad N = 100$
- $\mathbf{b} = (1, 1, 1, 1, 0, 0, 0, 0); \quad N = 500$
- $\mathbf{b} = (1, 1, 1, 1, 0, 0, 0, 0); \quad N = 1000$

For each of the six instances, I conducted 200 Monte Carlo simulations to assess the performance of the four models. In each simulation, the training data was generated with feature correlation levels $\rho \in \{0, 0.5, 0.95\}$, allowing for evaluation under varying degrees of multicollinearity. The models—LASSO, Ridge Regression, Principal Components Regression (PCR), and Forward Subset Selection—were then fit to the training data, and test MSEs were computed using the corresponding test sets. The results were visualized using box plots, which

illustrate the distribution of test MSEs across the 200 simulations for each model under every combination of N, $\boldsymbol{\beta}$, and ρ.

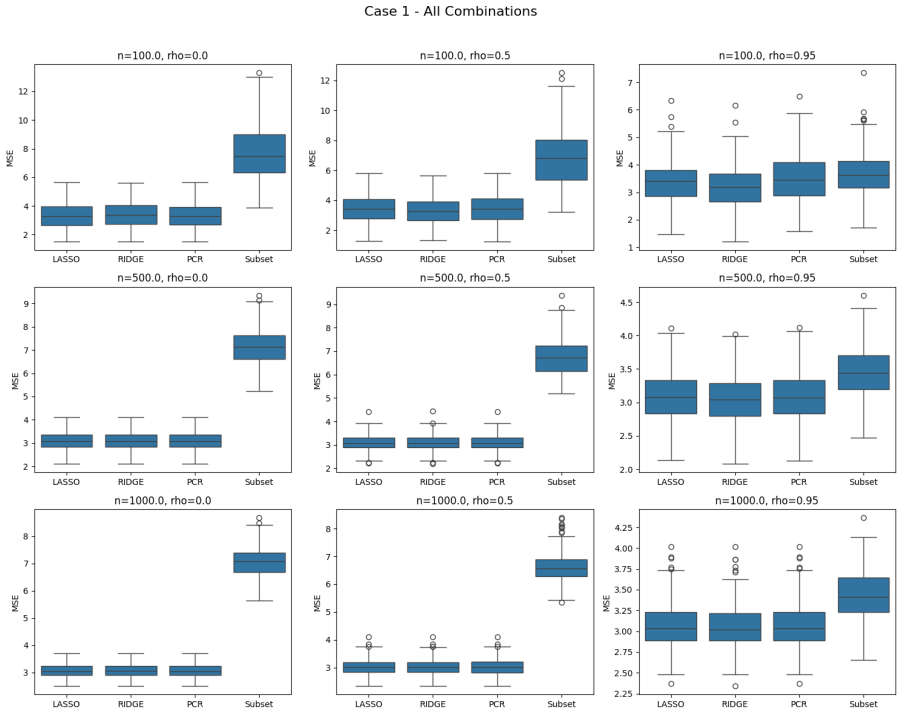

Figure 5: Case 1 all combinations

In the first figure 5 of Case 1, where $n = 100$ and $\rho = 0.0$, all models produced similar median MSE values, indicating low bias. However, the Subset model showed significantly higher variance and extreme outliers, suggesting instability when all predictors are equally important. LASSO, Ridge, and PCR all performed comparably with lower variance and more concentrated distributions. In this setting, Ridge appeared slightly more stable.

In the second plot with $n = 100$ and $\rho = 0.5$, the correlation among features introduced more dispersion in the test MSEs. Subset Selection again showed the worst performance with a noticeably wider spread and higher upper outliers. Ridge, LASSO, and PCR still maintained moderate variance, though Ridge slightly edged out the others in stability.

When the correlation increased to $\rho = 0.95$ (third plot), all models showed reduced variance, and their performances became more aligned. However, Subset Selection still lagged behind the regularized models. The high correlation among predictors minimized differences in the models' biases, though Ridge and LASSO retained an edge in robustness.

Moving to larger sample sizes, in the fourth plot where $n = 500$ and $\rho = 0.0$, all models exhibited reduced variance in test MSEs. Subset Selection, while still showing the widest spread, began to show some improvement in terms of stability. Ridge and LASSO had nearly identical performance, suggesting that with more data, the effect of regularization becomes

more consistent across these two methods.

In the fifth plot with $n = 500$ and $\rho = 0.5$, the spread of MSEs further decreased for all models. Subset Selection remained the worst performer, but the gap narrowed slightly. LASSO demonstrated a slight advantage over Ridge, likely due to its ability to shrink coefficients toward zero and manage moderate correlation effectively.

The sixth plot, at $n = 500$ and $\rho = 0.95$, illustrated a situation where high correlation and ample data led to a convergence of model performance. Although all methods performed similarly in terms of median MSE, the Subset model still displayed higher variance and a few upper outliers.

In the seventh plot with $n = 1000$ and $\rho = 0.0$, all models showed tightly grouped MSE distributions. Ridge and LASSO continued to perform well, with Ridge having a slight advantage. Subset Selection remained less stable, with wider variance and higher median MSE.

In the eighth plot, with $n = 1000$ and $\rho = 0.5$, the performances of LASSO, Ridge, and PCR nearly overlapped. Subset Selection continued to have higher variance and MSE, confirming that it struggles in non-sparse settings even with larger samples. Regularized models benefitted most from the combination of more data and moderate correlation.

Finally, in the ninth plot with $n = 1000$ and $\rho = 0.95$, all models displayed low variance and similar performance, though Subset Selection still had a slightly higher median MSE. LASSO and Ridge continued to show robust behavior under high correlation, with Ridge being slightly more stable overall.

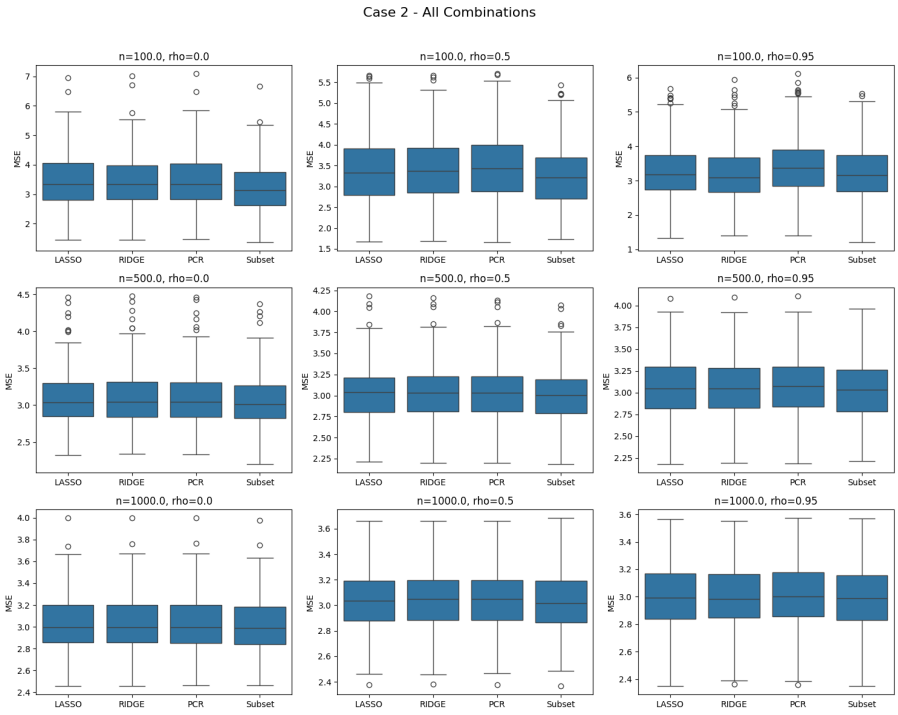

Figure 6: Case 2 all combinations

In the first figure 6 of Case 2, where $n = 100$ and $\rho = 0.0$, all models showed relatively high variance, but their median MSEs were comparable. Interestingly, Subset Selection performed slightly better in this case due to its capacity to exclude irrelevant predictors. This supports the idea that Subset methods can excel in sparse models where feature selection is crucial.

In the second plot, with $n = 100$ and $\rho = 0.5$, the variance remained high across all models, but the central tendency of the Subset model improved. It outperformed the regularized methods in terms of both lower median and tighter distribution. The presence of some irrelevant correlated features didn't drastically hurt Subset's performance, likely because the model could exclude the noisy predictors.

In the third plot, with $n = 100$ and $\rho = 0.95$, all models maintained a similar distribution of test MSEs. Subset Selection still performed slightly better, suggesting that when only a subset of predictors truly matters, the simplicity of Subset models is beneficial—even under high correlation.

In the fourth plot with $n = 500$ and $\rho = 0.0$, the variance of the test MSEs decreased for all models. Their performances became nearly indistinguishable, but Subset Selection continued to exhibit a competitive edge. Increasing the sample size helped stabilize performance, reducing the impact of random variability on variable selection.

In the fifth plot at $n = 500$ and $\rho = 0.5$, the results echoed those of the previous setting: all models clustered closely around similar MSE values. Subset remained slightly better in terms

of lower median and less spread, continuing to benefit from the sparsity of the true model.

In the sixth plot with $n = 500$ and $\rho = 0.95$, the convergence among model performances was more evident. All models showed tight distributions and close medians, with only slight differences between them. Subset was still marginally ahead in this sparse, high-correlation setup.

With the largest dataset, the seventh plot ($n = 1000$, $\rho = 0.0$) showed minimal variance and tight clustering of test MSEs. All models were nearly indistinguishable, with Subset, Ridge, LASSO, and PCR producing very similar performance. The advantage of model simplicity was less pronounced due to the overwhelming amount of data smoothing out individual predictor effects.

In the eighth plot, with $n = 1000$ and $\rho = 0.5$, the trend continued. Variance decreased further, and the performances overlapped almost completely. The increased sample size neutralized the sparsity advantage that Subset Selection initially had at smaller n.

Finally, in the ninth plot at $n = 1000$ and $\rho = 0.95$, all models exhibited the lowest variance observed so far. Subset Selection, LASSO, Ridge, and PCR converged to similar test MSEs, although a slight upward bias was noted across the board due to the strong correlation structure. At this stage, model choice became less critical, as all options yielded almost equivalent outcomes.

4 Conclusion

This report demonstrates the application of statistical learning techniques to both real-world classification problems and controlled regression experiments. For the flight delay prediction task, logistic regression provided the best trade-off between sensitivity and specificity, handling the class imbalance with a modified threshold. Feature selection through correlation filtering and LASSO proved essential in reducing model complexity without sacrificing accuracy.

In the second exercise, model performance varied with sparsity and data structure. Ridge regression performed best in dense settings, while Subset Selection and LASSO excelled in sparse cases. Larger sample sizes reduced MSE variance, and high predictor correlation lowered variance but increased bias slightly.

Overall, the experiments underscore the importance of model selection, feature engineering, and proper evaluation metrics in achieving reliable predictive performance in both classification and regression tasks.