Milestone2_Feature_Selection

Nosipho Precious Donkrag, Nontsikelelo Sharon Buhlungu, Tshepang Mogosi, Pitsi Pitsi

2024-10-13

Milestone2_Feature_Selection

Aim: - Decide which columns to keep for model training; - Ensure all columns sent to the ml model are of numeric nature.

```
setwd("C:/Users/nosip/Documents/third Year/BIN381/milestones")
```

```
library(readr)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggcorrplot)
library(fastDummies)
library(corrplot)
```

```
## corrplot 0.94 loaded
```

Read in data:

```
data_df <- read_csv("cleaned_cust.csv", show_col_types = FALSE)
```

```
names(data_df)
```

```
##  [1] "marital_status"   "street_address"   "postal_code"      "city"
##  [5] "state_province"   "Country_id"       "phone_number"     "email"
##  [9] "Education"        "Occupation"       "household_size"   "yrs_of_residence"
## [13] "Annual_Salary"    "Months_Annual"    "FRS.Contribution" "Year_of_Birth"
## [17] "Net_Salary"       "Net_months"       "Gross_Salary"     "Gross_Months"
## [21] "Qualify"
```

Cardinality

High cardinality refers to columns that have too many unique values; machine learning models can not be trained on such data as this data may cause Over-fitting, increase in dimensions, data leakage (when the

model gains access to data used for testing or validation during training).

High cardinal columns: Columns to remove

- street_address
- postal_code
- city
- state_province
- phone_number
- email
- Country_id These columns will not be added to the data sent to the ml model.

```
columns_to_exclude <- c("street_address", "postal_code", "city",
                        "state_province", "phone_number", "email", "Country_id")


data_for_ml <- data_df[ , !(names(data_df) %in% columns_to_exclude)]
```

```
head(data_for_ml)
```

```
## # A tibble: 6 x 14
##   marital_status Education Occupation household_size yrs_of_residence
##            <dbl> <chr>     <chr>               <dbl>            <dbl>
## 1              1 Masters   Prof.                   2                4
## 2              2 Masters   Prof.                   2                4
## 3              2 Masters   Prof.                   2                4
## 4              1 Masters   Prof.                   2                4
## 5              2 Masters   Prof.                   2                4
## 6              2 Masters   Prof.                   2                4
## # i 9 more variables: Annual_Salary <dbl>, Months_Annual <dbl>,
## #   FRS.Contribution <dbl>, Year_of_Birth <dbl>, Net_Salary <dbl>,
## #   Net_months <dbl>, Gross_Salary <dbl>, Gross_Months <dbl>, Qualify <dbl>
```

The following columns contain non-numeric data and this data will be transformed.

```
non_numeric_columns <- sapply(data_for_ml, is.character) | sapply(data_for_ml, is.factor)

non_numeric_data <- data_for_ml[ , non_numeric_columns]

print(colnames(non_numeric_data))
```

```
## [1] "Education"  "Occupation"
```

Leaky Feature: Occupation

```
unique_occupation <- unique(data_for_ml$Occupation)
unique_occupation
```

```
## [1] "Prof."   "Masters" "Sales"   "Bach."   "Cleric." "HS-grad" "Exec."
```

Data from education leaked into the Occupation column, this data must be removed.

```
values_to_drop <- c("Masters", "Bach.", "HS-grad")
data_for_ml <- data_for_ml[!data_for_ml$Occupation %in% values_to_drop, ]
```

```
occupation_counts <- table(data_for_ml$Occupation)

non_unique_occupation <- names(occupation_counts[occupation_counts > 1])
```

```r
print(non_unique_occupation)
```

```
## [1] "Cleric." "Exec."   "Prof."   "Sales"
```

Education

Education still contains some inconsistent data; this is because the data type is character so when it was null (no education); the inconsistent data leaked into this column.

```r
#unique values in Education
unique_education <- unique(non_numeric_data$Education)
print(unique_education[1:10])
```

```
##  [1] "Masters"               "Kitchens@company.com"  "Drumm@company.com"
##  [4] "Hanes@company.com"     "Ziegler@company.com"   "Evans@company.com"
##  [7] "Tien@company.com"      "Jewell@company.com"    "Lent@company.com"
## [10] "Salvadore@company.com"
```

Education

```r
rows_to_drop <- grepl("\\.com$", data_for_ml$Education)
data_for_ml <- data_for_ml[!rows_to_drop, ]
```

```r
unique_education <- unique(data_for_ml$Education)
print(unique_education)
```

```
## [1] "Masters" "Bach."   "HS-grad"
```

Contingency Table:

```r
contingency_table <- table(data_for_ml$Education, data_for_ml$Qualify)

print(contingency_table)
```

```
##
##               0     1
##   Bach.   38635 25902
##   HS-grad 25779 17228
##   Masters 26149 17440
```

Majority of qualifying customers hold a Bachelor's degree.

Data Transformation

Data Transformation: Education

Covert the column education to numeric by one hot encoding:

```r
library(fastDummies)
data_for_ml <- dummy_cols(data_for_ml,
                          select_columns = "Education",
                          remove_first_dummy = FALSE,
                          remove_selected_columns = TRUE)
```

Data Transformation: Occupation

To change this column into a numeric column One-hot Encoding will be utilized:

```r
# Apply one-hot encoding to the 'Occupation' column
data_for_ml <- dummy_cols(data_for_ml,
                          select_columns = "Occupation",
                          remove_first_dummy = FALSE,
                          remove_selected_columns = TRUE)
```

```r
colnames(data_for_ml)
```

```
##  [1] "marital_status"     "household_size"     "yrs_of_residence"
##  [4] "Annual_Salary"      "Months_Annual"      "FRS.Contribution"
##  [7] "Year_of_Birth"      "Net_Salary"         "Net_months"
## [10] "Gross_Salary"       "Gross_Months"       "Qualify"
## [13] "Education_Bach."    "Education_HS-grad"  "Education_Masters"
## [16] "Occupation_Cleric." "Occupation_Exec."   "Occupation_Prof."
## [19] "Occupation_Sales"
```

Feature Engineering: Replace Years of Birth with Age

```r
current_year <- as.numeric(format(Sys.Date(), "%Y"))

data_for_ml$age <- current_year - data_for_ml$Year_of_Birth

# Remove the Year_of_Birth
data_for_ml$Year_of_Birth <- NULL
```

```r
head(data_for_ml)
```

```
## # A tibble: 6 x 19
##   marital_status household_size yrs_of_residence Annual_Salary Months_Annual
##            <dbl>          <dbl>             <dbl>         <dbl>         <dbl>
## 1              1              2                 4          620.             2
## 2              2              2                 4          250.             3
## 3              2              2                 4          394.             4
## 4              1              2                 4          735.             1
## 5              2              2                 4          386.             6
## 6              2              2                 4          621.             3
## # i 14 more variables: FRS.Contribution <dbl>, Net_Salary <dbl>,
## #   Net_months <dbl>, Gross_Salary <dbl>, Gross_Months <dbl>, Qualify <dbl>,
## #   Education_Bach. <int>, `Education_HS-grad` <int>, Education_Masters <int>,
## #   Occupation_Cleric. <int>, Occupation_Exec. <int>, Occupation_Prof. <int>,
## #   Occupation_Sales <int>, age <dbl>
```

Correllation Matrix

The correlation matrix will be used to determine which columns to keep:

```r
cor_matrix <- cor(data_for_ml, use = "complete.obs")
```

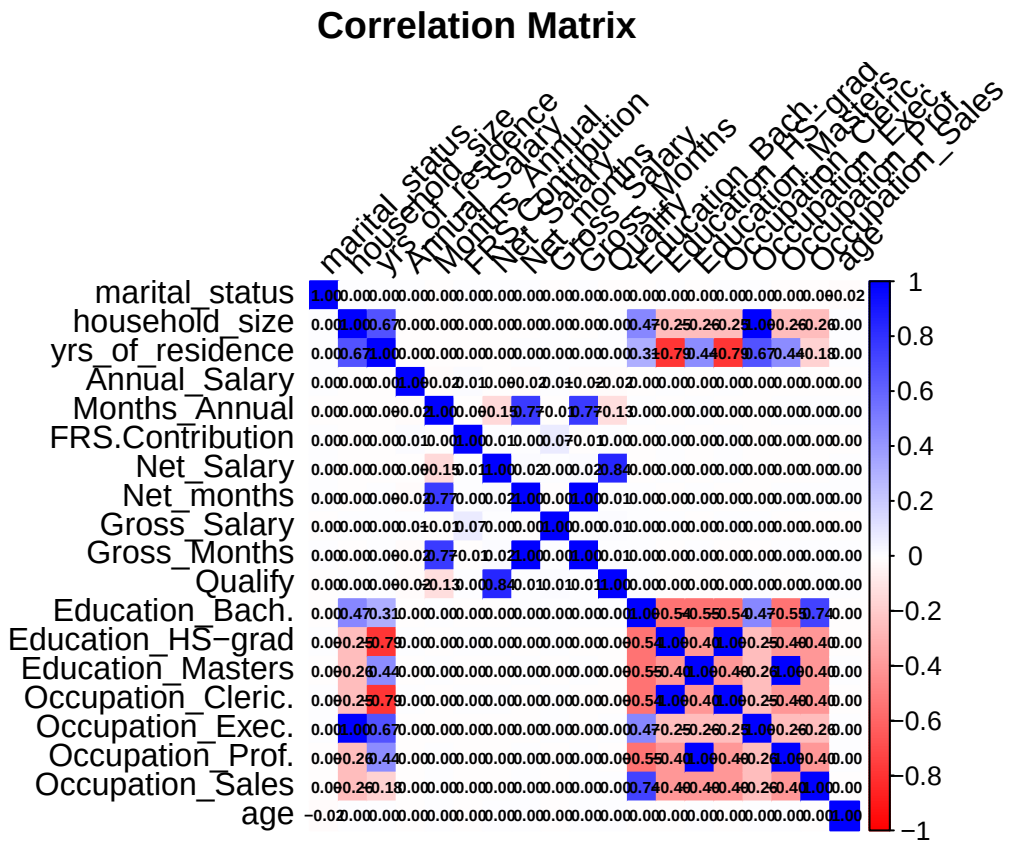
Extract the correlation with the target variable:

```r
qualify_correlations <- cor_matrix[, "Qualify"]

#filter
high_correlations <- qualify_correlations[abs(qualify_correlations) > 0.2]
```

create a correlation matrix with the target:

```r
# Create the correlation plot
corrplot(cor_matrix,
         method = "color",          # Use color for the correlation coefficients
         type = "full",             # Show the entire matrix
         tl.col = "black",          # Color of the text labels
         tl.srt = 45,               # Rotate text labels
         addCoef.col = "black",     # Add correlation coefficients in black
         number.cex = 0.5,          # Increase size of the coefficient numbers
         col = colorRampPalette(c("red", "white", "blue"))(200),  # Color gradient
         title = "Correlation Matrix", # Set title
         mar = c(0, 0, 2, 0)        # Margins for the plot
)
```



The education columns seem correlated with the occupation columns; however the target variable does not seem correlated with any of the columns, thus it would be risky to remove any columns as of yet. marital status is slightly positively correlated with age. The older the person the more likeliness of them being married.

the data has been prepared for the ml algorithm, the different algorithms that will be implemented will shed light on the feature importance that exists within the dataset.

#Save dataset

```r
write.csv(data_for_ml, file = "data_for_ml.csv", row.names = FALSE)
```