Big data analytics - Final report

B215284

10/12/23

Word count: 1068 Number of tables and figures: 6

Introduction and problem statement

Breast cancer, the most common cancer in women, also has the highest cancer-related mortality rate. Recent advances in the treatment of breast cancer have improved survival rates. The most common clinical question asked is prognosis and survival rates of the cancer. The survival rates are also helpful to assess the advantages recent advances in the cancer treatment and efficacy of the policy making, screen and healthcare spending. Current prognostication of breast cancer is typically based on the patient's clinical characteristics, tumor stage and histology, and treatment type. Factoring in the genetic data, can we predict the survival outcomes more accurately?

The data

Source

Collected by UK and Canada researchers collaborate, and the data hosted at cBioPortal.

Contents

Information about around 1980 breast cancer patients.

1. The first 31 columns contain clinical details and survival information.
2. The next 331 columns contain m-RNA levels reflecting gene expression by the tumor. The values are standardized and entered as z scores.
3. The next 175 columns contain information of gene mutations.

Goal of the project

The aim of this project is to develop a statistical model for prediction of survival outcome of a patient with given clinical and general sequencing information.

The data is explored with tables and graphs to arrive at initial impressions. The data is then prepared to model fitting by filling the missing values.

The model will be evaluated with metrics like accuracy, area under the curve and True & false positive/negative rates.

Data analysis

Importing libraries

```
library(sparklyr)
library(dplyr)
library(ggplot2)
library(knitr)
library(corrr)
library(dbplot)
library(reshape)
```

Creating a spark connection

The next block creates a spark connection to the master instance of the clusters. This Spark master node is responsible for delegating the computation to the executor nodes. The data is imported as a spark table with the spark_read_csv function. The connection acts as an SQL database. The dplyr operation acts as SQL

operations under the hood. Most of the operations are carried out on the spark table. The data is imported as R tibble when absolutely necessary.

```
sc <- spark_connect(
  master="spark://spark.eidf071:7077",
)

project_path <- '/work/eidf071/eidf071/shared/Projects/Breast cancer gene expression/'

metabric_RNA_data <- spark_read_csv(
  sc, name='diabetes_data',
  path=sprintf('file://%s/METABRIC_RNA_Mutation.csv', project_path),
)
```

Data overview

```
# number of rows
count(metabric_RNA_data)
```

```
## # Source: spark<?> [?? x 1]
##       n
##   <dbl>
## 1  1904
```

```
# number of columns
ncol(metabric_RNA_data)
```

```
## [1] 693
```

Data summary

```
# glimpse of data. printing first 10 columns
metabric_RNA_data %>%
  select(1:10) %>%  # select only first 10 columns
  glimpse()
```

```
## Rows: ??
## Columns: 10
## Database: spark_connection
## $ patient_id                <int> 0, 2, 5, 6, 8, 10, 14, 22, 28, 35, 36, 39, 4~
## $ age_at_diagnosis          <dbl> 75.65, 43.19, 48.87, 47.68, 76.97, 78.77, 56~
## $ type_of_breast_surgery    <chr> "MASTECTOMY", "BREAST CONSERVING", "MASTECTO~
## $ cancer_type               <chr> "Breast Cancer", "Breast Cancer", "Breast Ca~
## $ cancer_type_detailed      <chr> "Breast Invasive Ductal Carcinoma", "Breast ~
## $ cellularity               <chr> NA, "High", "High", "Moderate", "High", "Mod~
## $ chemotherapy              <int> 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1,~
## $ pam50__claudinlow_subtype <chr> "claudin-low", "LumA", "LumB", "LumB", "LumB~
## $ cohort                    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ er_status_measured_by_ihc <chr> "Positve", "Positve", "Positve", "Positve", ~
```

Number of missing values

```
missing_values <- metabric_RNA_data %>%
  mutate_all(is.na) %>%   # convert all the values to 1 or 0 based on if NA or not
  mutate_all(as.numeric) %>%    # convert them to numerical
```

```
  summarise_all(sum) %>%   # sum of all the na (or 1s if they are na)
  collect()  # concert to tibble

# transposing the tibble
transposed_data <- data.frame(t(missing_values),row.names = names(missing_values))

transposed_data %>%
  arrange(desc(t.missing_values.)) %>% # arranging by descending order of missing values
  head(15) %>%  # print first 15 columns with most missing values
  mutate(percentage = round(t.missing_values. / 1904 * 100, 2)) %>%
  kable(caption = "Summary of missing vaules")
```

Table 1: Summary of missing vaules

	t.missing_values.	percentage
tumor_stage	501	26.31
3gene_classifier_subtype	204	10.71
primary_tumor_laterality	106	5.57
neoplasm_histologic_grade	72	3.78
cellularity	54	2.84
mutation_count	45	2.36
er_status_measured_by_ihc	30	1.58
type_of_breast_surgery	22	1.16
tumor_size	20	1.05
cancer_type_detailed	15	0.79
tumor_other_histologic_subtype	15	0.79
oncotree_code	15	0.79
death_from_cancer	1	0.05
patient_id	0	0.00
age_at_diagnosis	0	0.00

The tumor stage, 3gene classification type and primary tumor laterality are the most common type of missing information. Rest of the columns have percentage of missing values less than 5%.

Explorative data analysis

Relationship between the clinical attributes and the outcome

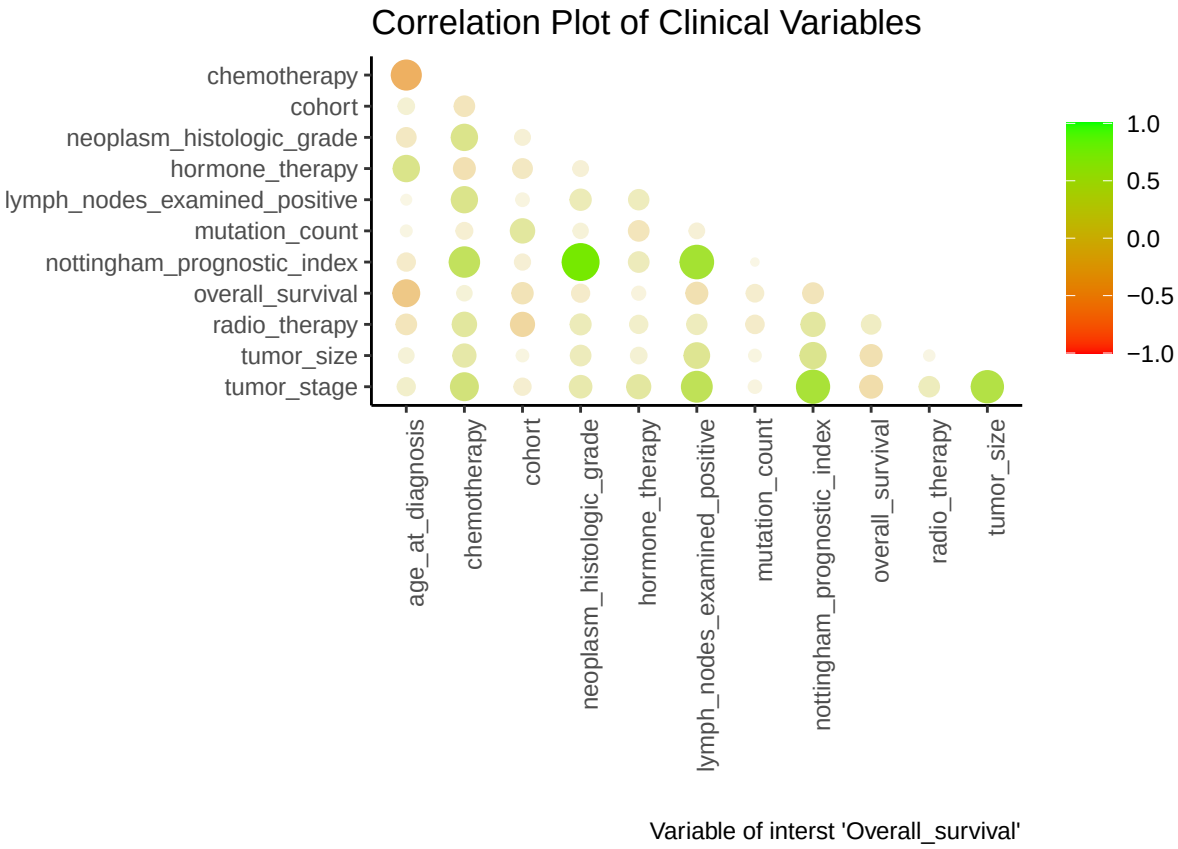
```
# Caching the clinical info in to a new spark table
clinical_info <- metabric_RNA_data %>%
  select(1:31) %>%
  compute('clinical_info')
```

The caching is creating another spark data frame with memory across the cluster nodes. The cached data frame acts similar to the spark data frame with same connection.

Correlation plot

```
corr_data <- clinical_info %>%
  collect() # Converting to R tibble
```

```
corr_data %>%
  # selecting only numerical columns. I tried to included all the variables, but hitting some errors
  select_if(is.numeric) %>%
  # patient id is not relevant. survival month is one of the outcome, so not to be used to correlate
  select(-patient_id, -overall_survival_months) %>%
  correlate(method = "pearson") %>%
  shave() %>%  # remove the half of the correlation matrix
  rplot(shape = 19, colors = c("red", "green"), legend = TRUE) +
  ggtitle("Correlation Plot of Clinical Variables") +
  labs(caption = "Variable of interst 'Overall_survival'",
       fill = "Correlation") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



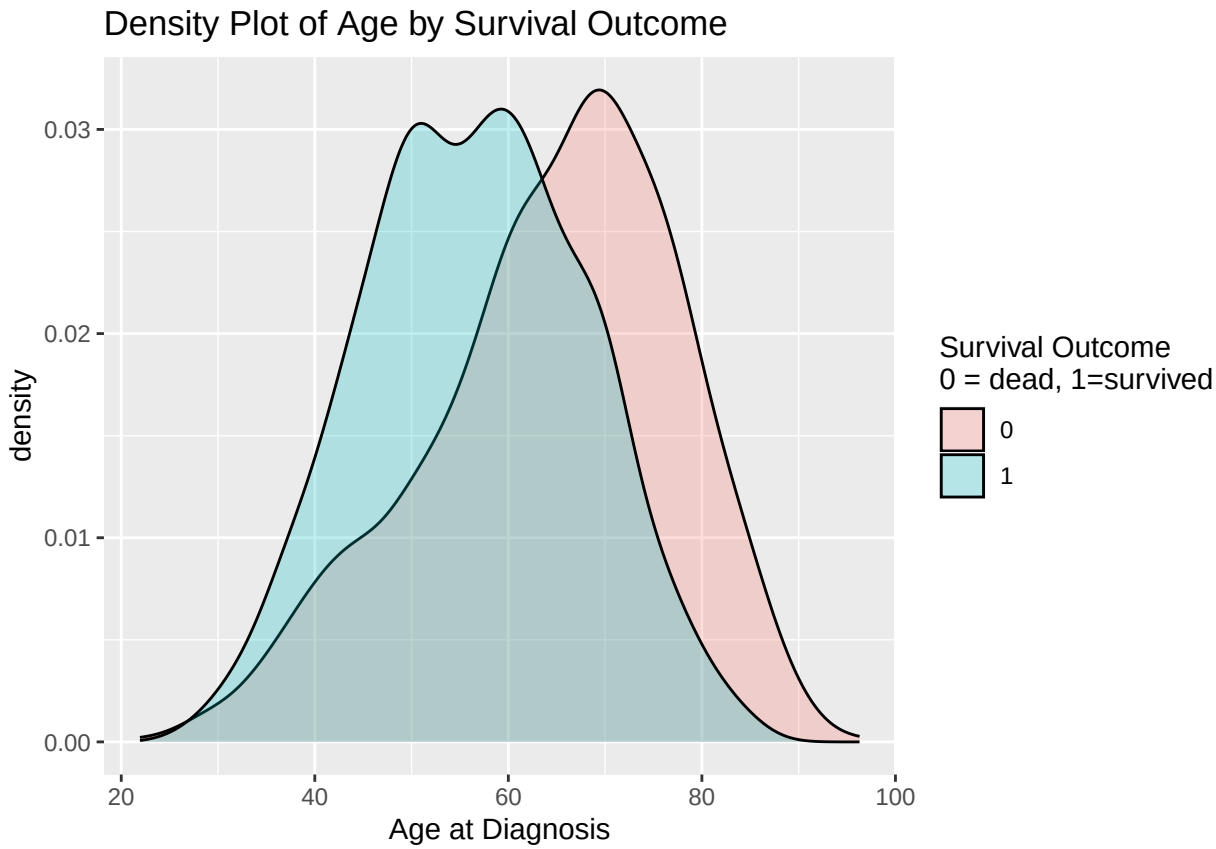
The survival is negatively correlated with age at diagnosis, number of positive lymph nodes, Nottingham prognostic index, tumour size, and tumour stage.

Age

```
age_data <- clinical_info %>%
  select(age_at_diagnosis, overall_survival) %>%
  collect()

age_data %>%
  mutate(overall_survival = as.factor(overall_survival)) %>%
  melt() %>%
  ggplot(aes(x=value, group=overall_survival, fill=overall_survival)) +
  geom_density(alpha=.25) +
```

```
  labs(
    title = "Density Plot of Age by Survival Outcome",
    x = "Age at Diagnosis",
    fill = "Survival Outcome \n0 = dead, 1=survived")
```



Age is an important predictor of survival in most of the medical conditions. Age affects the survival outcomes in various ways including clinical decision of offering radical treatment or with underlying co-morbidities.

The range age range of these two groups appears same but the density plot of the deceased group is negatively skewed.

mRNA expression

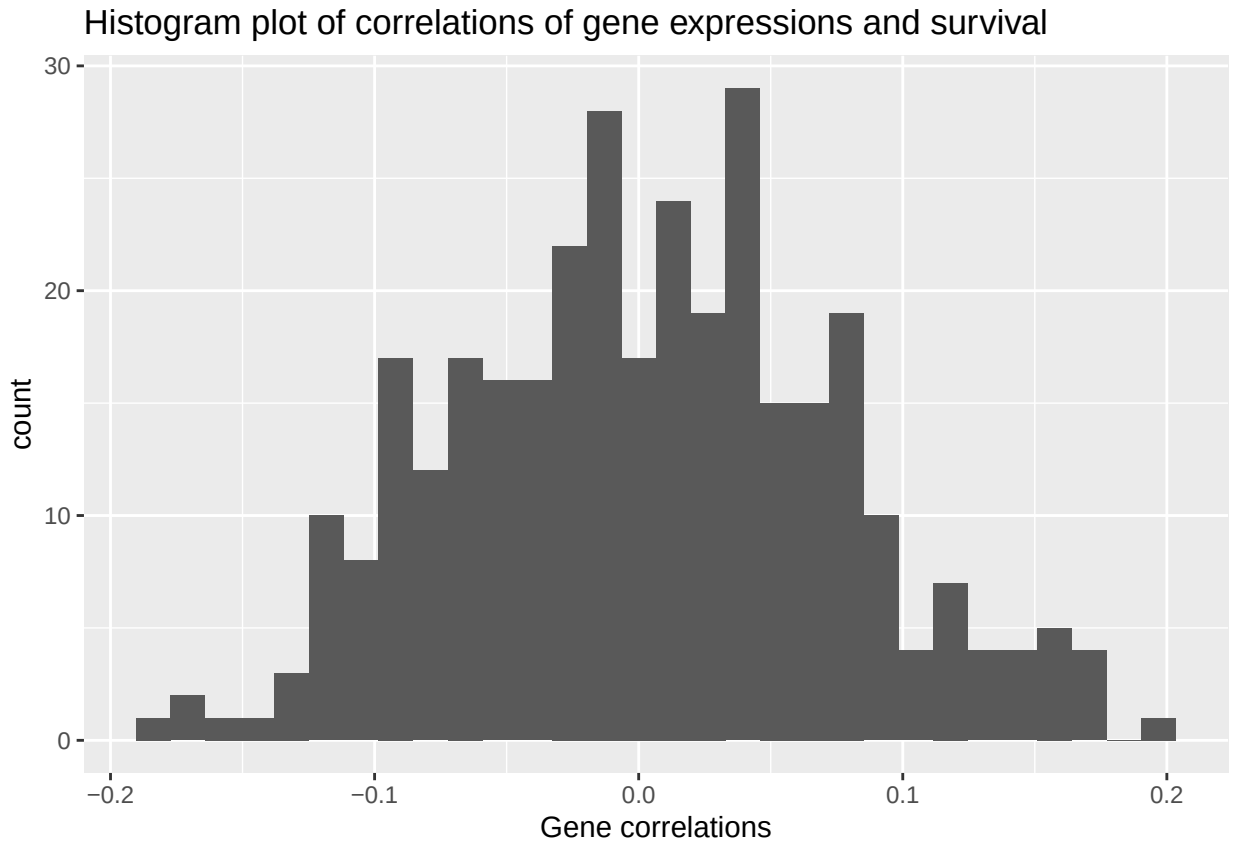
```
mrna_data <- metabric_RNA_data %>%
  select(25, 32:362) %>%
  compute('mrna_data')
```

```
mrna_corr <- mrna_data %>%
  ml_corr() %>%
  slice(1) %>%
  collect()

mrna_corr <- mrna_corr %>%
  pivot_longer(cols = -overall_survival, names_to = "Variable", values_to = "Value") %>%
  select(-overall_survival) %>%
  arrange(desc(Value))

mrna_corr %>%
```

```
ggplot(aes(x=Value)) +
geom_histogram() +
labs(
  title = "Histogram plot of correlations of gene expressions and survival",
  x = "Gene correlations")
```



The plot shows most of the mRNA carry a weak correlation with survival outcomes ranging from 0.19 to –0.15.

Most and least correlated genes

Now getting the most and least correlated genes.

```
mrna_corr %>% head(5)
```

```
## # A tibble: 5 x 2
##   Variable Value
##   <chr>    <dbl>
## 1 jak1     0.194
## 2 casp8    0.168
## 3 tgfbr2   0.166
## 4 abcb1    0.165
## 5 kit      0.164
```

```
mrna_corr %>% tail(5)
```

```
## # A tibble: 5 x 2
##   Variable  Value
##   <chr>     <dbl>
```

```
## 1 pdgfb    -0.145
## 2 tsc2     -0.162
## 3 map4     -0.165
## 4 kmt2c    -0.172
## 5 gsk3b    -0.186
```

Data cleaning

Due to limited computational resources and time for model fitting, a selected number of variables are chosen for model fitting.

1. Age, lymph nodes, Nottingham index, and tumor size - have been identified through correlation analysis.
2. Additionally, the topic of mastectomy versus breast-conserving surgery has been widely debated recently and thus included.
3. Finally, two genes of most positive and negative correlations.

The missing values of the selected variables can be handled as

1. Dropping the rows with the missing values
2. Dropping the column containing the missing values
3. Replacing the missing values as new category as "missing value" as analysis
4. Impute the values with mean, mode or advanced algorithms like K-nearest neighbour (KNN) imputation

In this notebook, the tumor size will be replaced with mean value and the type of surgery is replaced with mode.

```
# getting the most common type of surgery
metabric_RNA_data %>%
  count(type_of_breast_surgery)
```

```
## # Source: spark<?> [?? x 2]
##   type_of_breast_surgery     n
##   <chr>                  <dbl>
## 1 MASTECTOMY              1127
## 2 BREAST CONSERVING        755
## 3 <NA>                      22
```

```
# getting the mean tumor size
metabric_RNA_data %>%
  summarise(mean_size = mean(tumor_size))
```

```
## # Source: spark<?> [?? x 1]
##   mean_size
##       <dbl>
## 1      26.2
```

Imputation

```
metabric_RNA_data <- metabric_RNA_data %>%
  # replacing the missing values of tumor size with 26.2
  mutate(tumor_size = ifelse(is.na(tumor_size), 26.2, tumor_size)) %>%
  # replacing the missing values of surgery with mastectomy
  mutate(type_of_breast_surgery = ifelse(is.na(type_of_breast_surgery),
```

```
                                          "MASTECTOMY",
                                          type_of_breast_surgery))
```

Creating a table containing only the columns of intrest

```
metabric_RNA_data <- metabric_RNA_data %>%
  select(overall_survival,
         age_at_diagnosis,
         lymph_nodes_examined_positive,
         nottingham_prognostic_index,tumor_size,
         type_of_breast_surgery,
         jak1,
         pdgfb) %>%
  # converting the surgery variable into numerical. only one dummy variable is retained
  mutate(surgery_mastectomy = ifelse(type_of_breast_surgery == "MASTECTOMY", 1, 0)) %>%
  select(-type_of_breast_surgery) # dropping the original surgery column which is not required
```

Model fitting

A logistic regression model is used to fit binary outcomes.

Another model, a generalized linear model is also fitted. The model estimates coefficients for each variable. Exponential of these coefficients provides the odds ratio.

```
logistic_model <- ml_logistic_regression(metabric_RNA_data,
                                         overall_survival ~ age_at_diagnosis +
                                           lymph_nodes_examined_positive +
                                           nottingham_prognostic_index +
                                           tumor_size +
                                           surgery_mastectomy +
                                           jak1 +
                                           pdgfb)

gen_linear_model <- ml_generalized_linear_regression(metabric_RNA_data,
                                         overall_survival ~ age_at_diagnosis +
                                           lymph_nodes_examined_positive +
                                           nottingham_prognostic_index +
                                           tumor_size +
                                           surgery_mastectomy +
                                           jak1 +
                                           pdgfb,
                                         family = "binomial")
```

Model evaluation

1. Logistic model

```
validation_summary <- ml_evaluate(logistic_model, metabric_RNA_data)
print(paste("Model accurarcy: ", validation_summary$accuracy()))
```

```
## [1] "Model accurarcy:  0.69327731092437"
```

```
print(paste("Model Area under the curve: ", validation_summary$area_under_roc()))
```

```
## [1] "Model Area under the curve:  0.755381702156076"
```

"Accuracy" is how many times the model correctly predicts out of total predictions. it is equal to the number of true positives and true negatives over the total predictions. "The area under the curve" is a metric that evaluates how well the model classifies the positive and negative outcomes at the best cutoff probability values. The value is 0.75, when it is closer to 1, the better the model performance.

```
# TP, FP, precision and recall
validation_summary$true_positive_rate_by_label()
```

```
## [1] 0.7796917 0.5742821
```

```
validation_summary$false_positive_rate_by_label()
```

```
## [1] 0.4257179 0.2203083
```

```
validation_summary$precision_by_label()
```

```
## [1] 0.7160699 0.6543385
```

```
validation_summary$recall_by_label()
```

```
## [1] 0.7796917 0.5742821
```

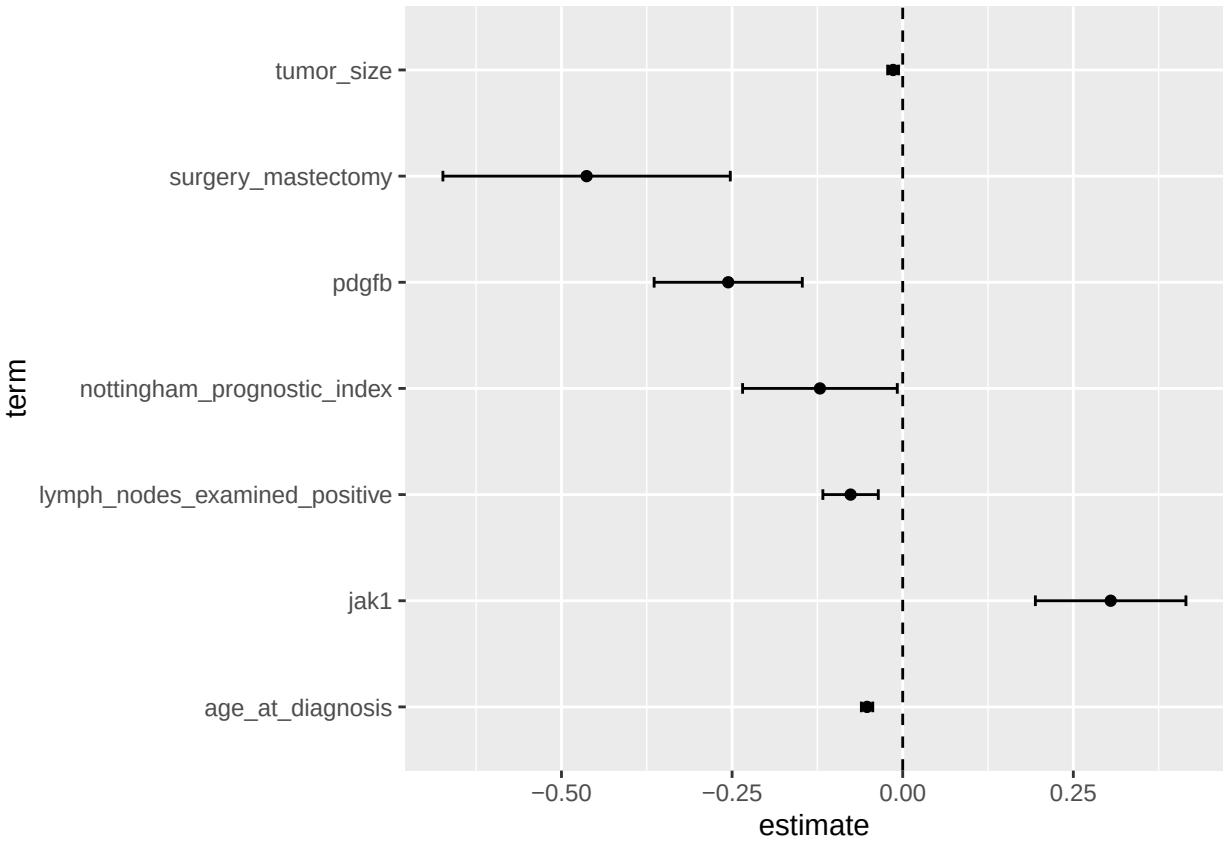
The above output shows the true-positive rates, false-positive rates precision and recall by the labels 0 and 1 (dead and survived). "Precision" is a measure of how many times the model correctly predicts the positive instances among all predicted positives (Precision = TP /(TP + FP)). "Recall" also known as sensitivity, is the measure of how many times the model correctly predicts the positive instances among all the underlying positives (Recall = TP /(TP + FN))

2. Generalised linear regression model

The linear model output is continuous and does not provide class probabilities because the coefficients are fit for a linear equation. To generate a coefficient plot, we can retrieve the coefficient estimates and their respective standard errors.

```
gen_linear_model <- tidy(gen_linear_model)


gen_linear_model %>%
  slice_tail(n=-1) %>%
  ggplot(aes(x = term, y = estimate)) +
  geom_point() +
  geom_errorbar(
    aes(ymin = estimate - 1.96 * std.error,
        ymax = estimate + 1.96 * std.error, width = .1)
  ) +
  coord_flip() +
  geom_hline(yintercept = 0, linetype = "dashed")
```



Please note that the mastectomy is associated with negative survival outcomes. This standalone information is not enough to conclude mastectomy causes less survival. We have not investigated the co-linearity between the dependent variables and have not included other relevant variables. There is a possibility that the patients with poorer clinical conditions are offered mastectomy over breast conservative surgery rather than mastectomy causing the increased mortality.

Closing the spark connection

```
spark_disconnect(sc)
```

Conclusion

The logistic model here shows good accuracy and area under the curve (AUC) reflecting a good fit. However, the model has some shortcomings. The model holds good only for this data and we have to be cautious about generalising to apply to different data. As mentioned above, the model is blind to other variables which be confounding and may predict erroneous outcomes. The address this, stratified modelling can be considered where one variable is evaluated with other variables kept constant. Another approach can be a prospective cohort of breast cancer patients to evaluate the effect of gene expressions, when ethically feasible.

Cancers are caused by gene mutations. Gene expressions affect the ability of a cancer to metastasis or its response to treatment like chemotherapy.

As a data scientist, we can gain insights of cancer genomics and understand the underlying mechanisms by analysing large-scale genomic datasets. Assisted by statistical models, we can identify biomarkers to predict the treatment response or help in developing therapies to prevent early metastasis, diease monitoring or early cancer detection.