# Your Compact Report

Your Name

November 06, 2025

## Contents



# 1 Problem description and objectives

# 2 Data description

The dataset used for this project is the Breast Cancer Wisconsin (Diagnostic) Dataset (WDBC), a classic benchmark for classification sourced from the UCI Machine Learning Repository.

The dataset contains 569 observations of breast mass samples. The problem is a binary classification task, where the target variable, diagnosis, is classified as either Malignant (212 observations) or Benign (357 observations).

Each instance is described by 30 numeric features that were computed from a digitized image of a fine needle aspirate (FNA) of the breast mass.

# 3 Pre-processing

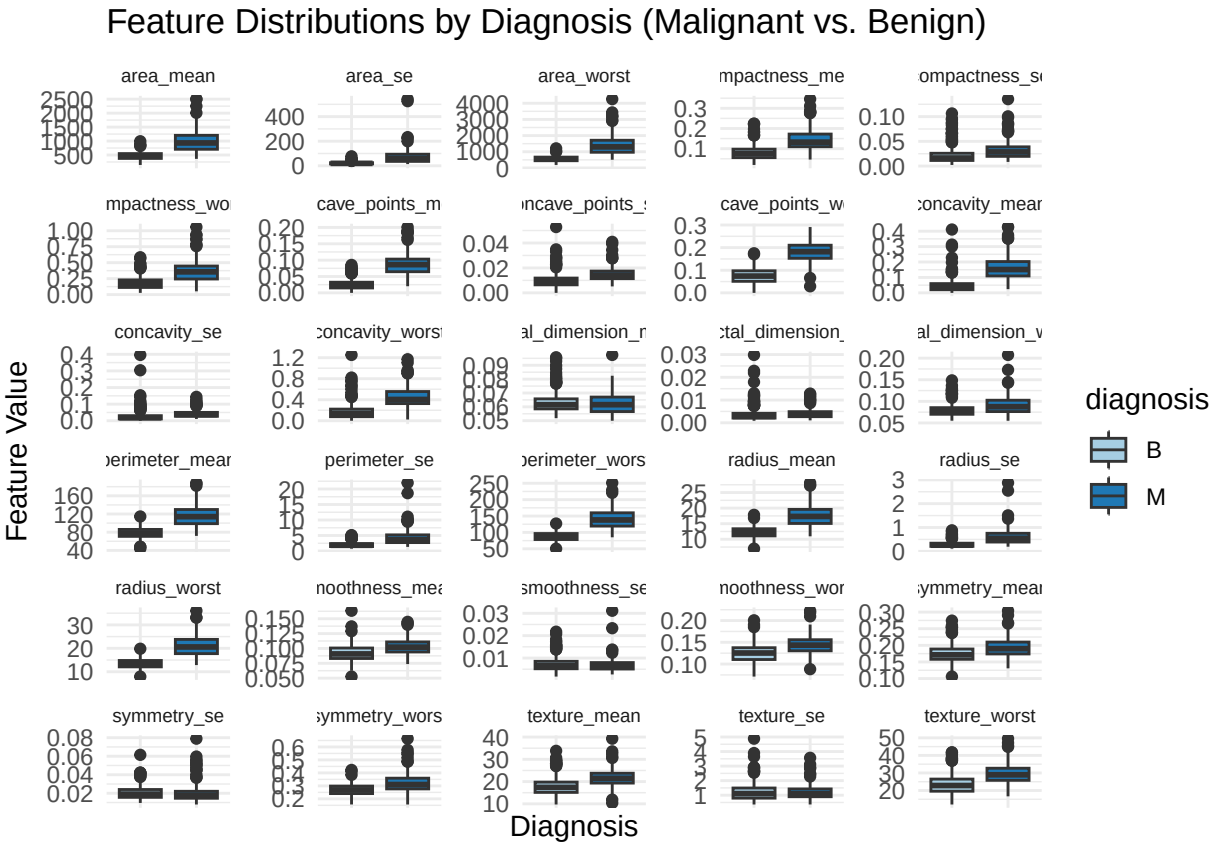
## 3.1 Missing values

```
## Total missing values in the entire dataset: 0
```

## 3.2 Outliers, features distributions - scales

```
# To plot all 30 features at once, we first "pivot" the data
# into a "long" format. This is the standard tidyverse way.
data_long <- data_clean %>%
  pivot_longer(
    cols = -diagnosis,       # Pivot every column *except* 'diagnosis'
    names_to = "feature",    # New column for the feature name
    values_to = "value"      # New column for its value
  )
```

```r
ggplot(data_long, aes(x = diagnosis, y = value, fill = diagnosis)) +
  geom_boxplot() +
  facet_wrap(~feature, scales = "free_y", ncol = 5) +
  scale_fill_brewer(palette = "Paired") +
  labs(
    title = "Feature Distributions by Diagnosis (Malignant vs. Benign)",
    x = "Diagnosis",
    y = "Feature Value"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_blank(),  # Hide x-axis labels for a cleaner look
    axis.ticks.x = element_blank(),
    strip.text = element_text(size = 7) # Smaller facet labels
  )
```




- **Outliers (Anomalous Value):** We immediately observed the presence of numerous outliers, represented by individual observations outside the IQR range. These are particularly present in features like `area_mean`, `area_worst`, and `concavity_worst`. However, because those trust the original source, we know that these are not data entry errors but likely represent true, extreme biological values. Therefore, we made the decision not to remove these outliers, as they are part of the real-world problem. We will proceed with the full dataset, relying on our models to be robust enough to handle this variance.

## 3.3 Normalization

Our plot also highlights the vast difference in scales across features. Gradient-based models like the Perceptron and Logistic Regression (GLM) are sensitive to feature scales. Without normalization, features with larger magnitudes would dominate the learning process. This plot provides a clear justification for standardizing the features before modeling.

### 3.4 Elimination of irrelevant variables

For nearly every feature, the boxplot for the Malignant (M) class is visually distinct from the boxplot for the Benign (B) class. The medians, quartiles, and overall ranges show clear separation. This is an excellent sign, as it indicates that our features contain strong predictive information, and we can expect our models to perform well.

```
# With AI
nzv_check <- nearZeroVar(data_clean, saveMetrics = TRUE)

# Filter to see only the features that *might* be a problem
# A "TRUE" in the 'nzv' column means it's a candidate for removal.
problematic_features <- nzv_check[nzv_check$nzv == TRUE, ]

# Print the results
if (nrow(problematic_features) > 0) {
  cat("Found", nrow(problematic_features), "near-zero variance features:\n")
  print(problematic_features)
} else {
  cat("No near-zero variance features found. All features have sufficient variance.\n")
}
```

```
## No near-zero variance features found. All features have sufficient variance.
```

First, we checked for "irrelevant" features by running a "Near-Zero Variance" (NZV) test. This test finds any feature that is constant or nearly constant, as a feature that doesn't change cannot be a predictor. We used the `caret::nearZeroVar()` function to check all 30 features.

we note that we used AI to help confirm this was the correct tool for this task.

The NZV test showed that all 30 features have enough variance, so we did not eliminate any features at this step.

### 3.5 Elimination of redundant variables

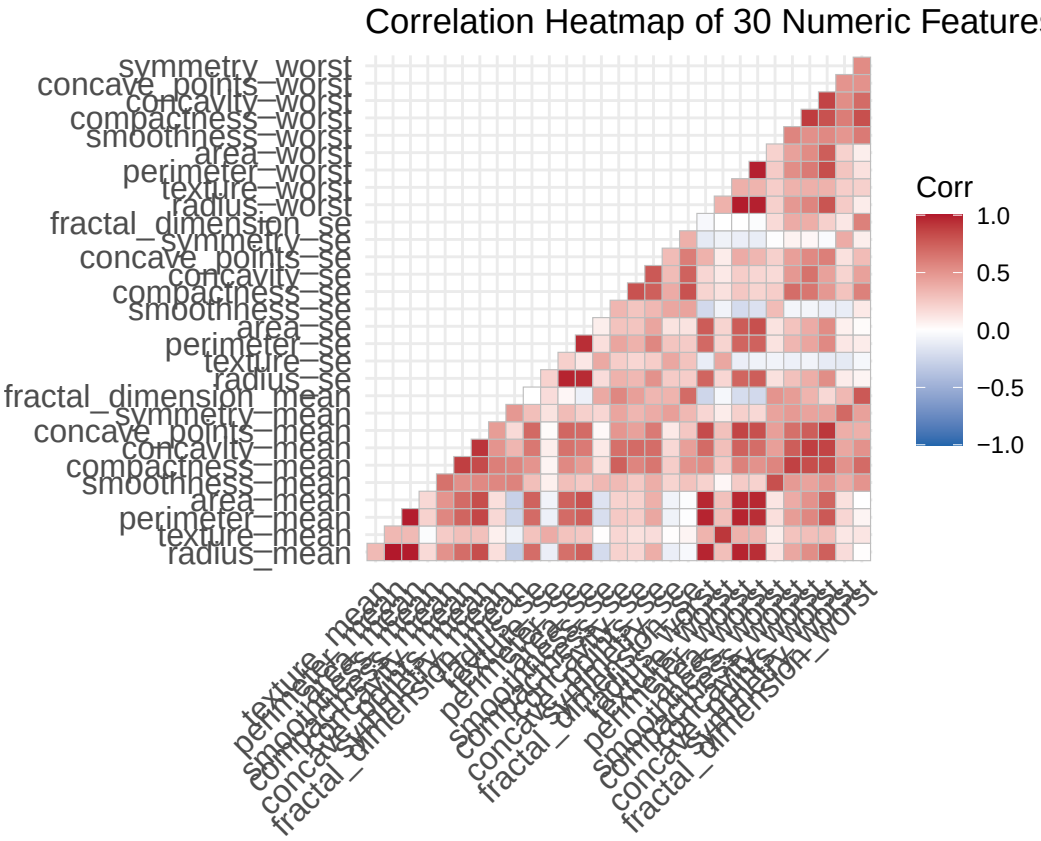
Our primary tool for this step was correlation analysis. We first calculated the correlation matrix for all numeric features and visualized it using a heatmap.

```
# Correlation matrix.
# numeric features selections
numeric_features <- data_clean %>%
  select_if(is.numeric)

cor_matrix <- cor(numeric_features)

# Plot the correlation heatmap
ggcorrplot(cor_matrix,
  type = "lower",
  lab = FALSE,
  colors = c("#2166AC", "white", "#B2182B"),
  title = "Correlation Heatmap of 30 Numeric Features"
)
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## i The deprecated feature was likely used in the ggcorrplot package.
##   Please report the issue at <https://github.com/kassambara/ggcorrplot/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




The heatmap immediately revealed large blocks of high correlation (dark red), confirming that our dataset contains significant multicollinearity. For example, features related to tumor size, like radius_mean, perimeter_mean, and area_mean, are all nearly perfectly correlated.

```
# Correlation and cutoff

highly_correlated_features <- findCorrelation(cor_matrix, cutoff = 0.90, names = TRUE)


cat("Found", length(highly_correlated_features), "redundant features to remove (cutoff > 0.90):\n")
```

```
## Found 10 redundant features to remove (cutoff > 0.90):
```

```
print(highly_correlated_features)
```

```
##  [1] "concavity_mean"      "concave_points_mean" "perimeter_worst"
##  [4] "radius_worst"        "perimeter_mean"      "area_worst"
##  [7] "radius_mean"         "perimeter_se"        "area_se"
## [10] "texture_mean"
```

```
# Create the reduced Dataset

data_clean_reduced <- data_clean %>%
  select(-all_of(highly_correlated_features))

cat("\nOriginal data had", ncol(data_clean), "columns (30 features + 1 target).\n")
```

```
##
## Original data had 31 columns (30 features + 1 target).
```

```
cat("Reduced data has", ncol(data_clean_reduced), "columns.\n")
```

```
## Reduced data has 21 columns.
```

```
head(data_clean_reduced)
```

```
##   diagnosis area_mean smoothness_mean compactness_mean symmetry_mean
## 1         M    1001.0         0.11840          0.27760        0.2419
## 2         M    1326.0         0.08474          0.07864        0.1812
## 3         M    1203.0         0.10960          0.15990        0.2069
## 4         M     386.1         0.14250          0.28390        0.2597
## 5         M    1297.0         0.10030          0.13280        0.1809
## 6         M     477.1         0.12780          0.17000        0.2087
##   fractal_dimension_mean radius_se texture_se smoothness_se compactness_se
## 1                0.07871    1.0950     0.9053      0.006399        0.04904
## 2                0.05667    0.5435     0.7339      0.005225        0.01308
## 3                0.05999    0.7456     0.7869      0.006150        0.04006
## 4                0.09744    0.4956     1.1560      0.009110        0.07458
## 5                0.05883    0.7572     0.7813      0.011490        0.02461
## 6                0.07613    0.3345     0.8902      0.007510        0.03345
##   concavity_se concave_points_se symmetry_se fractal_dimension_se texture_worst
## 1      0.05373           0.01587     0.03003             0.006193         17.33
## 2      0.01860           0.01340     0.01389             0.003532         23.41
## 3      0.03832           0.02058     0.02250             0.004571         25.53
## 4      0.05661           0.01867     0.05963             0.009208         26.50
## 5      0.05688           0.01885     0.01756             0.005115         16.67
## 6      0.03672           0.01137     0.02165             0.005082         23.75
##   smoothness_worst compactness_worst concavity_worst concave_points_worst
## 1           0.1622            0.6656          0.7119               0.2654
## 2           0.1238            0.1866          0.2416               0.1860
## 3           0.1444            0.4245          0.4504               0.2430
## 4           0.2098            0.8663          0.6869               0.2575
## 5           0.1374            0.2050          0.4000               0.1625
## 6           0.1791            0.5249          0.5355               0.1741
##   symmetry_worst fractal_dimension_worst
## 1         0.4601                 0.11890
## 2         0.2750                 0.08902
## 3         0.3613                 0.08758
## 4         0.6638                 0.17300
## 5         0.2364                 0.07678
## 6         0.3985                 0.12440
```

While the heatmap is excellent for visualization, we use a correlation matrix and a cutoff of 0.90 (standard threshold) for reducing feature redundancy. This function automatically identifies the minimum number of features to remove to ensure that no two features in the remaining dataset have a correlation greater than 0.90.

This analysis identified 10 features as redundant.

### 3.6 Normalization of the variables

done in models

# 4 Models

```
## Starting Experiment 1: 20 Reduced Features
```

```
## Starting Experiment 2: PCA Components
```

```
## Starting Experiment 3: LDA Component (2-Step Method)
```

```
## Training LDA model (to be used as transformer)...
```

```
## Extracting LD1 component...
```

```
## Training GLM on LDA component...
## Training GNB on LDA component...
## Training SVM on LDA component...
## All LDA-component models trained.
```

```
# --- 8. Final Results Comparison (Corrected) ---
# We now collect all our models.
all_model_results <- resamples(list(
  # Experiment 1 (Reduced 20 Features)
  GLM_Reduced = model_glm_reduced,
  GNB_Reduced = model_gnb_reduced,
  SVM_Reduced = model_svm_reduced,
  LDA_Reduced = model_lda_reduced,

  # Experiment 2 (PCA Components)
  GLM_PCA     = model_glm_pca,
  GNB_PCA     = model_gnb_pca,
  SVM_PCA     = model_svm_pca,

  # Experiment 3 (LDA Component)
  GLM_LDA_Comp = model_glm_lda,
  GNB_LDA_Comp = model_gnb_lda,
  SVM_LDA_Comp = model_svm_lda
))

# 1. Show the summary table
cat("Final CV Performance Summary:\n")
```

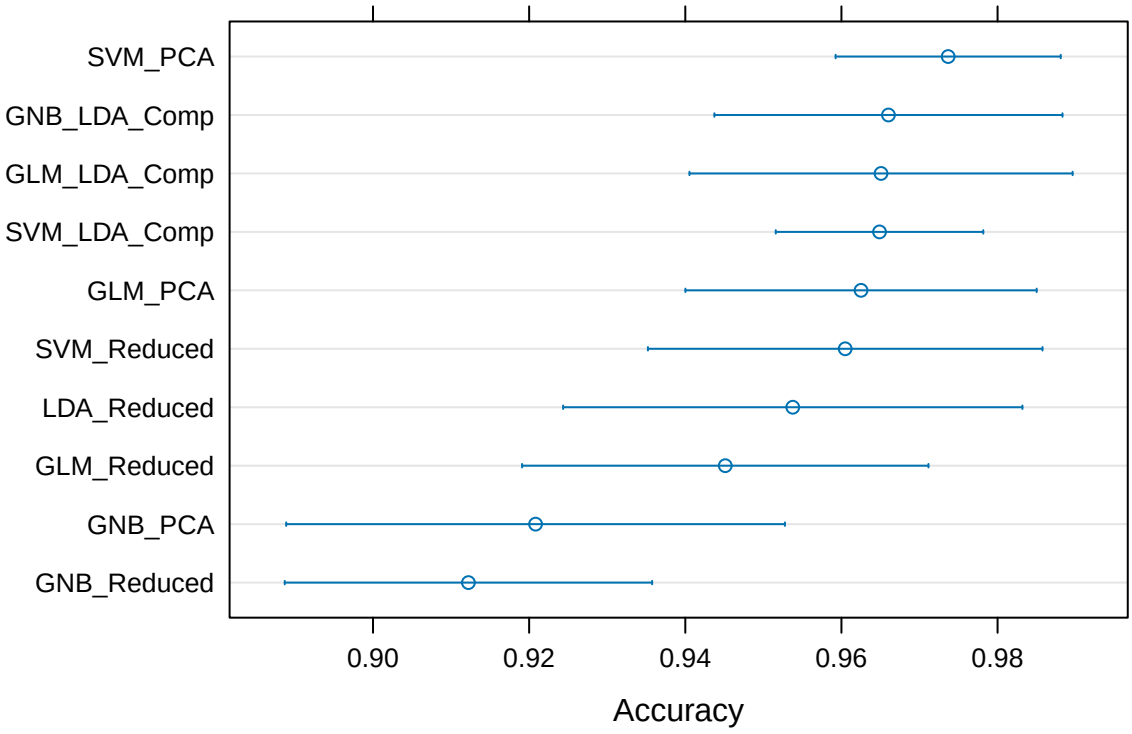
```
## Final CV Performance Summary:
```

```
summary(all_model_results)
```

```
##
## Call:
## summary.resamples(object = all_model_results)
##
## Models: GLM_Reduced, GNB_Reduced, SVM_Reduced, LDA_Reduced, GLM_PCA, GNB_PCA, SVM_PCA, GLM_LDA_Comp, GNB
## Number of resamples: 10
##
## Accuracy
##                   Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLM_Reduced  0.8888889 0.9170290 0.9565217 0.9451208 0.9777778 0.9782609    0
## GNB_Reduced  0.8666667 0.8894928 0.9120773 0.9122222 0.9347826 0.9565217    0
## SVM_Reduced  0.8913043 0.9399758 0.9560386 0.9604831 0.9944444 1.0000000    0
## LDA_Reduced  0.8666667 0.9402174 0.9565217 0.9537681 0.9777778 1.0000000    0
## GLM_PCA      0.9111111 0.9336957 0.9780193 0.9625121 0.9782609 1.0000000    0
## GNB_PCA      0.8444444 0.8888889 0.9347826 0.9208213 0.9562802 0.9782609    0
## SVM_PCA      0.9555556 0.9565217 0.9671498 0.9736715 0.9945652 1.0000000    0
## GLM_LDA_Comp 0.9130435 0.9347826 0.9666667 0.9650725 1.0000000 1.0000000    0
## GNB_LDA_Comp 0.9130435 0.9555556 0.9777778 0.9660225 0.9782609 1.0000000    1
## SVM_LDA_Comp 0.9333333 0.9557971 0.9777778 0.9648792 0.9782609 0.9782609    0
##
## Kappa
##                   Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLM_Reduced  0.7695391 0.8259754 0.9066384 0.8843928 0.9521785 0.9539078    0
## GNB_Reduced  0.7096774 0.7644073 0.8075748 0.8116355 0.8617234 0.9066937    0
## SVM_Reduced  0.7801147 0.8682082 0.9037959 0.9154383 0.9880446 1.0000000    0
```

```
## LDA_Reduced  0.6952596 0.8671441 0.9043659 0.8973903 0.9521785 1.0000000    0
## GLM_PCA      0.8109244 0.8602907 0.9524753 0.9200627 0.9536239 1.0000000    0
## GNB_PCA      0.6488294 0.7506651 0.8600198 0.8240334 0.9043659 0.9527721    0
## SVM_PCA      0.9032258 0.9043659 0.9305447 0.9430181 0.9884770 1.0000000    0
## GLM_LDA_Comp 0.8087318 0.8547368 0.9277022 0.9228347 1.0000000 1.0000000    0
## GNB_LDA_Comp 0.8038380 0.9032258 0.9521785 0.9248766 0.9527721 1.0000000    1
## SVM_LDA_Comp 0.8531012 0.9040928 0.9521785 0.9233203 0.9527721 0.9527721    0
```

```
# 2. Plot the results
dotplot(all_model_results, metric = "Accuracy")
```




```
# 3. Statistical Comparison (as requested by teacher)
cat("\nFinal Statistical Comparison (Paired t-test):\n")
```

```
##
## Final Statistical Comparison (Paired t-test):
```

```
all_model_diffs <- diff(all_model_results)
summary(all_model_diffs)
```

```
##
## Call:
## summary.diff.resamples(object = all_model_diffs)
##
## p-value adjustment: bonferroni
## Upper diagonal: estimates of the difference
## Lower diagonal: p-value for H0: difference = 0
##
## Accuracy
##              GLM_Reduced GNB_Reduced SVM_Reduced LDA_Reduced GLM_PCA
## GLM_Reduced                0.0328986  -0.0153623  -0.0086473  -0.0173913
## GNB_Reduced  1.00000                  -0.0482609  -0.0415459  -0.0502899
## SVM_Reduced  1.00000     0.13386                   0.0067150  -0.0020290
## LDA_Reduced  1.00000     1.00000     1.00000                  -0.0087440
```

```
## GLM_PCA      1.00000     0.34394     1.00000     1.00000
## GNB_PCA      1.00000     1.00000     1.00000     1.00000     1.00000
## SVM_PCA      0.81647     0.05915     1.00000     1.00000     1.00000
## GLM_LDA_Comp 1.00000     1.00000     1.00000     1.00000     1.00000
## GNB_LDA_Comp 1.00000     0.16423     1.00000     1.00000     1.00000
## SVM_LDA_Comp 1.00000     0.07574     1.00000     1.00000     1.00000
##              GNB_PCA    SVM_PCA    GLM_LDA_Comp GNB_LDA_Comp SVM_LDA_Comp
## GLM_Reduced   0.0242995 -0.0285507 -0.0199517   -0.0221685   -0.0197585
## GNB_Reduced  -0.0085990 -0.0614493 -0.0528502   -0.0587225   -0.0526570
## SVM_Reduced   0.0396618 -0.0131884 -0.0045894   -0.0099302   -0.0043961
## LDA_Reduced   0.0329469 -0.0199034 -0.0113043   -0.0025765   -0.0111111
## GLM_PCA       0.0416908 -0.0111594 -0.0025604   -0.0052067   -0.0023671
## GNB_PCA                 -0.0528502 -0.0442512   -0.0416532   -0.0440580
## SVM_PCA      0.35523                0.0085990    0.0095545    0.0087923
## GLM_LDA_Comp 0.98099     1.00000                 0.0024155    0.0001932
## GNB_LDA_Comp 1.00000     1.00000     1.00000                  0.0026302
## SVM_LDA_Comp 1.00000     1.00000     1.00000      1.00000
##
## Kappa
##              GLM_Reduced GNB_Reduced SVM_Reduced LDA_Reduced GLM_PCA
## GLM_Reduced               0.0727574  -0.0310455  -0.0129975  -0.0356699
## GNB_Reduced  0.86819                 -0.1038029  -0.0857548  -0.1084272
## SVM_Reduced  1.00000      0.10733                 0.0180480  -0.0046244
## LDA_Reduced  1.00000      1.00000     1.00000                -0.0226724
## GLM_PCA      1.00000      0.32346     1.00000     1.00000
## GNB_PCA      1.00000      1.00000     1.00000     1.00000     1.00000
## SVM_PCA      0.78052      0.05911     1.00000     1.00000     1.00000
## GLM_LDA_Comp 1.00000      1.00000     1.00000     1.00000     1.00000
## GNB_LDA_Comp 1.00000      0.20627     1.00000     1.00000     1.00000
## SVM_LDA_Comp 1.00000      0.08351     1.00000     1.00000     1.00000
##              GNB_PCA    SVM_PCA    GLM_LDA_Comp GNB_LDA_Comp SVM_LDA_Comp
## GLM_Reduced   0.0603594 -0.0586252 -0.0384418   -0.0427030   -0.0389275
## GNB_Reduced  -0.0123979 -0.1313826 -0.1111992   -0.1238031   -0.1116848
## SVM_Reduced   0.0914050 -0.0275797 -0.0073963   -0.0188340   -0.0078820
## LDA_Reduced   0.0733569 -0.0456278 -0.0254444   -0.0050273   -0.0259300
## GLM_PCA       0.0960293 -0.0229554 -0.0027720   -0.0083823   -0.0032576
## GNB_PCA                 -0.1189847 -0.0988013   -0.0931916   -0.0992869
## SVM_PCA      0.39734                0.0201834    0.0224362    0.0196978
## GLM_LDA_Comp 1.00000     1.00000                 0.0055245   -0.0004856
## GNB_LDA_Comp 1.00000     1.00000     1.00000                  0.0048287
## SVM_LDA_Comp 1.00000     1.00000     1.00000      1.00000
```