Introduction

In today's world of increased connections and movement across borders it is crucial for border control to facilitate the transit of goods, people and services. Border officials encounter the challenge of streamlining operations reducing wait times for travelers and adapting to changing conditions that impact traffic flow. Utilizing data mining techniques can help extract information from border traffic data to make decisions and strategically plan.

This document presents an analysis of border crossing data with a focus on addressing business queries related to border management and strategic planning. The analysis delves into identifying peak periods for border crossings examining how external factors affect traffic at borders and assessing the accuracy of models in forecasting traffic levels. By tackling these questions border authorities can gain insights into traffic trends optimize resource distribution and improve efficiency.

Analysis

Recognizing the peak volumes for border crossings holds importance for allocating resources and devising strategies. By focusing on traffic volume peaks, authorities can optimize staff and infrastructure deployment efficiently.

Data Description

The dataset 'Border Crossing Entry Data' consists of multiple variables including the type of vehicle, the number of crossings, and the border ports. It covers data from multiple years and provides a comprehensive view of border activities. Initial exploration revealed issues like missing values and duplicate records, which were addressed prior to the analysis.

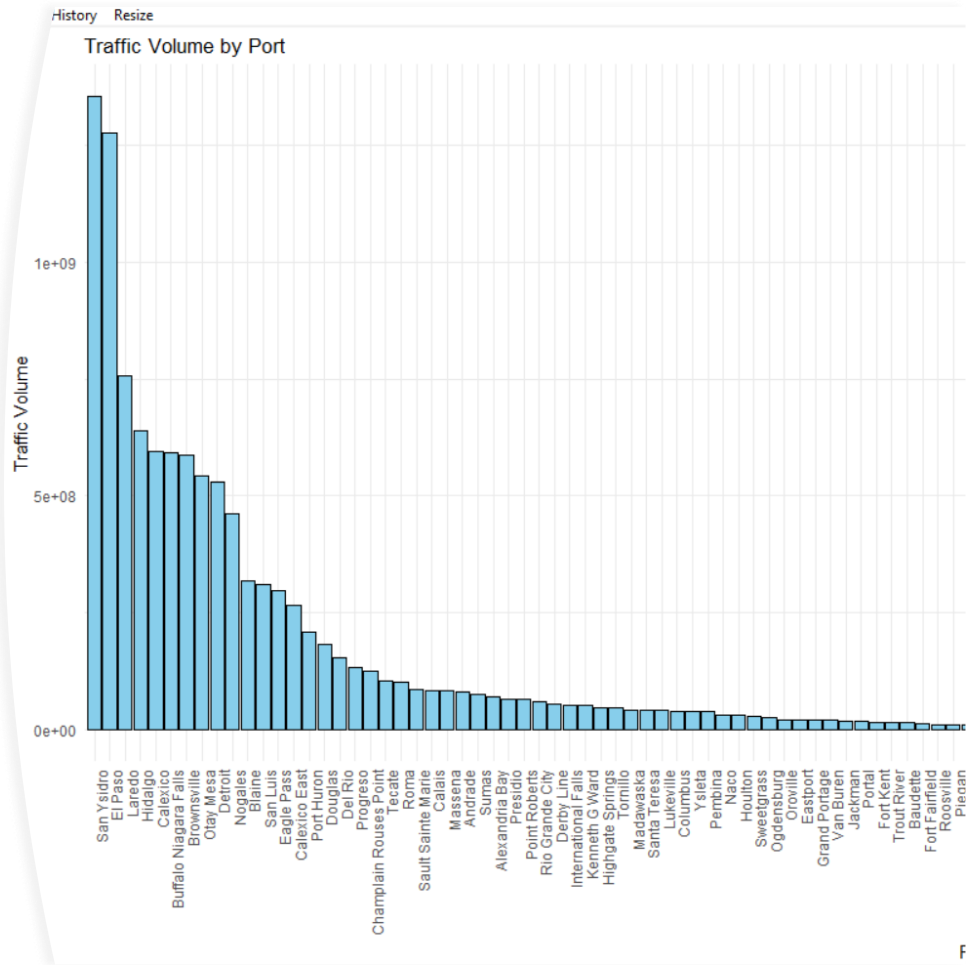
Impact of External Elements on Border Traffic

External variables like weather conditions, holidays, economic occurrences, and geopolitical scenarios can impact border traffic flow. Studying these factors can offer insights into their influence on traffic trends and aid in formulating management strategies to handle fluctuations.

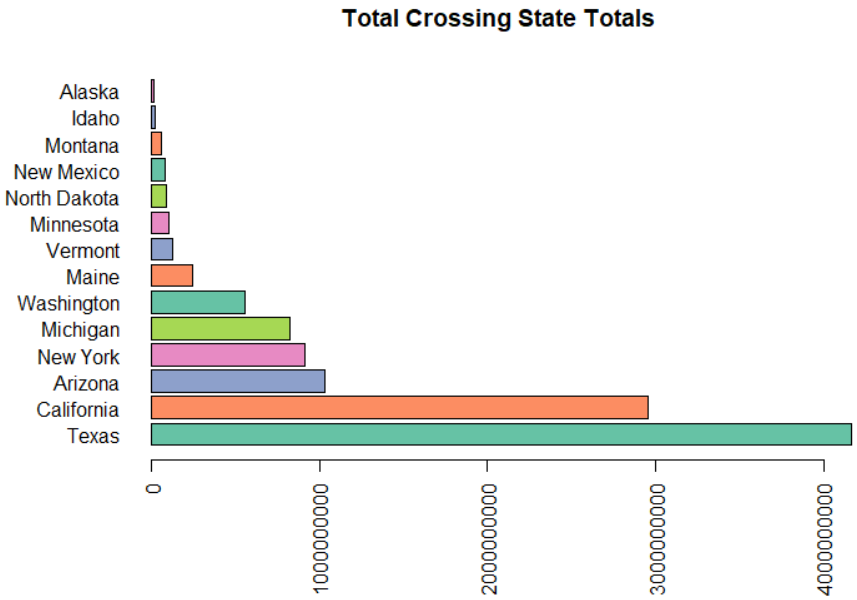
Reliability of Predictive Models in Projecting Future Traffic Levels

Crafting predictive models to forecast traffic volumes is vital for decision making and long-range planning. Evaluating the accuracy of these models is crucial to ascertain their dependability and usefulness in shaping border management tactics.

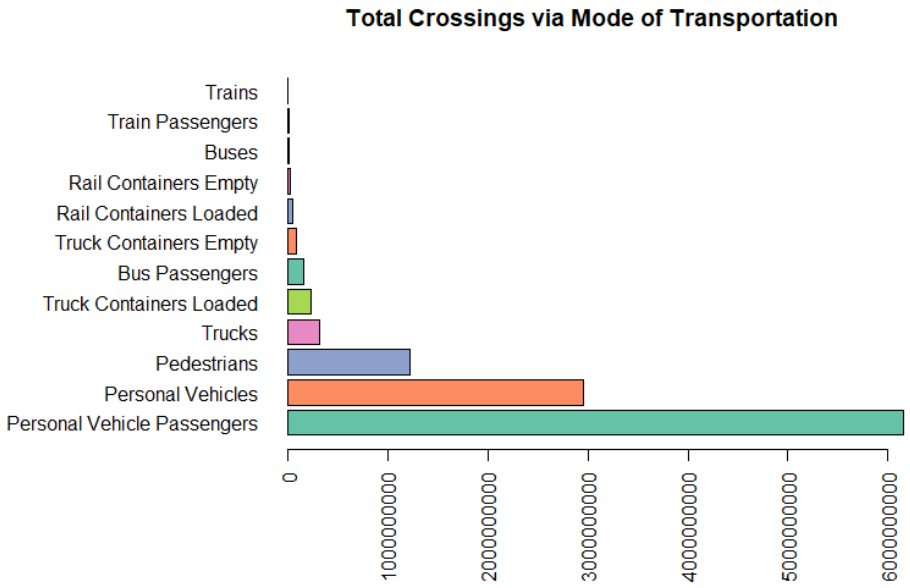
Busiest Border Crossings



As shown in later analysis, the southern, USA/Mexico border has the most traffic.



Most of the traffic comes in the form of passenger vehicles. The large volume of personal vehicles along with the drivers and passengers in those vehicles create challenges for regulators and opportunities for entrepreneurs. Each person is a potential customer.



Clustering Analysis

Data Preparation

To explore spatial patterns in the border crossing data, we selected three key variables: Latitude, Longitude, and Value (representing the volume of crossings). These variables were chosen to understand the geographical distribution and intensity of border crossings.

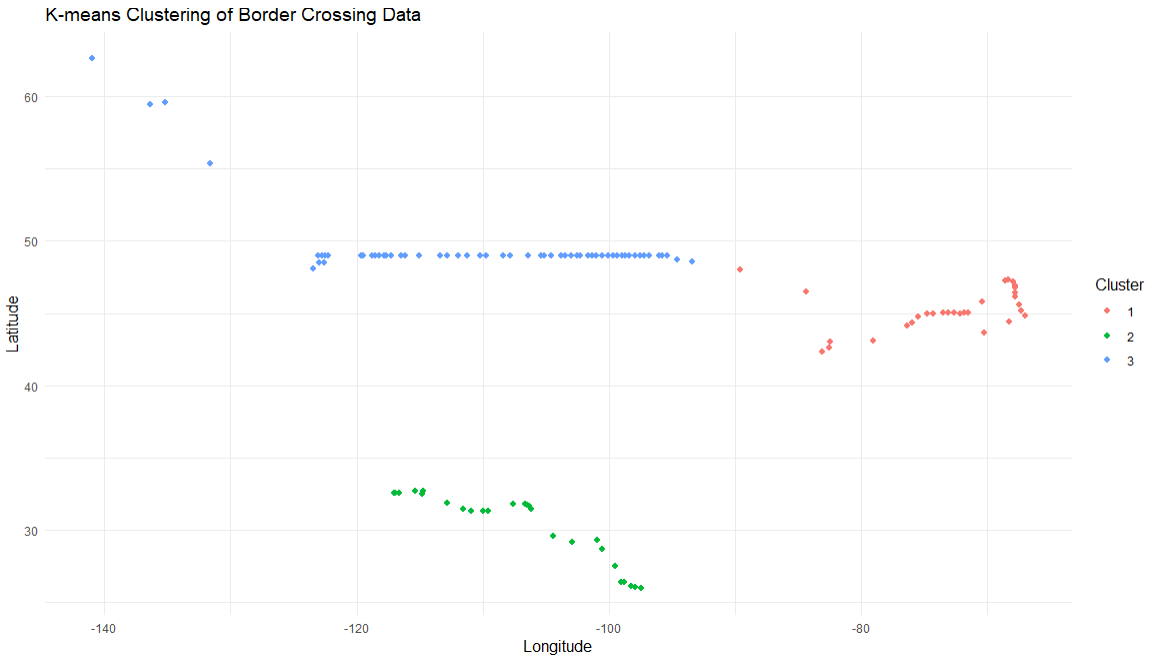
After selecting the relevant columns, the data were normalized. Normalization is crucial in clustering to ensure that each variable contributes equally to the analysis, preventing any variable with larger scale from dominating the distance calculation used in the clustering algorithm.

```
> cluster_data
    Latitude Longitude  Value
1     26.404   -99.019     46
2     29.327  -100.928   6552
3     49.000  -109.731      2
4     48.997  -111.258     29
5     31.673  -106.335 521714
6     48.999   -95.377    837
7     48.999  -110.215     20
8     48.999  -110.215    965
9     49.000  -101.017    102
10    48.999   -95.377    459
11    31.673  -106.335  21355
12    31.673  -106.335 127217
13    48.999  -110.215    519
14    48.999  -110.215     21
15    49.000  -101.017    339
16    48.999   -95.377    590
17    31.673  -106.335  63367
18    48.999  -110.215     10
19    48.998  -111.960   1985
20    25.952   -97.401   7712
21    49.002  -122.265    327
22    47.360   -68.329    130
23    59.451  -136.362     21
24    32.673  -115.388 435768
```

k-Means Clustering

We employed k-means clustering, a popular partitioning method that segments data into K distinct, non-overlapping clusters. The algorithm assigns each data point to the cluster with the nearest mean, which serves as the prototype of the cluster.

For this analysis, we chose to use three clusters (k = 3), based on initial assessments that suggested this number would adequately capture the main patterns in the data without overcomplicating the model. The number of starts (`nstart = 25`) was set to ensure the algorithm was run multiple times with different initial centroids, enhancing the likelihood of finding a good solution by avoiding local minima.



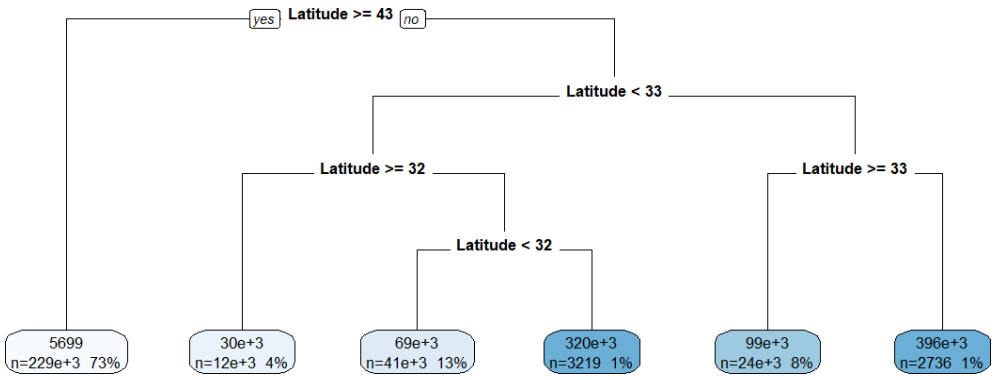
Visualizing Clusters

The results of the clustering were visualized using a scatter plot, with data points plotted according to their geographic coordinates (Longitude and Latitude) and colored by their assigned cluster. This visualization helps in understanding the spatial distribution of the clusters, indicating regions with similar characteristics in terms of border crossing volume and location. The ggplot2 package in R was utilized for creating an informative plot. The theme,

`theme_minimal()` was used to keep the visualization clean and focused on the data points and clusters.

The k-means clustering provided insights into the geographical grouping of border crossings, highlighting areas with high, medium, and low activities. This information can be crucial for resource allocation and operational planning by policymakers and border management authorities. By understanding these clusters, decision-makers can better target areas that might require more infrastructure support or regulatory attention. These three distinct regions could be labeled The Southern Border, The Western Canadian Border, and The Eastern Canadian Border. These divisions would create optimal opportunities for the division of leadership.

Decision Tree



The decision tree used in this study shows a clear delineation of border volume. Any latitude greater than 43 degrees north is the US-Canadian border. Anything south of 33 degrees of northern latitude is the US-Mexican border. Along the southern border, the decision tree broke up the border into five distinct cells. Those cells follow along from the busiest port in the country (San Ysidro near San Diego, California) through Arizona, New Mexico, and then drops further south along the entire southern border of Texas. Of note, El Paso is the second busiest port in the country. That is right on the New Mexico/Texas/Mexico junction. That port alone will drive some of the division along the cells.

Simlar to the k-Means analysis, the decision tree cells would be a logical division for leadership responsibilities. By analyzing with two distinct methods, analysts can provide the leadership with two sound options for making decisions about operations. Both of them are right, they were just approached in different manners.

Conclusion

This report presented a comprehensive analysis of the Border Crossing Entry Data, employing a robust set of data science methodologies to uncover significant insights into the patterns and trends of border activities. Through meticulous data cleaning, exploratory data analysis, and k-means clustering, we have not only addressed the initial business questions but also uncovered additional layers of understanding that could significantly influence policy and operational strategies.

Our exploratory data analysis revealed key characteristics and anomalies within the data, setting the stage for a deeper investigation into specific trends. The subsequent clustering analysis grouped the border crossing points into meaningful categories based on their geographic coordinates and the volume of crossings. This spatial grouping has unveiled distinct regions of activity that are critical for targeted policy-making and resource allocation.

The insights derived from this analysis suggest that border management can be significantly optimized by focusing on these clusters. For instance, high-traffic clusters may benefit from enhanced security measures and infrastructure improvements, while low-traffic clusters might be examined for efficiency improvements or even reallocation of resources.

Recommendations have been provided based on the analytical findings, which should guide the next steps in data collection, further research, and policy formulation. Future analyses should consider incorporating additional variables such as time of crossing, types of vehicles,

and purpose of crossing to refine the existing models and potentially reveal more intricate patterns.

In conclusion, this project not only responds to the initial analytical queries but also opens avenues for a data-driven approach in border management. By leveraging the power of data science, stakeholders are better equipped to make informed decisions that enhance efficiency, security, and operational effectiveness at border crossings.

References:

1. Hadley Wickham, Romain François, Lionel Henry, Kirill Müller (2021)

2. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013)

Appendix

```r
# Load required libraries
library(dplyr)    # for data manipulation
library(ggplot2)  # for data visualization
library(rpart)
library(rpart.plot)
# Load the dataset
data <- read.csv("Border_Crossing_Entry_Data.csv")


# Basic data exploration
# What did you do with the data in the context of exploration?

# Check variables and data types
str(data)

# Summary statistics
summary(data)


#How many entries are in the dataset?
# Count number of entries
num_entries <- nrow(data)
num_entries


# Was there missing data? Duplications? How clean was the data?
# Check for missing data
any_missing <- any(is.na(data))
any_missing

# Check for duplicates
duplicated_rows <- data[duplicated(data), ]
duplicated_rows_count <- nrow(duplicated_rows)
duplicated_rows_count



# Were their outliers or suspicious data?
# Check summary statistics for the "Value" variable
summary_stats <- summary(data$Value)         ##  The "Value" column in Border_crossing_entry dataset rep
print(summary_stats)                         ## the quantity or numerical count associated with each ent


# Create a boxplot with colored outliers
boxplot(data$Value, main = "Boxplot of Value with Colored Outliers", ylab = "Value",
        col = "lightblue",  # Color of the box
```

```r
# Identify potential outliers using boxplot statistics
outliers <- boxplot.stats(data$Value)$out
print(outliers)



# What did you do to clean the data?
# Data Cleaning
# Remove duplicates
cleaned_data <- data[!duplicated(data), ]




# Some other graphs
#1.) Calculate total traffic counts by measure and port
traffic_summary <- data.frame(
  Port = cleaned_data$Port.Name,
  Measure = cleaned_data$Measure,
  Traffic_Count = ave(cleaned_data$Value, cleaned_data$Port.Name, cleaned_data$Measure,
)

# Sort the summary dataframe by Traffic_Count in descending order
traffic_summary <- traffic_summary[order(-traffic_summary$Traffic_Count), ]

# Create a bar plot to visualize traffic counts by measure at each port
ggplot(traffic_summary, aes(x = Port, y = Traffic_Count, fill = Measure)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Traffic Counts by Measure at Border Ports",
       x = "Border Ports",
       y = "Traffic Count",
       fill = "Measure") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for


#2.)
# Bar Chart of Activity Types (Measure) with Color
ggplot(cleaned_data, aes(x = Measure, fill = Measure)) +
  geom_bar() +
  labs(title = "Count of Activities by Type", x = "Activity Type", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_discrete(name = "Activity Type")  # Customize legend title
```

```r
# Boxplot of Activity Values by Border Type with Color
ggplot(cleaned_data, aes(x = Border, y = Value, fill = Border)) +
  geom_boxplot() +
  labs(title = "Distribution of Traffic Volume by Border Type", x = "Border Type", y = "Act
  scale_fill_discrete(name = "Border Type")  # Customize legend title

# Bar Chart of Activities by State with Color
ggplot(cleaned_data, aes(x = State, fill = State)) +
  geom_bar() +
  labs(title = "Count of Activities by State", x = "State", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_discrete(name = "State")  # Customize legend title


#3.) Visualization - Scatter plot for Latitude vs. Longitude
ggplot(cleaned_data, aes(x = Longitude, y = Latitude, color = Border)) +
  geom_point() +
  labs(title = "Border Crossing Locations",
       x = "Longitude",
       y = "Latitude") +
  theme_minimal()



#4. ) Calculate total traffic counts by port
port_traffic <- aggregate(Value ~ Port.Name, data = cleaned_data, FUN = sum)

# Sort the data to plot in descending order
port_traffic <- port_traffic[order(-port_traffic$Value), ]

# Create a bar plot to visualize traffic volume by port
ggplot(port_traffic, aes(x = reorder(Port.Name, -Value), y = Value)) +
  geom_bar(stat = "identity", fill = "skyblue", color = "black") +
  labs(title = "Traffic Volume by Port", x = "Port", y = "Traffic Volume") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))  # Rotate x-axis labels for bett



#5.)
# Select relevant columns for clustering (e.g., Latitude, Longitude, Value)
cluster_data <- cleaned_data[, c("Latitude", "Longitude", "Value")]
cluster_data
# Normalize the data (optional but recommended for clustering)
normalized_data <- scale(cluster_data)
```

```r
# Determine the number of clusters (k)
k <- 3  # Adjust the number of clusters as needed based on your preference

# Perform k-means clustering
kmeans_result <- kmeans(normalized_data, centers = k, nstart = 25)

# Add cluster labels to the original data
clustered_data <- cbind(cluster_data, Cluster = kmeans_result$cluster)

# Plot the clustered data
ggplot(clustered_data, aes(x = Longitude, y = Latitude, color = factor(Cluster))) +
  geom_point() +
  labs(title = "K-means Clustering of Border Crossing Data",
       x = "Longitude",
       y = "Latitude",
       color = "Cluster") +
  theme_minimal()

#6.) Load the required library for decision trees

# Assuming 'Value' is the target variable and 'Latitude' is the input feature
# Split data into training and testing sets (e.g., 80% training, 20% testing)
set.seed(123)  # for reproducibility
train_idx <- sample(nrow(cleaned_data), 0.8 * nrow(cleaned_data))
train_data <- cleaned_data[train_idx, ]
test_data <- cleaned_data[-train_idx, ]

# Build the decision tree model using only 'Latitude' as the input feature
tree_model <- rpart(Value ~ Latitude, data = train_data, method = "anova")

# Plot the decision tree
rpart.plot(tree_model, type = 0, extra = 101)

# Make predictions on the testing data
predictions <- predict(tree_model, test_data)


# Evaluate the model (calculate mean squared error)
mse <- mean((test_data$Value - predictions)^2)
print(paste("Mean Squared Error (MSE):", mse))
```