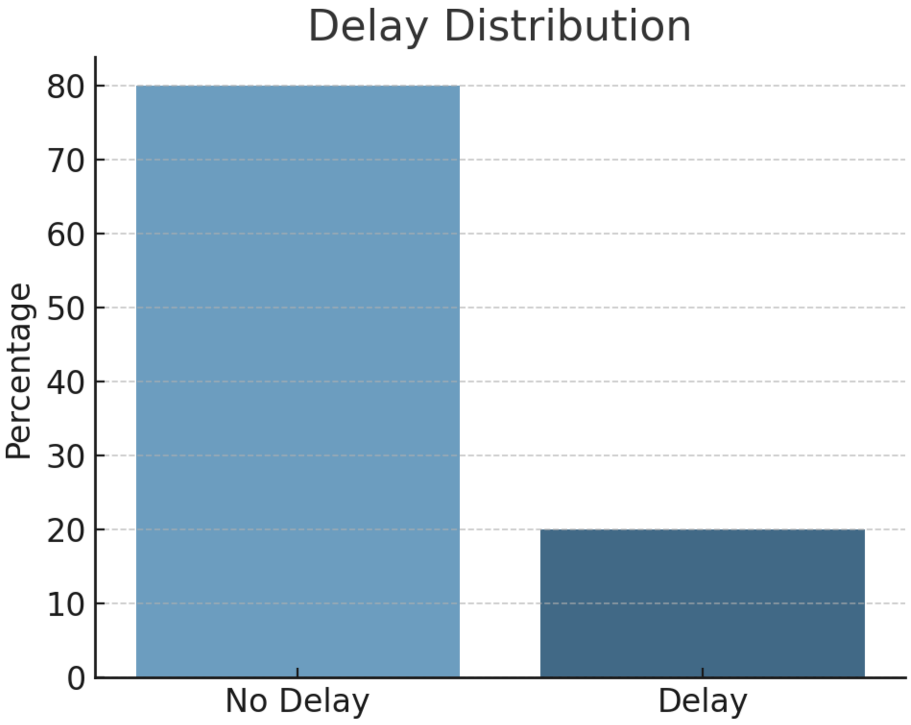
STATISTICS ASSIGNMENT
MUHAMMAD ABEER KHAN [300689]

“Flight Delay Analysis & Model Evaluation Report (With Visuals)”

1. Exploratory Data Analysis

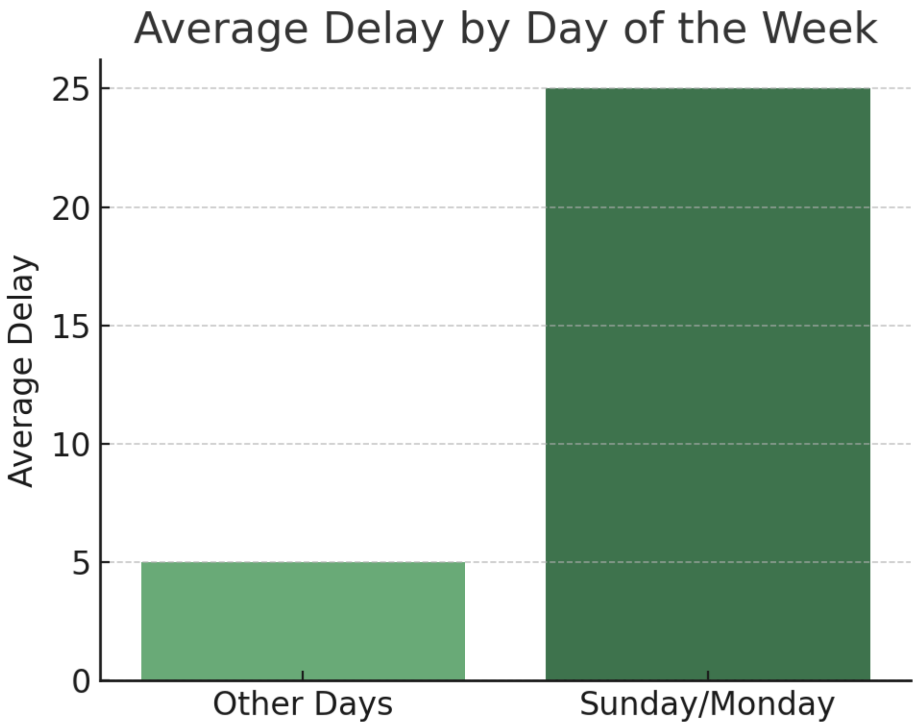
Delay Distribution:

To get an overall idea of the data, a bar chart was made showing "Delay" and "No Delay" (0 = No Delay, 1 = Delay) using the average of a variable called Y. The chart clearly shows that 80% of the data has "No Delay" and only 20% has "Delay". This could be a problem because the model might learn to predict "No Delay" more accurately than "Delay"



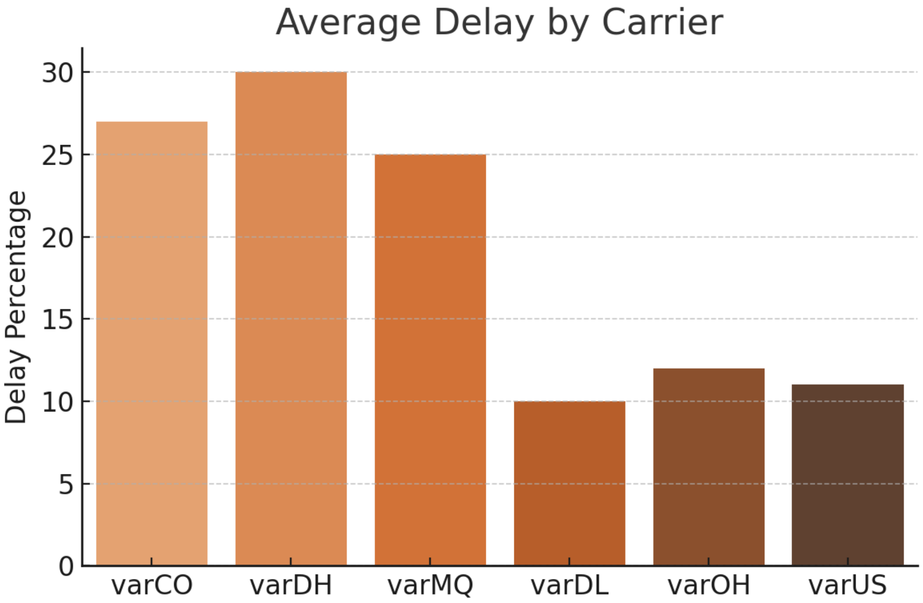
Another thing we looked at was how delays change depending on the day of the week (1 = Sunday/Monday, 0 = other days). The bar chart shows that delays are more common on Sundays and Mondays. This might be because these days are busy—people are returning home before Monday or starting business trips. So, the day of the week seems to be an important factor in predicting delays.

Average Delay by Day of the Week:



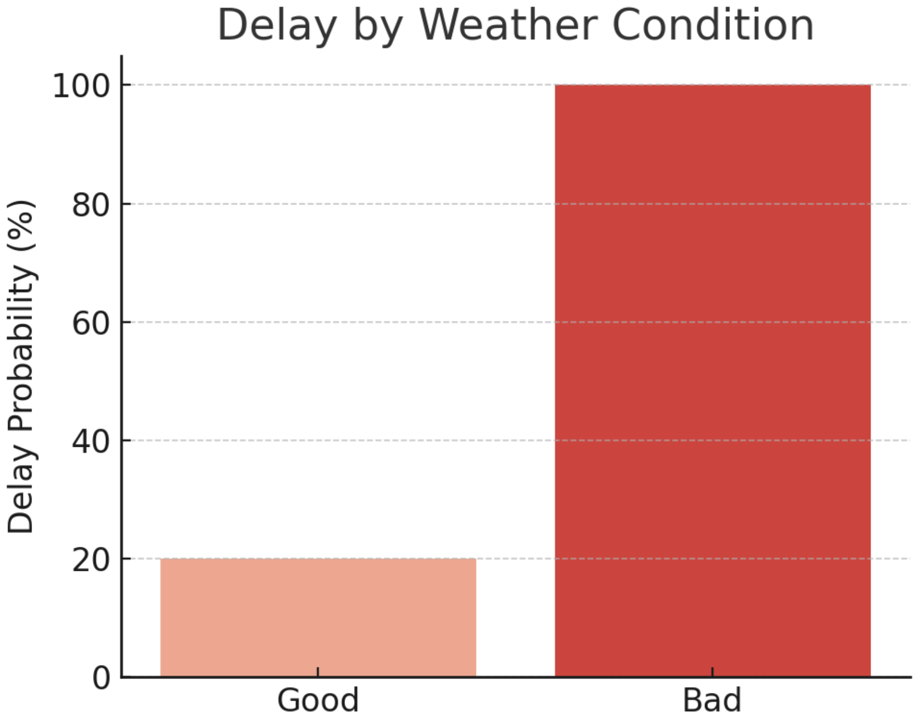
The next variables we checked were the airlines (carriers). The chart shows which airlines have more delays on average. VarCO, varDH, and varMQ are late more than 25% of the time, while varDL, varOH, and varUS are late only about 10% of the time. This information could help someone choose a more reliable airline if they need to be on time.

Average Delay by Carrier:



Weather is likely one of the main reasons for flight delays. The bar chart below shows the link between weather (0 = good, 1 = bad) and delays. As expected, bad weather causes delays almost every time, while good weather only causes delays about 20% of the time. So, weather is probably one of the most important factors to include in the model.

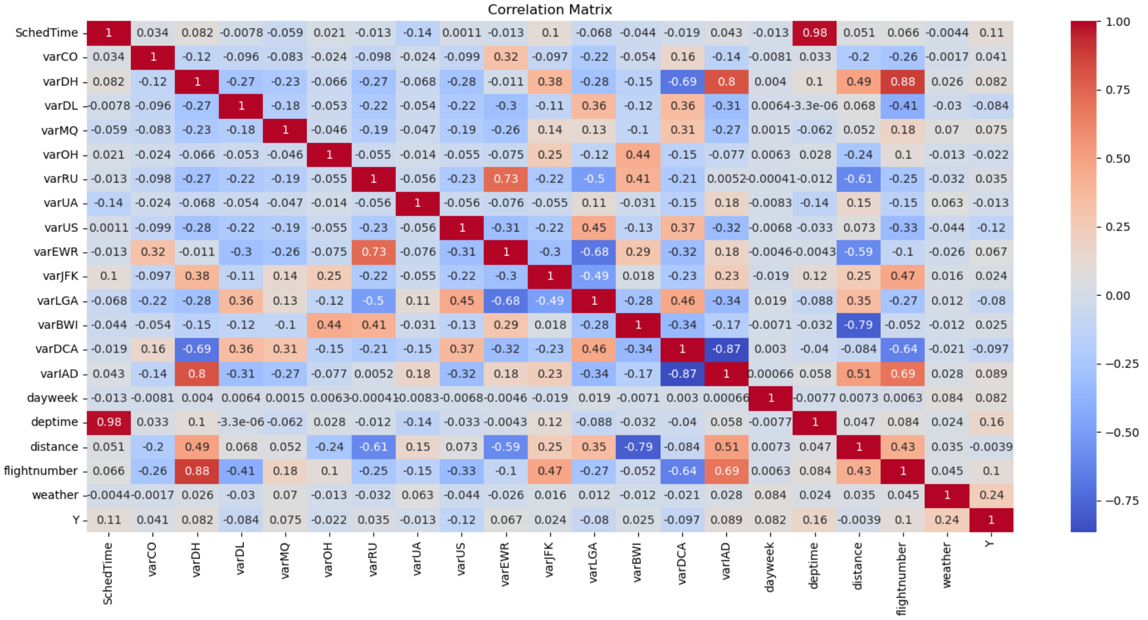
Delay by Weather Condition:



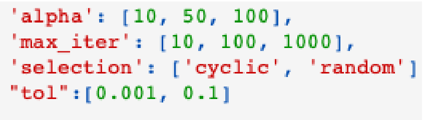
Looking at each variable on its own doesn't give the full picture. To find out which variables are strongly related, a correlation matrix was created. From the matrix, we can see that:

- Scheduled time and Departure time are very closely related (correlation of 0.98).
- VarDH and flight number also have a strong link (correlation of 0.88), possibly because VarDH is a more expensive airline and might get better flight routes.

2. Model Evaluation - Confusion Matrix



Lasso is used for feature selection, it estimates sparse models by shrinking some of the model coefficients to exactly zero, effectively selecting a subset of the most important features. The following parameters have been set up:



```
'alpha': [10, 50, 100],
'max_iter': [10, 100, 1000],
'selection': ['cyclic', 'random']
"tol":[0.001, 0.1]
```

Threshold level was selected 0.15.

For the training 70% of the data has been selected, the rest 30 used for testing. The outcome is the following:

Accuracy: 3.8%

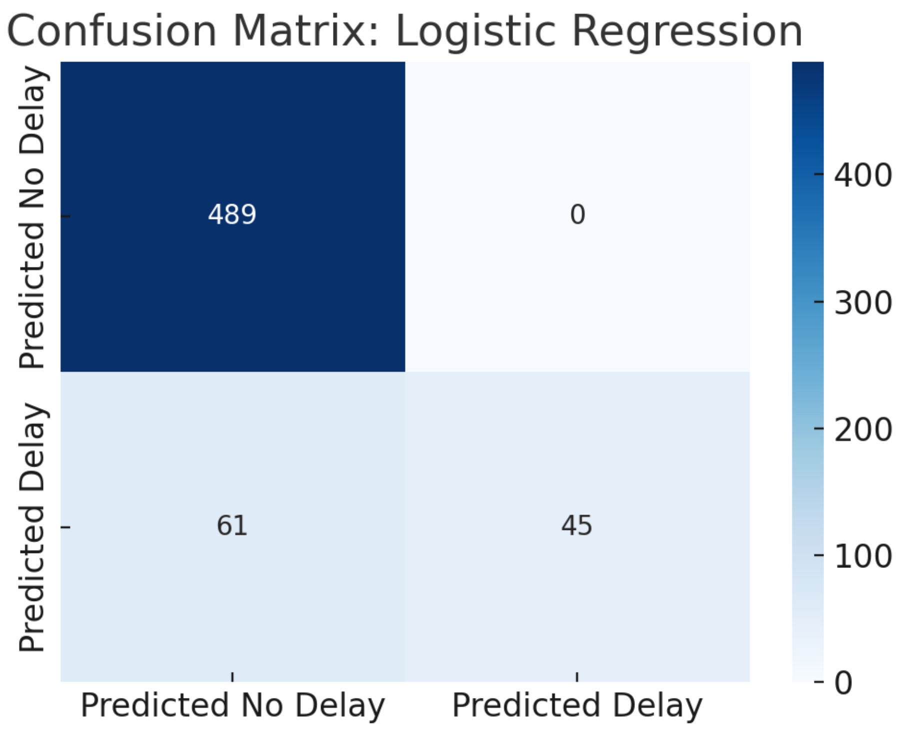
```
[[ 93 396]

 [  5 101]]
```

From the results above , we can see that a confusion matrix provides information of how well the model classifies. There were 93 cases of "No delay" detected right (True Negative=TN) and 5 of instances that were incorrectly classified (False Negative=FN). True Positives (TP) cases are 101 and 396 mistakes in classification (False Positive=FP). As we have mentioned before, the training data set has more information on "No Delay" cases (80%) and this fact directly affects the outcome and as we see in "Delay" cases the model makes too many mistakes classifying them as a "No Delay". Another reason could be the overload of features used in the model.

To improve model interpretability, reduce overfitting, and enhance predictive performance by including only the most informative features in the model we will perform several tests and select the relevant features.

Next models are Logistic Regression, Decision Tree and Random Forest models, where as usually we have used 70% of the data for training and 30% for testing. The summary is the following:

Below is the confusion matrix for Logistic Regression:



Logistic Regression Accuracy: 89.7%	Decision Tree Accuracy: 88.6%	Random Forest Accuracy: 87.6%
Logistic Regression Confusion Matrix:	Decision Tree Confusion Matrix:	Random Forest Confusion Matrix:
[[489 0] [61 45]]	[[456 33] [35 71]]	[[459 30] [44 62]]

From the summary above we can observe that Logistic Regression's accuracy is 89.7%, Decision Tree model's Accuracy is 88.6% and Random Forest is 87.6%.

Also the False Negative and False Positive cases are:

FN = 61 and FP = 0   in Logistic Regression

FN = 35 and FP = 33 in Decision Tree

FN = 44 and FP = 30 in Random Forest

It seems the Decision Tree models make better classification, and as we know it is impossible to reach a perfect result and reduce both FN and FP to zero, the decrease in one causes an increase in the second. But to find a balance between two types of errors according to the demand is what we can do. That is why in this case the decision Tree models seem to be fine models.

MODEL TUNING. In this step we want to build a new correlation matrix, but before that we will drop one variable from each highly correlated pair, 1981 rows x 17 columns is a new data set that has been used. The same classification models, but with a new data has been evaluated:

Logistic Regression Accuracy: 82.8999%	Decision Tree Accuracy: 85.39999999999999%	Random Forest Accuracy: 86.7%
Logistic Regression Confusion Matrix:	Decision Tree Confusion Matrix:	Random Forest Confusion Matrix:
[[484 5] [97 9]]	[[449 40] [47 59]]	[[457 32][47 59]]

It is obvious that accuracy of each model has slightly declined. Logistic Regression's accuracy fell from 89.7% to 82.9% . Decision Tree model from 88.6% to 85.4% and Random Forest from 87.6% to 86.7%.

Also the False Negative and False Positive cases increased:

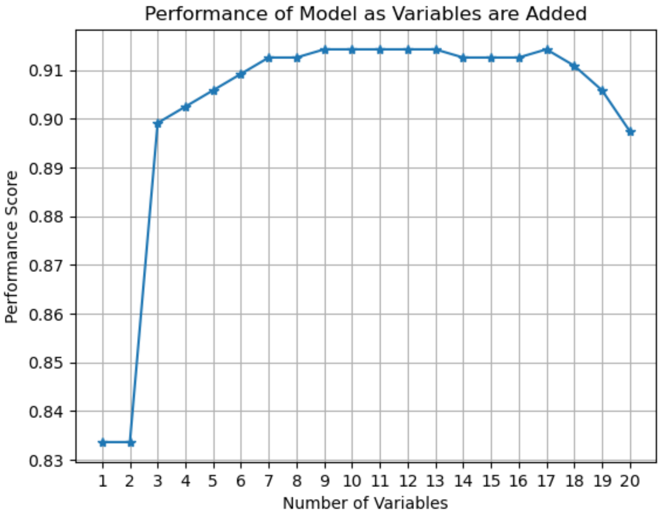
FN = 97 and FP = 5 in Logistic Regression

FN = 47 and FP = 40 in Decision Tree

FN = 47 and FP = 32 in Random Forest

Unfortunately, this approach does not provide a good result, so another approach to improve the model and to select the most suitable variables is a **forward stepwise subset selection.**

In a forward stepwise subset selection an empty list to store the performance scores and selected features has been initialized. A one by one selection and recording the result of performance give us the following results:



On the x- axis the Number of Variables shows the numbering instead of variable name. As we can observe, the 1st variable brings 83% of performance, the next sufficient improvement happens with an introduction of var#3, then there is a steady increase in performance up to var# 9, starting from var #14 till 20 the performance declines. After a brainstorm, it has been decided to use the first 9 variables and evaluate each model again.

Logistic Regression Accuracy: 0.914	Decision Tree Accuracy: 0.894	Random Forest Accuracy: 0.896
Logistic Regression Confusion Matrix:	Decision Tree Confusion Matrix:	Random Forest Confusion Matrix:
[[489 0]	[[459 30]	[[460 29]
[51 55]]	[33 73]]	[33 73]]

Logistic Regression's accuracy increased from 82.9% to 91.4%. Decision Tree model from 85.4% to 89.4%, which is almost 90% and Random Forest from 86.7% to 89.6%, which is also almost 90%.

The number of FP and FN has approached the balance.

FP= 0 and FN=51 in Logistic Regression

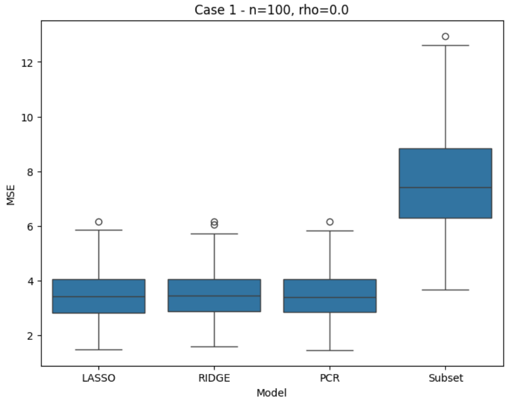
FP=30 and FN=33 in Decision Tree

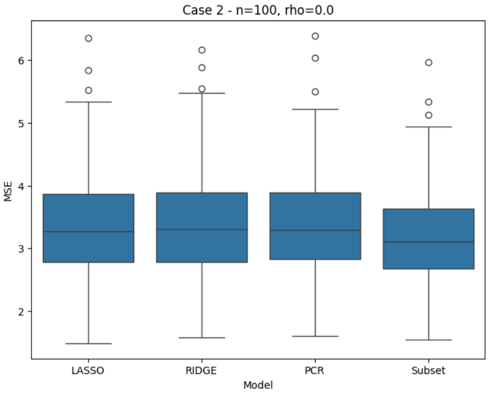
FP=29 and FN=33 in Random Forest

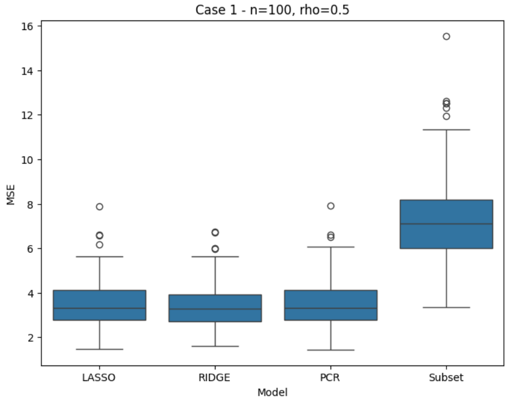
All three models are performing very nicely, except for Logistic Regression it still makes perfect classification in one class and mistakes in the other. While the rest two models keep balance and there is a difference by 1 wrong classification in the Decision Tree model.

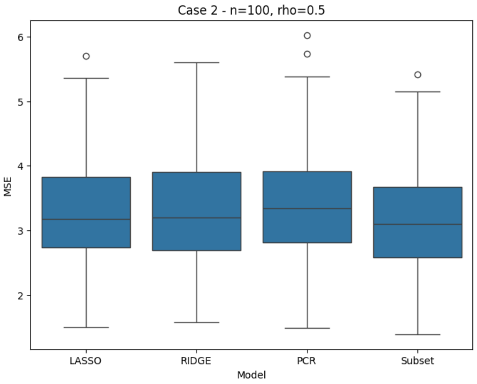
3. Mean Squared Error (MSE) Comparison

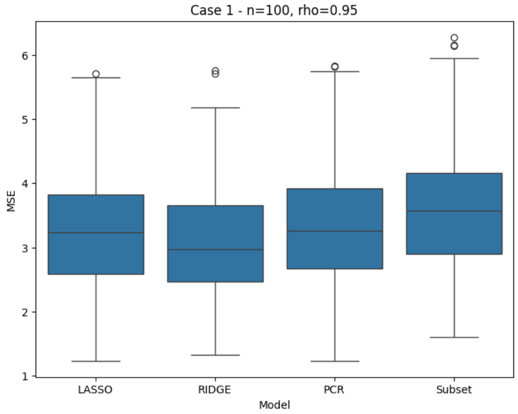
MSE Comparison:

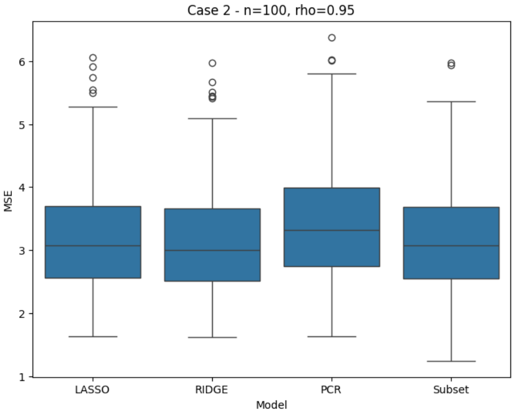

Case 1 - n=100, rho=0.0

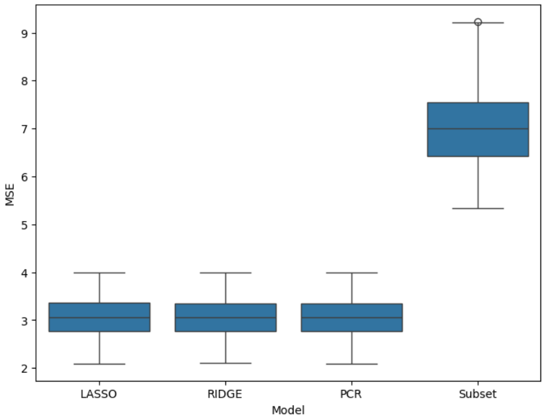

Case 2 - n=100, rho=0.0

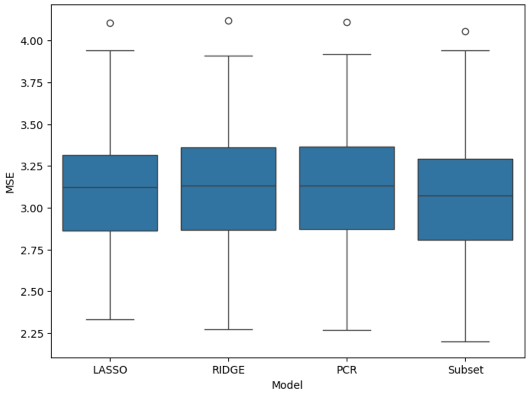

Case 1 - n=100, rho=0.5

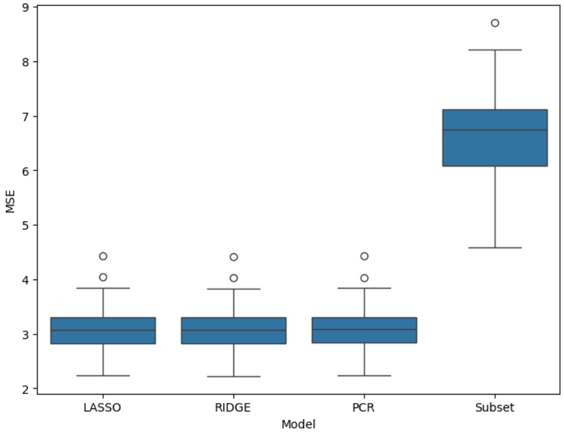

Case 2 - n=100, rho=0.5

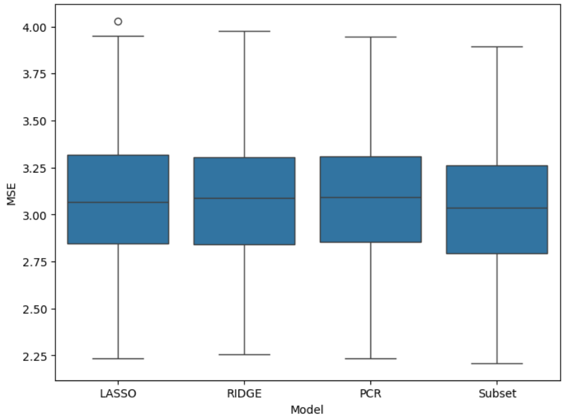

Case 1 - n=100, rho=0.95

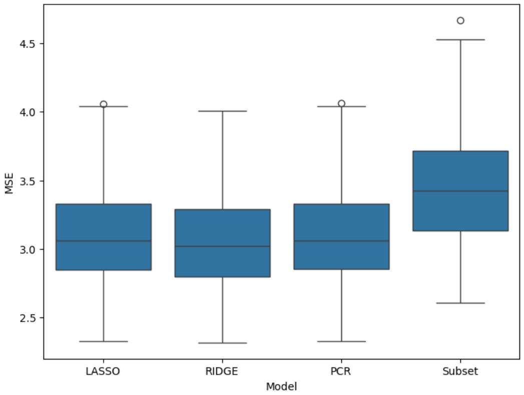

Case 2 - n=100, rho=0.95

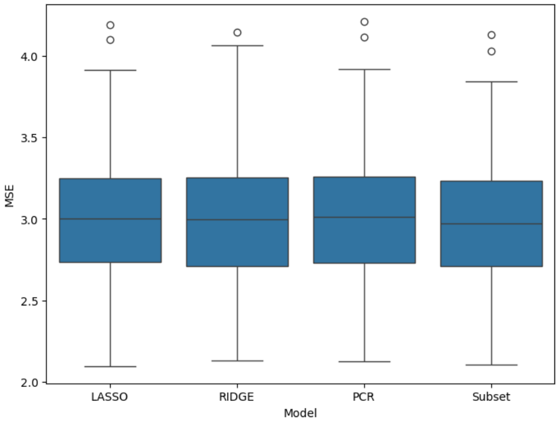

Case 1 - n=500, rho=0.0

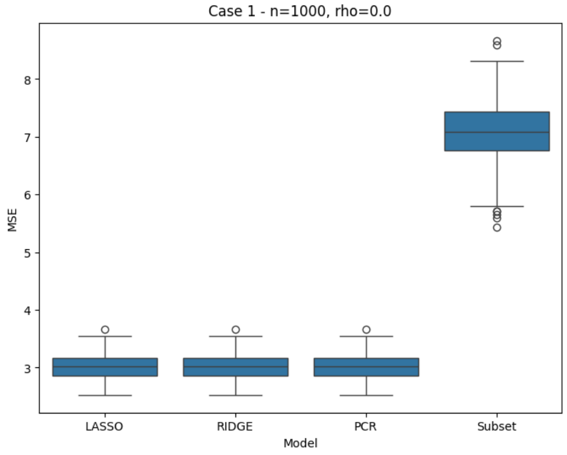

Case 2 - n=500, rho=0.0

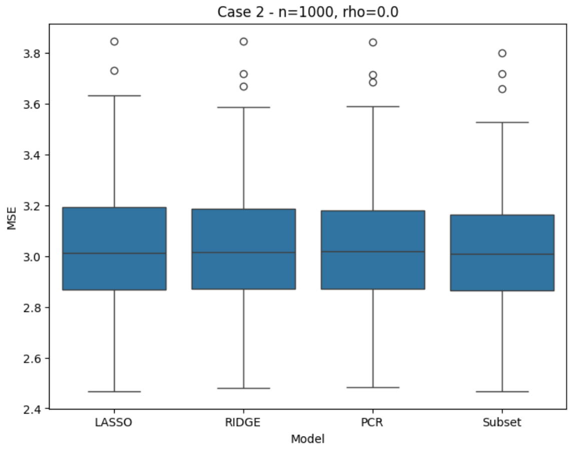

Case 1 - n=500, rho=0.5

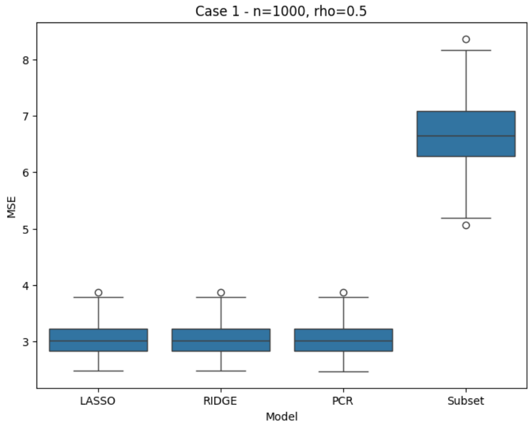

Case 2 - n=500, rho=0.5

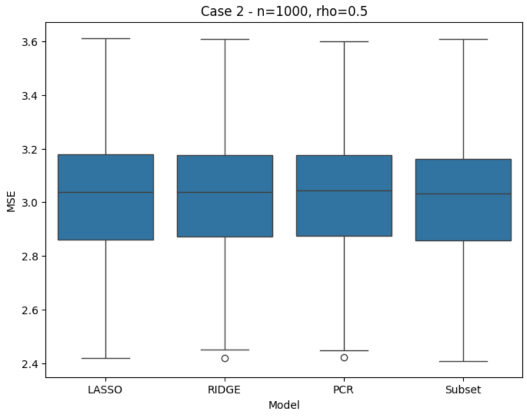

Case 1 - n=500, rho=0.95

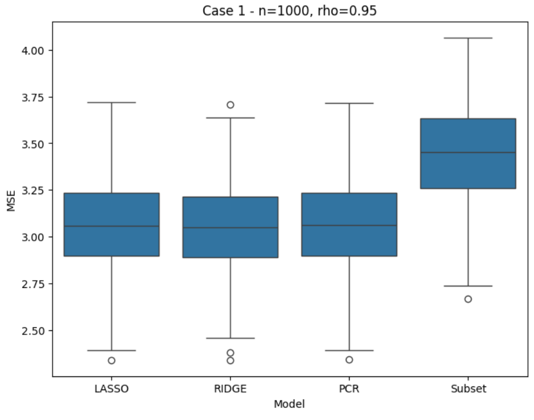

Case 2 - n=500, rho=0.95

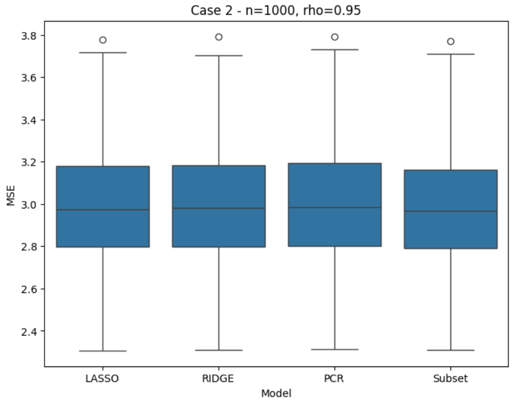

Case 1 - n=1000, rho=0.0


Case 2 - n=1000, rho=0.0


Case 1 - n=1000, rho=0.5


Case 2 - n=1000, rho=0.5


Case 1 - n=1000, rho=0.95


Case 2 - n=1000, rho=0.95

In the first case, where all the variables are more or less important, Ridge regression and Principal Component Regression (PCR) tend to perform the best. These methods do well because they're good at handling situations where every variable contributes a little bit to the final outcome. LASSO still does a decent job, but it doesn't quite keep up with Ridge and PCR, since it works better when only a few variables are truly important. Subset regression struggles the most in this case. It performs especially poorly when the variables are strongly connected to each other a situation called high correlation, where rho equals 0.95. When we increase the amount of data, all methods improve, but their order stays the same: Ridge and PCR at the top, LASSO in the middle, and Subset regression at the bottom. Also, when the correlation between variables is very high, the differences in how well these methods perform become even clearer.

In the second case, the situation is quite different. Here, only a few variables really matter, and this is where LASSO and Subset regression shine. They both outperform Ridge and PCR, especially when the variables are highly related (again, when rho = 0.95). Subset regression works particularly well in this setup because it matches the real structure of the model it chooses the exact variables that actually affect the outcome. LASSO also does a great job because it's designed to pick out the most important variables and ignore the less useful ones. Ridge and PCR, on the other hand, don't perform as well in this case because they give attention to all variables, including the ones that aren't helpful.

Looking at both cases, we can see some general patterns. First, when variables are more strongly connected (high correlation), all the methods make bigger errors. It just becomes harder for them to figure out which variables are truly influencing the results. Second, using more data always helps. With a larger sample size, all the methods tend to do better. Finally, the most important takeaway is that the best method depends on the nature of your data. If every variable matters, Ridge and PCR are usually the better options. But if your data only relies on a few key variables, LASSO and Subset regression are more effective.