Explaining Urban Transport Decisions: Local Interpretation of Random Forest Models Using SHAP and LIME

Aleksandra Tatko (648925)

December, 2025

Introduction

Urban transport today is heavily dominated by private cars, creating problems such as congestion, pollution, noise, and inefficient use of road space. This trend threatens the liveability of cities and highlights the need for more sustainable mobility options (Gössling 2020). To support the Swedish Ministry of Infrastructure in addressing these challenges, this paper analyzes which factors shape Swedish citizens' commuting choices, providing insights that can guide urban planning and the transition toward greener cities. Using a Random Forest model as a baseline and applying local interpretation methods such as SHAP and LIME, the study aims to answer the question: *"What mode of transport do Swedish citizens choose to commute to work, and what factors influence their decision the most?"*

Data Description

The dataset contains 2,357 observations of commuting trips made by residents of Oslo municipality, described by 14 variables capturing both trip characteristics and socio-demographic factors. The outcome variable, mode, indicates whether an individual commutes by public transport (0) or by car (1), and is strongly imbalanced (367 PT users vs. 2,108 car users). Key predictors include travel times by public transport and car (time_pt, time_car) and their ratio (time_ratio), along with several dummy variables describing trip conditions: whether the journey involves one or multiple transfers (one_transfer, mult_transfer), whether walking distance to the nearest stop exceeds 500 meters (walk_500), whether interchange waiting time exceeds five minutes (wait_5), and whether the total travel distance is above 20 km (dist_20). Individual attributes include income level (high_inc), education level (high_ed), gender (woman), and age. Since Random Forest requires minimal preprocessing, the main data preparation step involved converting time variables character to numeric and replacing 17 missing values in walk_500 with 0, reflecting the reasonable assumption that missing entries indicate no long walking distance reported.

Methodology

The transport mode choice was modeled using a Random Forest classifier, a non-parametric "black-box" method that aggregates many decision trees to achieve strong predictive accuracy but provides little insight into how specific variables influence individual predictions. Because black-box models are difficult to interpret, both global and local explanations are needed. Global to understand general behavioural patterns and local to understand why the model predicts a certain mode for a specific person. Local interpretation is especially relevant for policy use, where diagnosing misclassifications or identifying subgroup behaviours can help institutions such as the Ministry of Infrastructure improve transport planning. Among several available local interpretability techniques (e.g., counterfactuals, local surrogate models), this study focuses on the two most widely used: SHAP and LIME. **SHAP** (SHapley Additive Explanations) is grounded in cooperative game theory and assigns each feature a contribution value based on how much it increases or decreases the predicted probability for a given class. **LIME** (Local Interpretable Model-Agnostic Explanations) instead

constructs a simple surrogate model around the observation of interest by generating perturbed samples and weighting them according to their proximity. Then, it fits a sparse linear model that approximates the black-box model's behaviour locally, highlighting the most influential features. SHAP was chosen for its theoretical consistency and additive structure, while LIME complements it by offering an intuitive, model-agnostic approximation of the Random Forest's local decision rules. Using both methods provides a robust, cross-validated view of why the model predicts "car" or "public transport" for specific individuals.

Results

The dataset was split into a training set and a small ten-case test set to allow detailed local interpretation. Because the outcome variable mode is highly imbalanced, the Random Forest learned a strong bias toward predicting car use. In the test set, it classified all ten cases as car users, producing nine correct predictions and one misclassification. Global variable importance (*Appendix A.1*) confirms that the model relies mainly on travel-time variables: time_ratio is by far the strongest predictor, followed by time_car, time_pt, and age. Service-quality indicators (transfers, waiting time, frequency) contribute little, suggesting that the model primarily learns a rule based on travel-time efficiency. Local interpretability methods clarify how this rule plays out for individuals. For the correctly classified commuter (Case 2463, true mode = car), SHAP and LIME (*Appendix A.2 and A.3*) both show that time_ratio (2.41) and time_car (23.48 minutes) are the main factors pushing the prediction toward car, consistent with the global patterns. Smaller SHAP effects from age (23) and woman = 0 indicate minor demographic influences, while wait_5 = 1 slightly offsets the car prediction. LIME confirms the same structure, with time_ratio, time_car, and dist_20 as the key positive contributors. For the misclassified individual (Case 1038, true mode = PT), both SHAP and LIME explain why the model incorrectly predicted car use. The commuter's extremely high time_ratio (4.56) overwhelmingly pushes the model toward car, and short time_car (7.53 minutes) further reinforces this. Features that might support PT use have negligible effect, revealing a strong mismatch between the model's learned decision rule and this person's actual behaviour. Overall, SHAP and LIME show that the Random Forest effectively applies a simple heuristic—if public transport is much slower than car, predict car. This works for most commuters but fails for users who choose public transport despite large time disadvantages. Local explanations therefore not only clarify correct predictions but also highlight precisely when and why the model misclassifies.

Conclusion and Discussion

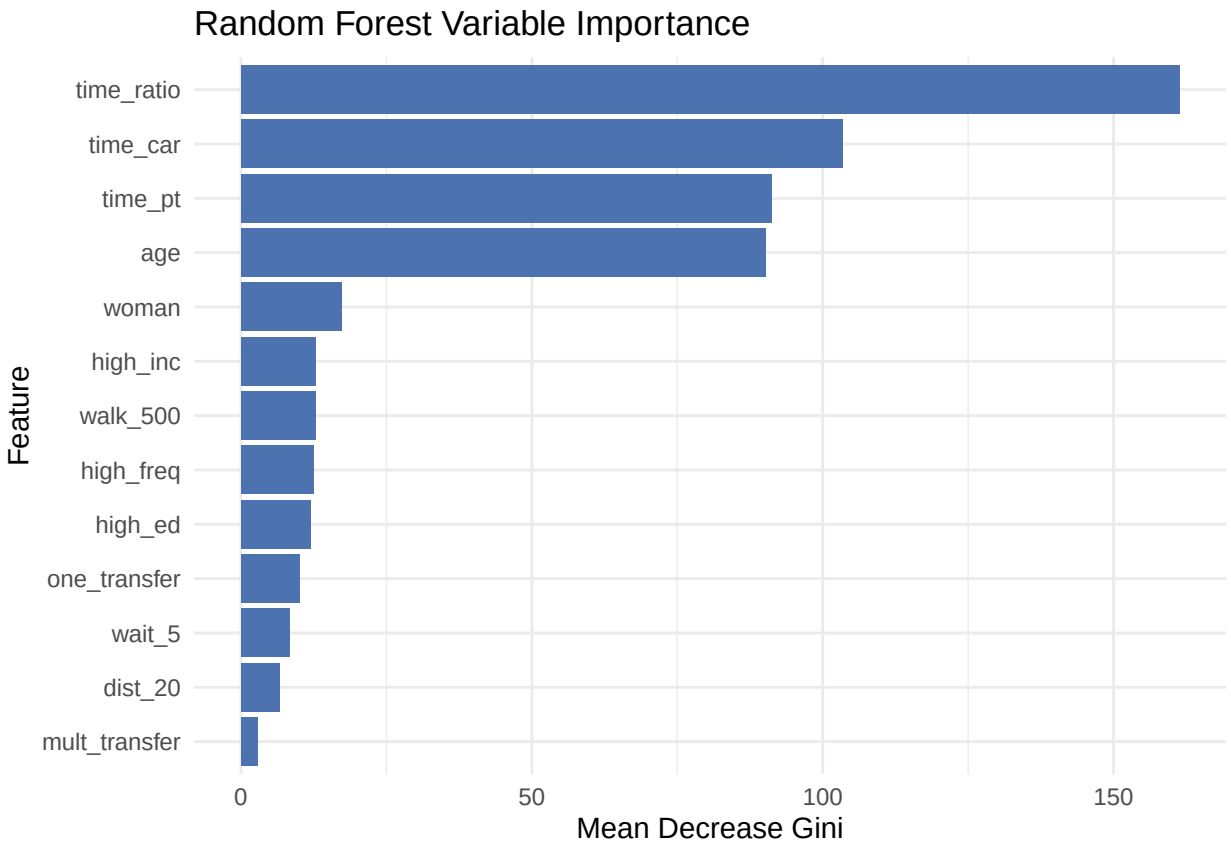
This paper combined a Random Forest model with SHAP and LIME to examine the factors shaping Swedish citizens' commuting choices. It shows that the model significantly prioritises travel-time efficiency, particularly the ratio between public transport and car travel time, while demographic and service-quality factors play only marginal roles. Local explanations further revealed that this rule works well for most commuters but breaks down for individuals who choose public transport despite substantial time disadvantages. This underscores the value of local interpretation tools for diagnosing model blind spots. However, important limitations remain: the dataset lacks social and behavioural variables that often influence mobility decisions, the analysis of only two local cases limits generalisability, and the strong class imbalance means the model reflects patterns rather than causal mechanisms. Future research should therefore incorporate additional behavioural factors and interpret a broader set of individual cases to capture more diverse decision profiles. Nevertheless, the findings still offer actionable insights for the Swedish Ministry of Infrastructure, such as: policies that reduce public transport travel times, well- organized transfers and waiting, and target groups whose preferences are not time-driven can support more effective urban planning and help accelerate the transition toward greener, more sustainable mobility.

References

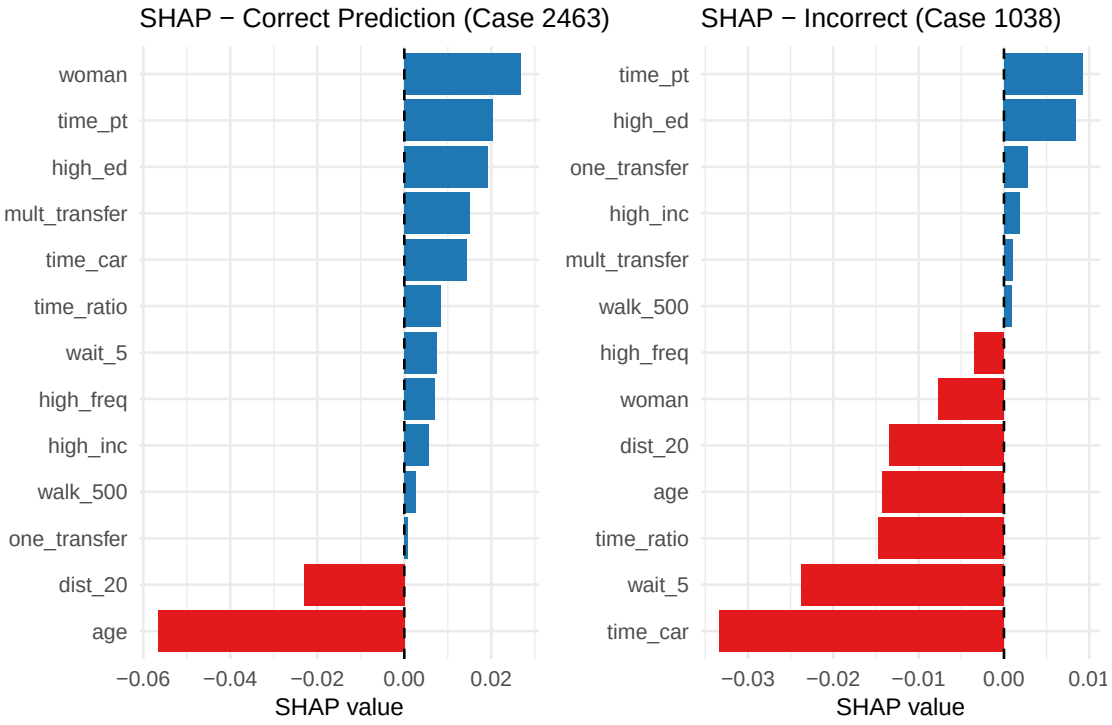
Gössling, Stefan. 2020. "Why Cities Need to Take Road Space from Cars - and How This Could Be Done." *Journal of Urban Design* 25 (4): 443–48. https://doi.org/10.1080/13574809.2020.1727318.

Appendix A

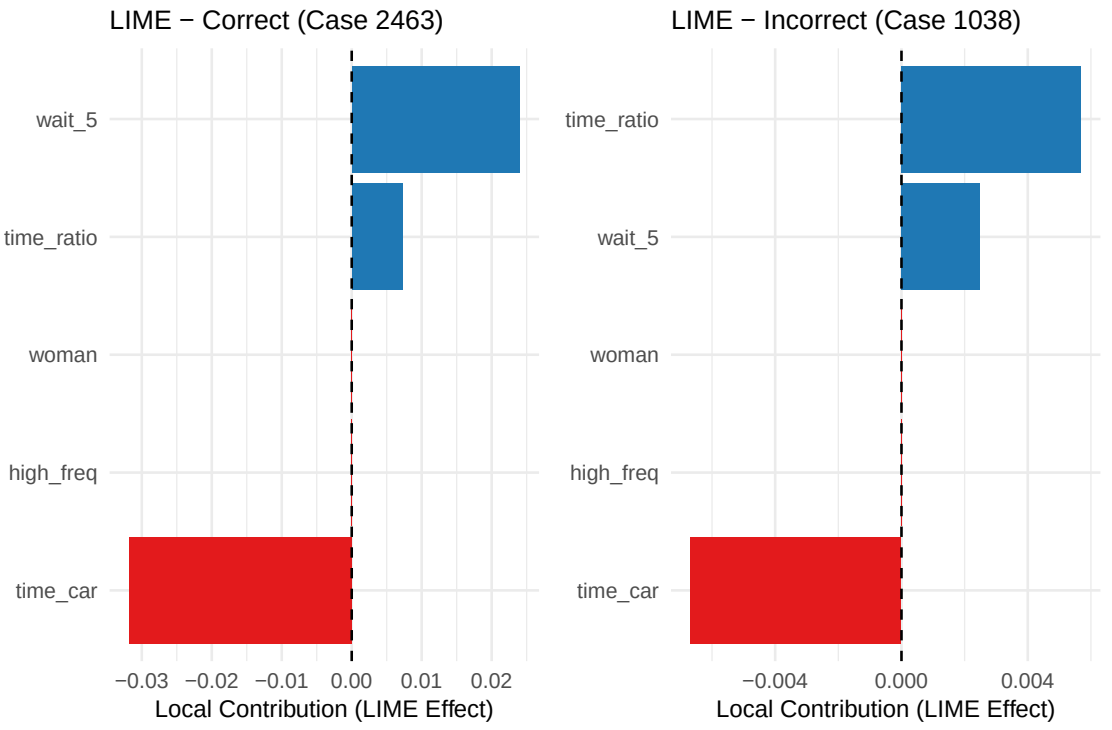
Appendix A.1 Global Interpretation: RF Variable Importance



Appendix A.2 Local Interpretation: SHAP Model



Appendix A.3 Local Interpretation: LIME Model



Appendix B Code

```
#-------------- Preprocessing---------------
str(data)
colSums(is.na(data))
data$walk_500[is.na(data$walk_500)] <- 0
data$mode <- as.factor(data$mode)
table(data$mode)

numeric_vars <- c("time_pt", "time_car", "time_ratio") # Fix decimal commas and convert to numeric
data[numeric_vars] <- lapply(data[numeric_vars], function(x) {
  x <- gsub(",", ".", x)
  as.numeric(x)})
# Random forests do NOT require scaling, normalization, distribution checks, etc.
#-------------RF Model-------------
set.seed(123)
test_idx <- sample(1:nrow(data), 10)
test <- data[test_idx, ]
train <- data[-test_idx, ]
# Train Random Forest Model
rf <- randomForest(
  mode ~ .,
  data = train,
  ntree = 500,
  importance = TRUE)
# Variable importance plot (Appendix A)
imp <- as.data.frame(importance(rf))
imp$Feature <- rownames(imp)
ggplot(imp, aes(x = reorder(Feature, MeanDecreaseGini),
                y = MeanDecreaseGini)) +
  geom_col(fill = "#4C72B0") +
  coord_flip() +
  labs(title = "Random Forest Variable Importance",
       x = "Feature", y = "Mean Decrease Gini") +
  theme_minimal()
# Predictions on Test Set
pred <- predict(rf, test)
comparison <- data.frame( actual = test$mode, predicted = pred)
# Identify one correct + one incorrect case
correct_idx   <- which(comparison$actual == comparison$predicted)[1]
incorrect_idx <- which(comparison$actual != comparison$predicted)[1]
correct_case   <- test[correct_idx, ]
incorrect_case <- test[incorrect_idx, ]
confusionMatrix <- table(predicted = pred, actual = test$mode)
# ------ SHAP Explanations ---------
predictor <- Predictor$new( model = rf, data = train %>%
                              select(-mode), y = train$mode, type = "prob")
# Check prediction for person A and person B
predictor$predict(correct_case %>% select(-mode))
predictor$predict(incorrect_case %>% select(-mode))
# SHAP for correct
set.seed(100)
pred_class_A <- as.numeric(as.character(correct_case$mode))
```

```r
shap_correct <- Shapley$new(
  predictor,
  x.interest = correct_case %>% select(-mode))
# filter for predicted class only
df_correct_filtered <- shap_correct$results %>%
  filter(class == pred_class_A) %>%
  mutate(direction = ifelse(phi > 0, "Positive", "Negative"))

P1 <- ggplot(df_correct_filtered,
             aes(x = phi,y = reorder(feature, phi), fill = direction)) +
  geom_col() +
  scale_fill_manual(values = c("Positive" = "#1f78b4","Negative" = "#e31a1c")) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  theme_minimal(base_size = 10) +
  theme(legend.position = "none", axis.title.y = element_blank(), axis.title.x = element_text(size = 9)
  labs(title = "SHAP - Correct Prediction (Case 2463)", x = "SHAP value")

# SHAP for incorrect
set.seed(100)
pred_class_B <- as.numeric(as.character(incorrect_case$mode))
shap_incorrect <- Shapley$new(
  predictor,
  x.interest = incorrect_case %>% select(-mode))

# filter for predicted class only
df_incorrect_filtered <- shap_incorrect$results %>%
  filter(class == pred_class_B) %>%
  mutate(direction = ifelse(phi > 0, "Positive", "Negative"))

P2 <- ggplot(df_incorrect_filtered,
             aes(x = phi,y = reorder(feature, phi), fill = direction)) +
  geom_col() +
  scale_fill_manual(values = c("Positive" = "#1f78b4", "Negative" = "#e31a1c")) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  theme_minimal(base_size = 10) +
  theme(legend.position = "none", axis.title.y = element_blank(), axis.title.x = element_text(size = 9)
  labs(title = "SHAP - Incorrect (Case 1038)", x = "SHAP value")
# ---------- LIME Interpretation ------------
predictor_lime <- Predictor$new(
  model = rf,data = train %>% select(-mode), y = train$mode, type = "prob", class = "1" )

# LIME correct prediction
set.seed(100)
lime_correct <- LocalModel$new(
  predictor_lime,
  x.interest = correct_case %>% select(-mode),
  k = 5)

df_lime_correct <- lime_correct$results %>%
  mutate(direction = ifelse(effect > 0, "Positive", "Negative"))

p_lime_correct <- ggplot(df_lime_correct, aes(x = effect, y = reorder(feature, effect), fill = directio
  geom_col() +
```

```
  scale_fill_manual(values = c("Positive" = "#1f78b4", "Negative" = "#e31a1c")) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  theme_minimal(base_size = 10) +
  theme(legend.position = "none", axis.title.y = element_blank(), axis.title.x = element_text(size = 9)
    axis.text = element_text(size = 8), plot.title = element_text(size = 10), plot.margin = margin(5,5,5
  labs(title = "LIME - Correct Prediction (Case 2463)", x = "Local Contribution (LIME Effect)")
# ---------------- LIME FOR INCORRECT PREDICTION ----------------
set.seed(100)
lime_incorrect <- LocalModel$new(predictor_lime,
  x.interest = incorrect_case %>% select(-mode),
  k = 5)

df_lime_incorrect <- lime_incorrect$results %>%
  mutate(direction = ifelse(effect > 0, "Positive", "Negative"))

p_lime_incorrect <- ggplot(df_lime_incorrect,aes(x = effect, y = reorder(feature, effect), fill = direc
  geom_col() +
  scale_fill_manual(values = c("Positive" = "#1f78b4","Negative" = "#e31a1c")) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  theme_minimal(base_size = 10) +
  theme(
    legend.position = "none",
    axis.title.y = element_blank(),
    axis.title.x = element_text(size = 9),
    axis.text = element_text(size = 8),
    plot.title = element_text(size = 10),
    plot.margin = margin(5,5,5,5)
  ) +
  labs(title = "LIME - Incorrect (Case 1038)", x = "Local Contribution (LIME Effect)")
lime_correct$results
lime_incorrect$results
correct_case
incorrect_case
```