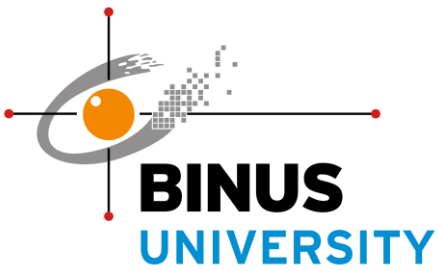
PROJECT REPORT

LAB ASSIGNMENT DATA MINING AND VISUALISASI

By:

2502055204 - Elena Ghini Rachman



Even Semester 2022/2023, class:
BA09 – Lab
LA09 - Lec

PRODI COMPUTER SCIENCE

FAKULTAS SCHOOL OF COMPUTER SCIENCE

UNIVERSITAS BINA NUSANTARA

JAKARTA

2502055204 – Elena Ghini Rachman

Below is the lab assignments for this semester, which you can submit at most on the week of the last session.

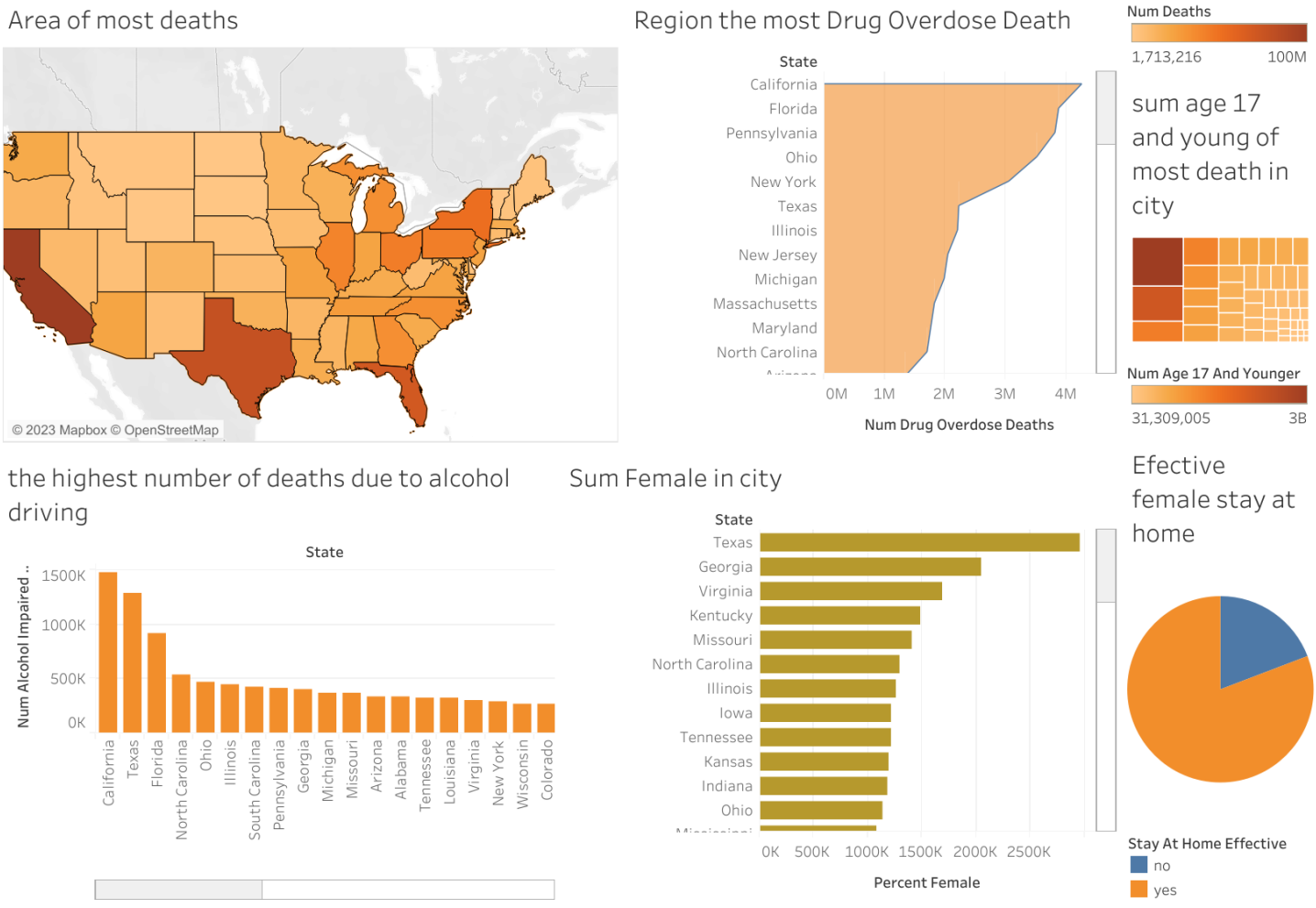
1. Download the US Counties: COVID19 + Weather + Socio/Health dataset (https://www.kaggle.com/datasets/johnjdavisiv/us-counties-covid19-weather-sociohealth-data?select=us_county_sociohealth_data.csv), perform the following tasks on this dataset which contain three separate CSV files!
   a. Perform Explanatory Data Analysis on the data!
   b. Visualize them using Tableau!
   Files to submit: R files (containing codes and comments on what each line of code do), report in pdf format containing screenshots of the Tableau dashboard.

Jawaban

a) Link github for code: https://github.com/elenaghini/US_Counties_COVID19_Weather_SocioHealth_data.git
b) Visualization with Tableau

The file contains 3 csv namely "us_county_sociohealth_data.csv", "us_county_geometry" and "US_counties_COVID19_health_weather_data.csv".

This report uses the CSV file “US_counties_COVID19_health_weather_data.csv” to visualize this data in Tableau. The following is the result of the visualization:



Link Tableau:
https://public.tableau.com/views/AOLDatmining/Dashboard1?:language=en-US&:display_count=n&:origin=viz_share_link.

2502055204 – Elena Ghini Rachman

2. Download the Telco Customer Churn dataset (https://www.kaggle.com/datasets/blastchar/telco-customer-churn), perform an Exploratory Data Analysis, and build a predictive model for customer churn!

   Files to submit: R files (containing codes and comments on what each line of code do), report in pdf format containing the evaluation results of the predictive model.

   You are encouraged to upload the files to your GitHub or other portfolios of your work, which will be a proof of your skills during your time as a student.
   All of the above tasks should be done individually! Any forms of plagiarism will not be tolerated!
   Good luck and GBU.

Jawaban

Link Github: https://github.com/elenaghini/Telco-Customer-Churn.git
The predictive model used is SVM (Support Vector Machine)

```
# Load the required libraries
> library(e1071)
>
> # Read file CSV
> df <- read.csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
>
> # Removed the irrelevant customerID column
> df <- df[, -1]
>
> # Fill in the missing values with the median
> df$TotalCharges[is.na(df$TotalCharges)] <- median(df$TotalCharges, na.rm = TRUE)
>
> # Changed the data type of the TotalCharges column to numeric
> df$TotalCharges <- as.numeric(df$TotalCharges)
>
> # Turning the dependent variable into a factor
> df$Churn <- as.factor(df$Churn)
>
> # Separation of datasets into features and labels
> X <- df[, -20]
> y <- df[, 20]
>
> # Encoding one-hot on categorical features
> X <- model.matrix(~.-1, data = X)
>
> # The division of the dataset into training sets and test sets
> set.seed(42)  # Untuk reproduktibilitas
> train_idx <- sample(1:nrow(X), 0.8 * nrow(X))
> X_train <- X[train_idx, ]
> y_train <- y[train_idx]
> X_test <- X[-train_idx, ]
> y_test <- y[-train_idx]
>
> # Train the SVM model for classification
> model <- svm(x = X_train, y = y_train)
>
> # Make predictions on the test set data
> y_pred <- predict(model, X_test)
>
> # Calculates prediction accuracy
> accuracy <- sum(y_pred == y_test) / length(y_test)
> print(paste("Akurasi:", accuracy))
[1] "Akurasi: 0.792760823278921"
```

In this case, the resulting accuracy is 0.792760823278921, or around 79.28%.
Such accuracy does not provide a complete picture of the model's performance. Consider other metrics such as precision, recall, F1-score, or confusion matrix for a clearer sense of model performance.

```
# Removed the irrelevant customerID column
df <- df[, -1]

# Fill in the missing values with the median
df$TotalCharges[is.na(df$TotalCharges)] <- median(df$TotalCharges, na.rm = TRUE)
```

2502055204 – Elena Ghini Rachman

```
# Changed the data type of the TotalCharges column to numeric
df$TotalCharges <- as.numeric(df$TotalCharges)

# Turning the dependent variable into a factor
df$Churn <- as.factor(df$Churn)

# Separation of datasets into features and labels
X <- df[, -20]
y <- df[, 20]

# Encoding one-hot on categorical features
X <- model.matrix(~.-1, data = X)

# The division of the dataset into training sets and test sets
set.seed(42)  # Untuk reproduktibilitas
train_idx <- sample(1:nrow(X), 0.8 * nrow(X))
X_train <- X[train_idx, ]
y_train <- y[train_idx]
X_test <- X[-train_idx, ]
y_test <- y[-train_idx]

# Train the SVM model for classification
svm_model <- svm(x = X_train, y = y_train)

# Make predictions on the test set data
y_pred <- predict(svm_model, newdata = X_test)

# Calculates prediction accuracy
accuracy <- sum(y_pred == y_test) / length(y_test)
print(paste("Akurasi:", accuracy))

# Calculating the confusion matrix
confusion <- confusionMatrix(data = y_pred, reference = y_test)

# Fetch evaluation metrics
precision <- confusion$byClass['Precision']
recall <- confusion$byClass['Recall']
f1_score <- confusion$byClass['F1']

# Showing results
cat("Precision:", precision, "\n")
cat("Recall:", recall, "\n")
cat("F1-score:", f1_score, "\n")
>
> # Calculating the confusion matrix
> confusion <- confusionMatrix(data = y_pred, reference = y_test)
>
> # Fetch evaluation metrics
> precision <- confusion$byClass['Precision']
> recall <- confusion$byClass['Recall']
> f1_score <- confusion$byClass['F1']
>
> # Dispalay
> cat("Precision:", precision, "\n")
Precision: 0.822807
> cat("Recall:", recall, "\n")
Recall: 0.9124514
> cat("F1-score:", f1_score, "\n")
F1-score: 0.8653137
>
```

Analysis:

1. Precision
   Positive predictive measures are true or not with Precision. Precision measures the extent to which the data labeled positive by the model are actually positive data. The results of the run that has been carried out, the precision is 0.822807, which means around 82.28%. This is measured correctly
2. Recall
   Recall is 0.9124514, which means the model is able to find around 91.25% of all positive cases in the data. Finding all positive cases in the data Recall (sensitivity) measures the extent to which the model is

able to find all positive cases in the data. Recall is important when we want to minimize the number of false negatives (cases where the model predicts a negative when it is actually positive).

3. F1-score
   Precision and recall are combined with the F1-score into one number that reflects the balance between the two. F1-score of 0.8653137 indicates the degree of harmony between precision and recall. The higher the F1-score, the better the model's performance in achieving a balance between precision and recall.

From this analysis, it can be concluded that the model tends to be better at finding positive cases than correctly predicting all existing positive cases. The model has fairly good accuracy (79.28%), but there is a significant difference between precision (82.28%) and recall (91.25%). An increase in precision can help reduce the number of false positives, while an increase in recall can help reduce the number of false negatives.