Online-Shoppers Purchase Intention

Fabienne Bölsterli, David Gerner, Valérie Lüthi

Machine Learning 1 - FS25

1 Introduction

The e-commerce industry has grown rapidly over the past years, yet conversion rates are not increasing as much. To address this gap, online retailers are looking for solutions to present online shoppers with personalized promotions. Whereas physical retail relies on their salespeople's knowledge and experience to realize successful sales, online retail makes use of machine learning to detect and predict customers behavior patterns (Sakar et al., 2019)[1].

In this project, our goal is to predict whether a visitor of an online shopping website will complete a purchase or exit the website without purchasing anything. To achieve this, we will apply the different machine learning models that were taught in the "Applied Machine Learning and Predictive Modelling 1" class at Hochschule Luzern (HSLU) in the spring semester of 2025.

1.1 Dataset

The data set used in this project was initially collected by Sakar et al. (2019)[1] for their real-time online shopper behavior analysis system to predict visitor's shopping intent and likelihood of website abandonment. It is made available through the UCI Machine Learning Repository (Sakar & Kastro, 2018)[2].

The data set consists of 12'330 observations of shopping sessions for 18 variables in total. Each session belongs to a different user within a one-year time frame. This ensures that the data is not confounded by specific campaigns, special days, user profiles, or seasonal effects. According to Sakar et al. (2019)[1], the variables can be described as follows.

There are six variables containing information about different types of pages visited, both as the total number of such pages and the total time spent on them. *Administrative* measures the total number of pages about account management visited, *Administrative_Duration* measures the total amount of time in seconds spent on such pages. *Informational* measures the total number of pages with information on the Website, communication and address visited, *Informational_Duration* measures the total amount of time in seconds spent on such pages. *ProductRelated* measures the total number of pages related to products visited, *ProductRelated_Duration* measures the total amount of time in seconds spent on such pages.

There are three variables containing metrics measured by Google Analytics for the pages in the e-commerce site. *BounceRates* represents the average bounce rate of the web pages visited by this visitor. The bounce rate of a web page indicates the proportion of visitors who leave the web site after viewing only this page. *ExitRates* represents the average exit rate of the web pages visited by a visitor during their session. The exit rate of a web page indicates the proportion of visitors who leave the web site from this page. *PageValues* represents the average page value of the web pages visited by the visitor. The page value is the average monetary value of a web page visited by a visitor before completing a transaction.

There is an additional variable *SpecialDay* indicating the closeness of the visiting time of a web page to a special day (e.g. Valentine's Day), taking into account the duration between the order and delivery date.

The data set also contains multiple categorical variables with session and user information. There is information on the operating system (*OperatingSystems*) and browser (*Browser*) used by the visitor and the geographic region (*Region*) where the session was started. *TrafficType* specifies how the visitor arrived at the website (e.g. direct, banner) and *VisitorType* specifies whether the visitor is new, returning, or other. There is also information on the visiting date, whether it falls on the weekend (*Weekend*) and during which month it is (*Month*). Finally, there is *Revenue* indicating whether the session was finalized with a transaction.

2 Data Preparation

This chapter provides an overview how the data set was prepared before the exploratory graphical analysis. For this report and the different analysis, several libraries were used.

▶ *Click to see all libraries*

The "Online Shoppers Intention" data set contains 12'330 observations of 18 variables of which ten are numeric and eight categorical. The data set contains no missing values.

▶ *Click to see the full dataset*

As the final step of data preparation, all variables representing categorical data were converted into factors to ensure their accurate representation. Since the data is from the same year, we decided to convert the variable *Month* into a factor instead of a date. It only has ten levels since the months January and April do not appear in the data set.

▶ *Click to see the data set after factor definition*

3 Exploratory Graphical Analysis

The exploratory graphical analysis was conducted in order to gain extensive understanding of the different variables and their relationships. The main focus was on *Revenue* as a binary target variable to indicate whether the visit had been finalized with a transaction or not.

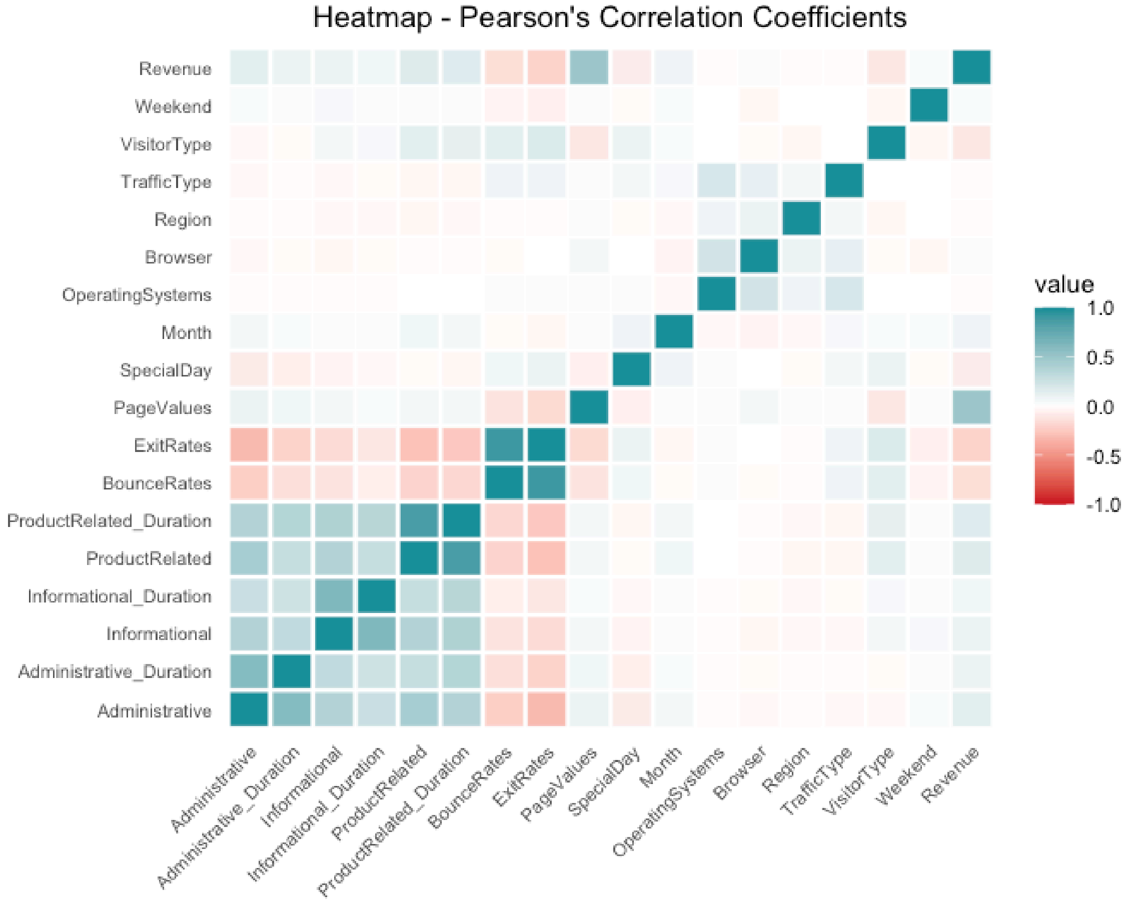
Since class balance is crucial for model training, we first examined the distribution of our target variable *Revenue*. The analysis showed a significant imbalance with 10’422 (84.5%) instances of visitors abandoning the shopping session and only 1’908 (15.5%) instances of finalized transactions. This imbalance might lead to our models being biased toward predicting the majority class unless corrective measures are taken.

Revenue	Count	Proportion
FALSE	10422	0.845
TRUE	1908	0.155

3.1 Correlation Matrix

To get a better understanding of the relationships between the variables, we computed the pearson correlation coefficients for every relationship. This is visualized in the correlation heat map with the color indicating the strength and direction of the correlations.

▶ *Click to see the code for the correlation heatmap*



For the variables *OperatingSystems*, *Region* and *TrafficType* (cor = -0.01) as well as *Browser* (cor = 0.02), almost no linear correlation with our target variable *Revenue* can be observed. For *Revenue*, the largest linear correlation by far is with *PageValues* (cor = 0.49). This makes sense, considering *PageValues* reflects the average monetary value of the pages a user visits. A small to medium correlation is found with *ExitRate* (cor = -0.21), implicating that higher exit rates are associated with lower revenue. All the other variables show weak correlations with *Revenue*.

There are noteworthy large correlations between other variables too. *ProductRelated* shows a strong positive relationship with *ProductRelated_Duration* (cor = 0.86), meaning the higher the number of product related pages visited, the higher the time spent on those pages. *BounceRates* and *ExitRates* unsurprisingly show another strong positive correlation (cor = 0.91), as well as *Administrative* and *Administrative_Duration* (cor = 0.60) and *Informational* and *Informational_Duration* (cor = 0.62). It can be assumed that those high correlations are due to the variable pairs measuring different aspects of the same underlying shopping behavior.

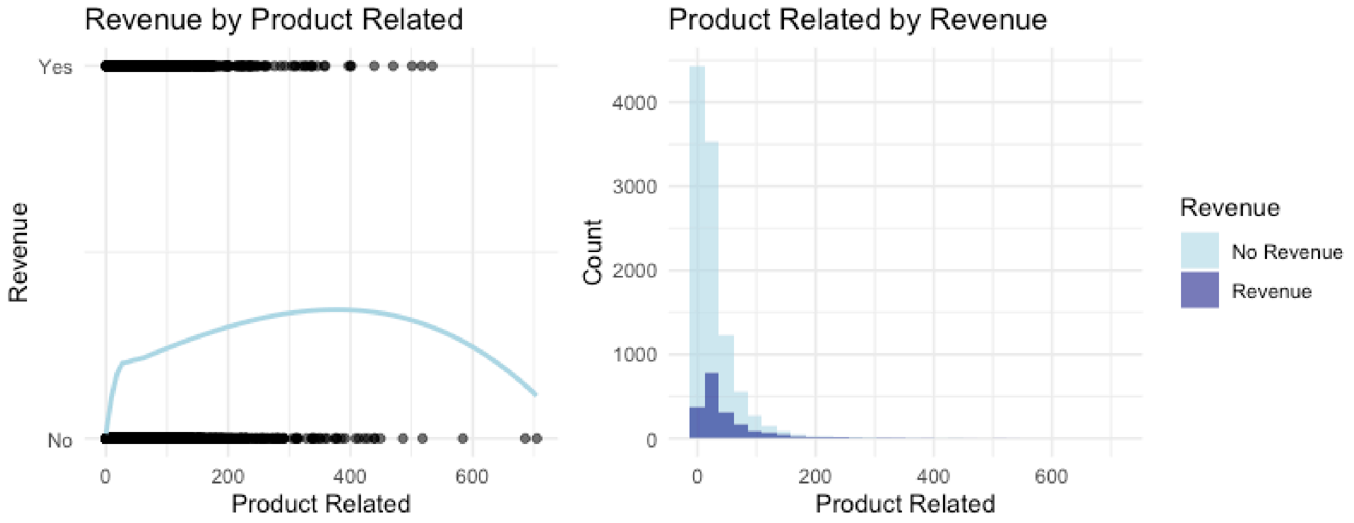
▶ *Click to see the correlation coefficient matrix*

3.2 Key Variables

We now further explore the variables that showed the highest correlations with *Revenue*.

3.2.1 Product Related and Product Related Duration

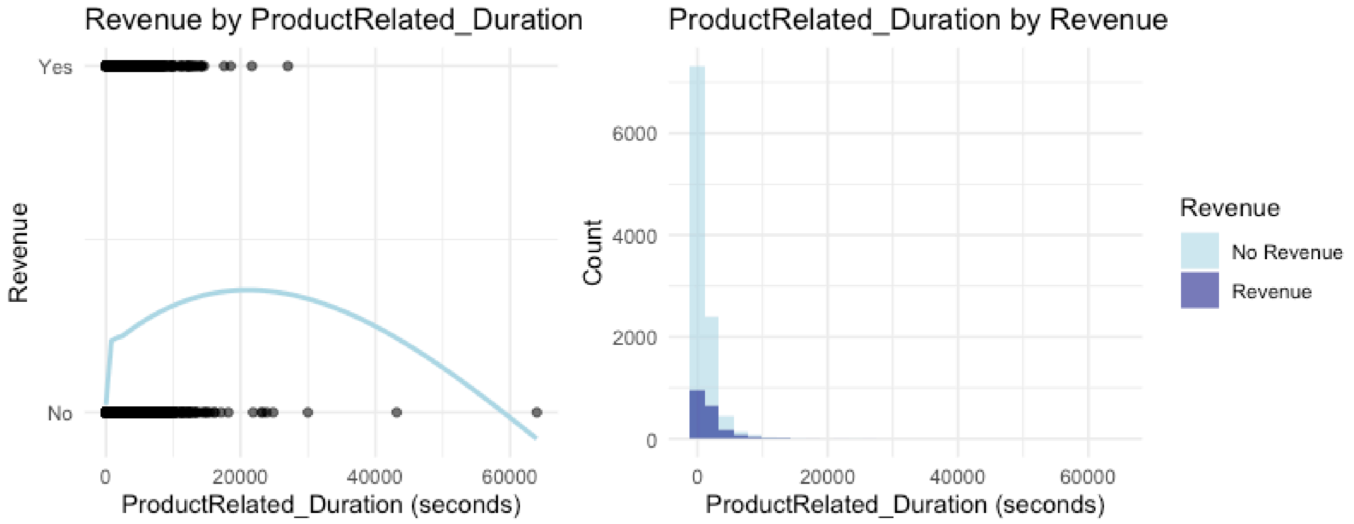
▶ *Click to see the code to create the plots for ProductRelated*



The curve in the scatterplot on the left shows the effect of product-related pages on revenue. As the number of product-related page views increases, the probability of making a purchase initially increases. However, after around 400 product-related pages, the probability of a purchase decreases with additional product-related page views.

The histogram on the right displays the distribution of visitors based on the number of product-related page views. Most visitors viewed fewer than 200 product-related pages, and the number of visitors drops off quickly as the number of product-related page views increases. While the overall shape of the distribution is similar for both groups, visitors who made a purchase tend to view slightly more product-related pages compared to visitors that did not make a purchase.

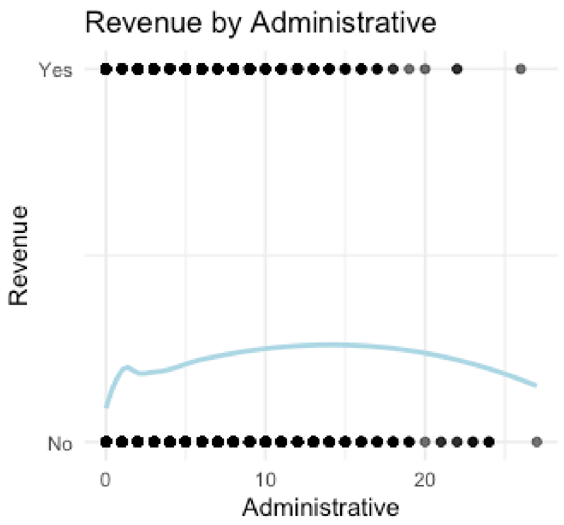
▶ *Click to see the code to create the plots for ProductRelated_Duration*

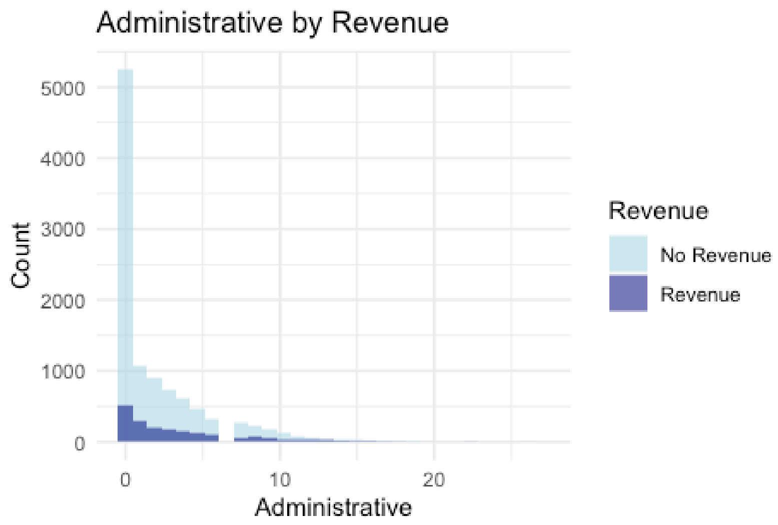


The scatterplot on the left shows how the number of seconds spent on product-related pages relates to the probability of generating revenue. The curve rises until about 20’000 seconds and drops significantly afterwards. This might indicate that more engaged browsing initially increases the likelihood of a visitor to make a purchase. In the histogram on the right we see that most visitors spend less than 10’000 seconds on product pages. Overall, the plots for *ProductRelated* and *ProductRelated_Duration* show very similar patterns.

3.2.2 Administrative and Informative

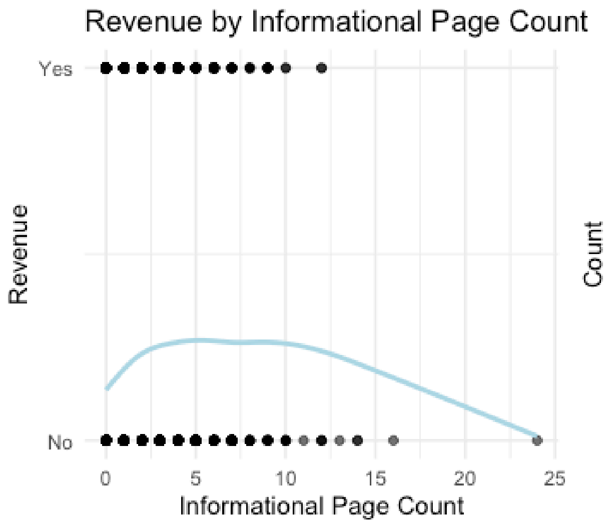
▶ *Click to see the code to create the plots for Administrative*

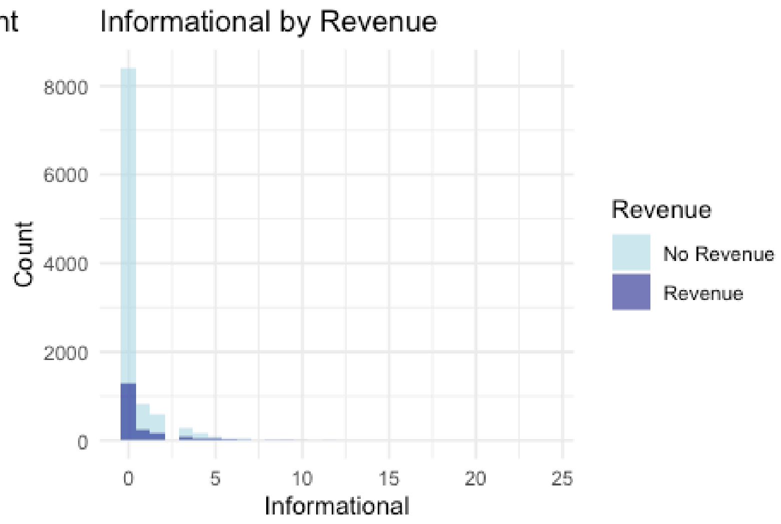




The scatterplot on the left shows how the number of administrative pages relates to the probability of generating revenue. The probability for revenue increases as users view more administrative pages peaking at around 12 but then diminishes at higher counts. This suggests that moderate use of administrative pages may signal purchasing intent, while excessive use of such pages does not translate into more revenue. The histogram shows that the majority of visitors visited zero or few administrative pages, especially visitors that did not purchase anything often visited no administrative page at all.

▶ *Click to see the code to create the plots for Informational*

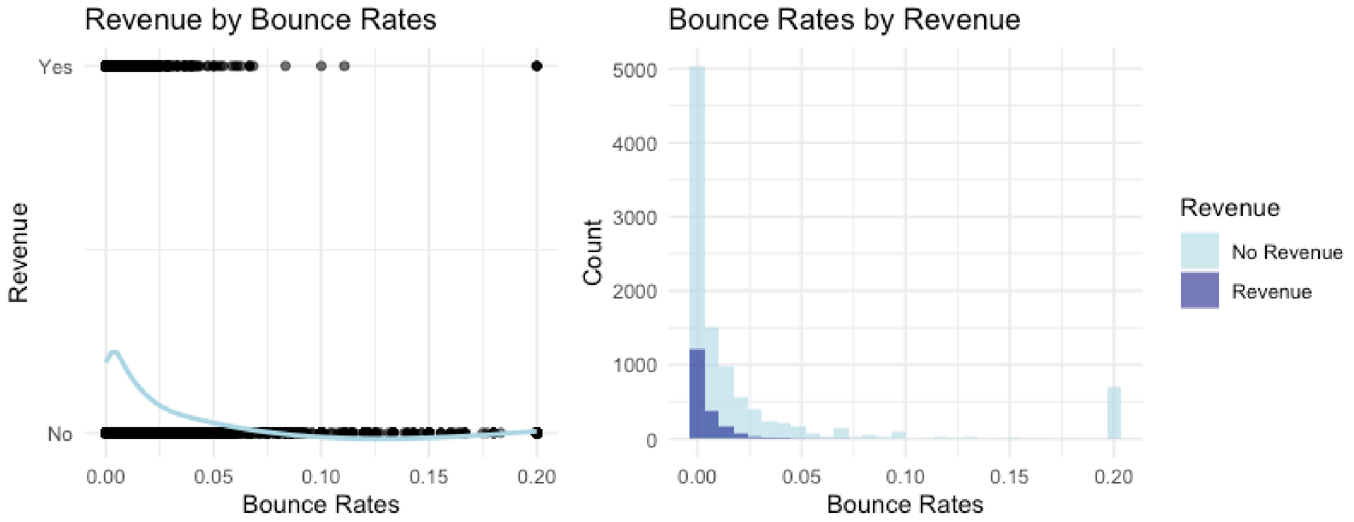




The scatterplot on the left shows how the number of informational pages relates to the probability of generating revenue. The likelihood of generating revenue increases with the number of informational pages a user visits until around five informational pages before flattening and eventually declining after around ten informational pages. This suggests that high engagement with informational pages alone is not associated with a greater likelihood of generating revenue. The histogram shows that the majority of visitors visited zero or very few informational pages, both for revenue as well as non-revenue events.

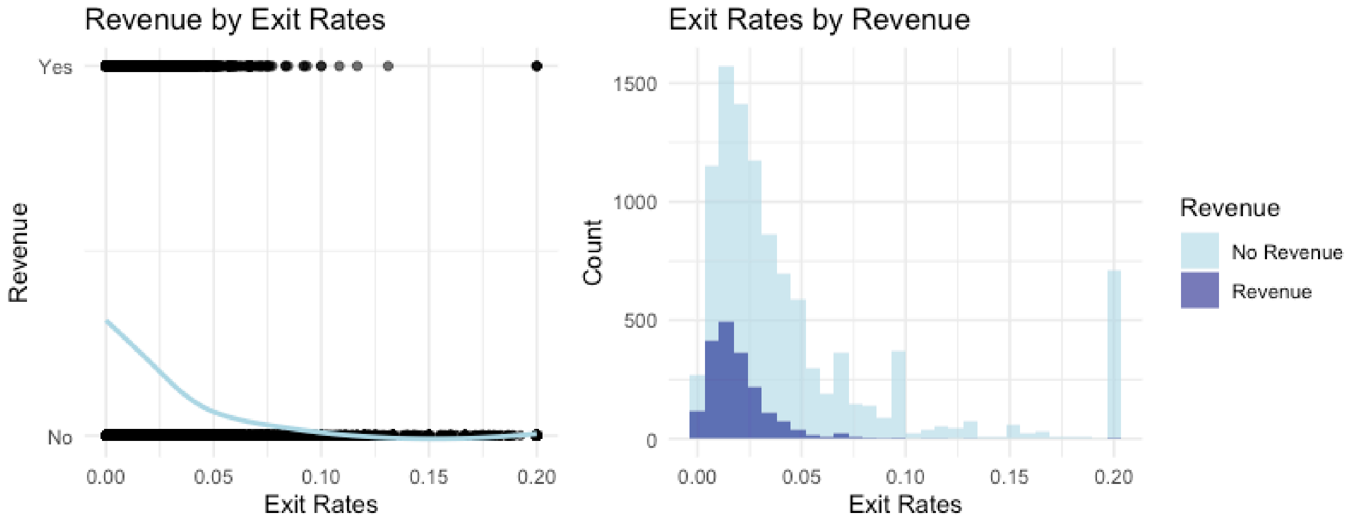
3.2.3 Bounce and Exit Rates

▶ *Click to see the code to create the plots for BounceRates*



The scatterplot on the left shows how bounce rates relate to the likelihood of generating revenue. The curve peaks at very low bounce rates and then drops quickly. This suggests that visitors with lower bounce rates are more likely to make a purchase. As the bounce rates go up, the likelihood of making a purchase drops quickly. The histogram on the right shows how bounce rates differ between visitors who made a purchase and those who did not. Most visitors who made a purchase have very low bounce rates. Visitors who did not make a purchase have higher bounce rates with a noticeable cluster at around 0.2. This indicates that visitors who interact with the site and thus have lower bounce rates are more likely to generate revenue.

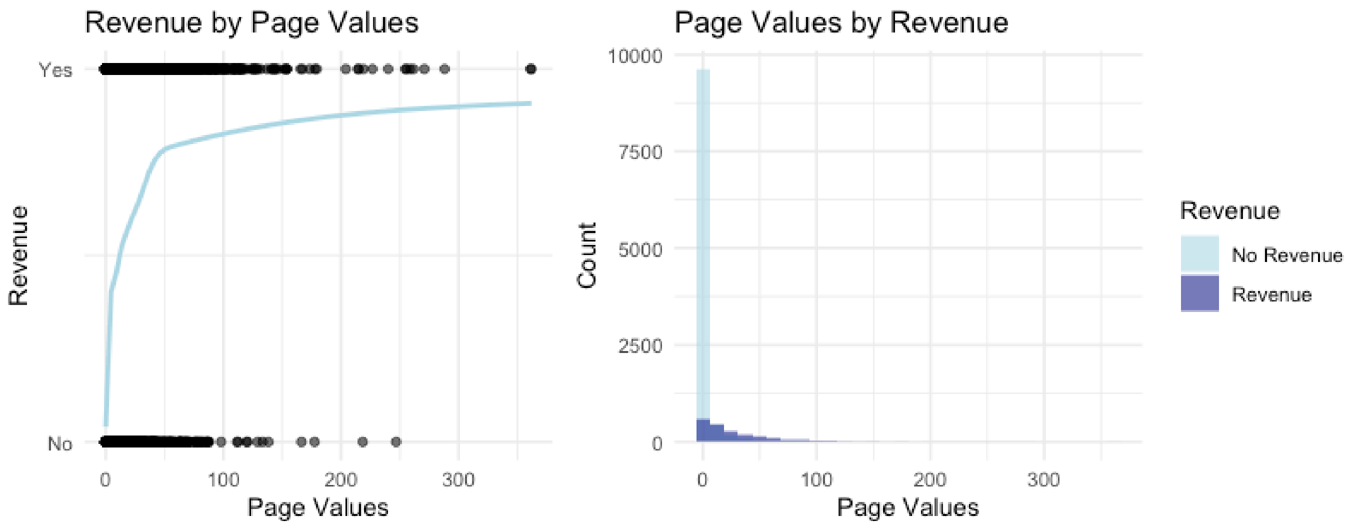
▶ *Click to see the code to create the plots for ExitRates*



The scatterplot on the left shows how exit rates are related to the likelihood of generating revenue. The probability of visitors making a purchase is high for low exit rates and decreases quickly as the exit rates increase. The histogram on the right compares the exit rates of visitors who made a purchase and those who did not. Visitors who made a purchase tend to have lower exit rates, mostly under 0.05. On the other hand, visitors who did not make a purchase are spread out more with an additional peak for high exit rates around 0.20. This indicates that visitors who continue to browse and do not leave the webpage right away and thus have lower exit rates are more likely to generate revenue

3.2.4 Page Values

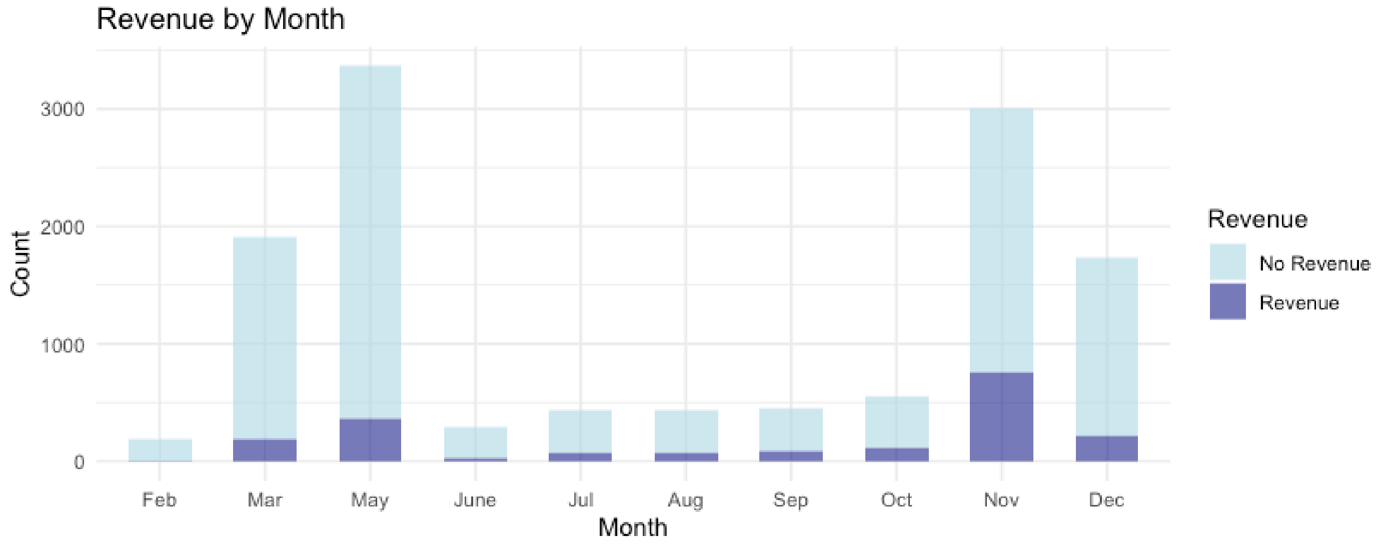
▶ *Click to see the code to create the plots for PageValues*



The scatterplot on the left shows how page values relate to the likelihood of generating revenue. The probability of visitors making a purchase is low for very low page values but increases drastically with higher page values. The histogram on the right shows how page values differ between visitors who made a purchase and those who did not. Most visitors who ended up not buying anything visited pages with low values. On the other hand, visitors who made a purchase visited pages with higher values. This indicates that visitors who visit pages with higher values are more likely to generate revenue.

3.2.5 Month

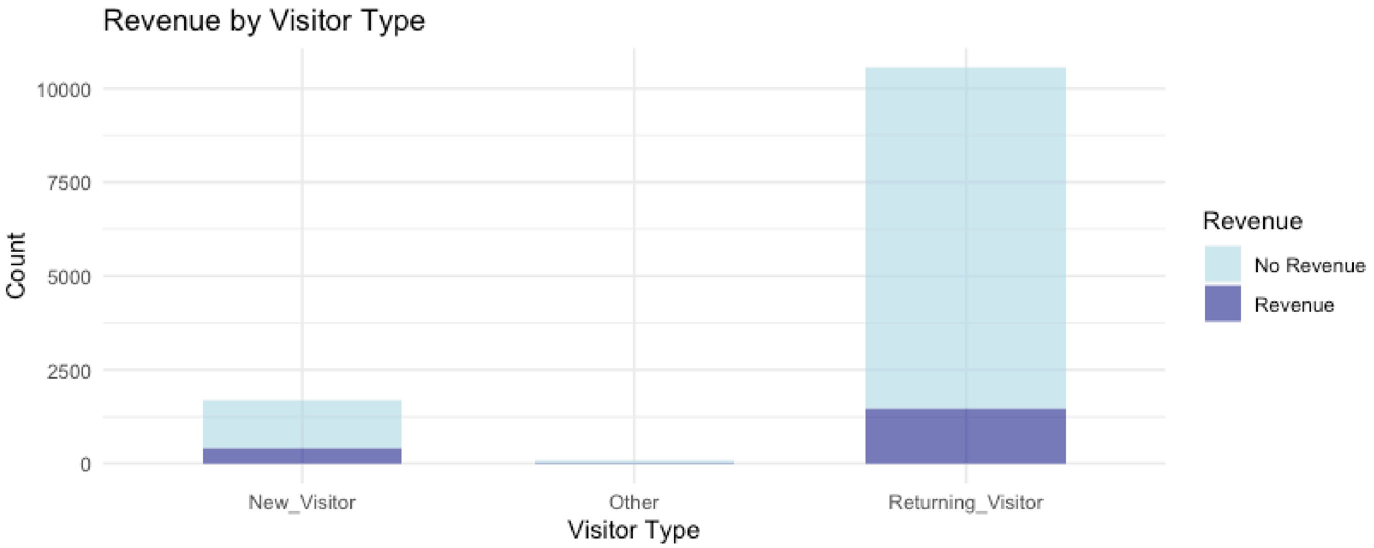
▶ *Click to see the code to create the barplot for Month*



Looking at the relationship between month and revenue, there are some noticeable differences both in terms of overall visits and visits resulting in revenue. November generated the highest revenue, followed by May, December and March. This might be related to gift-buying for holidays like Christmas and Easter, and the peak in November might be related to major shopping events such as Black Friday and Cyber Monday. Since the data covers only a single year, it is difficult to identify seasonal patterns.

3.2.6 Visitor Type

▶ *Click to see the code to create the barplot for VisitorType*



Looking at the relationship between month and visitor type, there is a large differences both in terms of overall visits and visits resulting in revenue. Overall, there were significantly more returning visitors, but they were less likely to actually make a purchase. On the other hand, there were fewer new visitors, but they were more likely to make a purchase. This suggests that new visitors are more likely to buy, while returning visitors tend to browse more without completing a purchase.

4 Linear Model

Our target variable *Revenue* is binary and therefore violates key assumptions of linear regression. Since *Revenue* showed the largest correlation with the continuous variable *PageValues*, we decided to use it as target variable for our linear model. *PageValues* is an estimate of how much a page contributes to revenue during a shopping session. This helps to not just predict immediate sales but also future purchasing potential. Since *PageValues* is an amount and its distribution heavily right skewed, log-transformation was necessary. Due to *PageValues* containing zero values, *log(y + 1)* was used to ensure that all values are positive and can be log-transformed.

▶ *Click to see the code for the linear regression model*

```
## Significant Predictors (p < 0.05):
```

```
##                                Estimate   Std. Error    t value     Pr(>|t|)
## (Intercept)                 0.201555474 0.0617521916   3.263940 1.101752e-03
## Administrative              0.049276783 0.0035841896  13.748375 1.079378e-42
## Informational               0.025327257 0.0090811551   2.788991 5.295411e-03
## ProductRelated             -0.001196428 0.0004058605  -2.947880 3.205639e-03
## BounceRates                 3.890484752 0.4462397292   8.718374 3.183299e-18
## ExitRates                  -5.728682217 0.4710405417 -12.161760 7.778769e-34
## SpecialDay                 -0.214274970 0.0493616557  -4.340919 1.430254e-05
## MonthMay                    0.189435512 0.0496198236   3.817739 1.353433e-04
## OperatingSystems2           0.115412382 0.0475457146   2.427398 1.522186e-02
## OperatingSystems4             0.166205425 0.0494753418   3.359359 7.836168e-04
## OperatingSystems6             0.594364654 0.2213954947   2.684628 7.270808e-03
## Region5                       0.130638889 0.0547505310   2.386075 1.704434e-02
## TrafficType3                 -0.070518319 0.0290996102  -2.423342 1.539280e-02
## TrafficType7                  0.418978076 0.1503640833   2.786424 5.337503e-03
## TrafficType16                -1.177037610 0.5445742180  -2.161391 3.068447e-02
## VisitorTypeReturning_Visitor  0.098061109 0.0275827597   3.555159 3.791721e-04
## RevenueTRUE                   2.099383217 0.0246912044  85.025549 0.000000e+00
```

```
## R-squared:  0.4524511
```

```
## Residual Standard Error (RSE):  0.9407011
```

Intercept (Estimate: 0.2016, p < .01) The intercept indicates that if in the context of the linear model all predictor variables were set to zero, the expected average log-transformed *PageValues* would be 0.20. While this this might not represent a realistic scenario in practice, it serves as a reference point in the model for predictions.

In order to keep this report concise we will only look at the interpretation of some selected predictor variables.

Administrative (Estimate: 0.0493, p < .001) For each additional administrative page viewed, the average log-transformed *PageValues* is expected to increase by 0.049, assuming all other variables remain constant. This suggests that administrative pages may contribute positively to user engagement or conversion.

Informational (Estimate: 0.0253, p < .01) For each additional informational page visited, the average log-transformed *PageValues* is expected to increase by 0.025, holding all other predictors constant. This indicates a small but statistically significant positive association between informational content and the *PageValues*.

Product Related (Estimate: -0.0012, p < .01) For each additional unit increase in the number of product-related pages visited, the average log-transformed *PageValues* is expected to decrease by 0.0012, if all other predictors are kept constant. This could indicate that, on average, more product-related interactions correlate with a slight reduction in the *PageValues*. Yet this result is surprising since the variables *ProductRelated* and *PageValues* showed a weak positive correlation (cor = 0.06) in the initial exploratory analysis. This difference could be due to multicollinearity between the variables within the linear model.

Bounces Rates (Estimate: 3.8905, p < .001) For each unit increase in *BounceRates*, the average log-transformed *PageValues* is expected to increase by 3.89. Since the bounce rate changes in much smaller increments in our data, we could as well say that each percentage point increase in *BounceRates* corresponds to an increases of 0.0389 in $log(PageValues + 1)$, if all other predictors are kept constant. This suggests that shopping sessions with higher bounce rates tend to have higher page values, which could reflect more engagement in those sessions.

Exit Rates (Estimate: -5.7287, p < .001) For each unit increase in *ExitRates*, the average log-transformed *PageValues* is expected to decrease by 5.73, holding all other variables constant. On average, higher *ExitRates* are associated with lower *PageValues*. This reflects the lack of further engagement after a user exits the page.

Special Day (Estimate: -0.2143, p < .001) Shopping sessions occurring closer to a *SpecialDay* are associated with a decrease of 0.21 in *log(PageValues + 1)*. This suggests that visits around special days may be less likely to generate value.

VisitorType: Returning Visitor (Estimate: 0.0981, p < .01) Compared to new visitors, returning visitors are associated with a 0.098 increase in log-transformed *PageValues*, holding all other variables constant. This implies that returning visitors tend to create a slightly higher average *PageValues*.

Revenue (Estimate: 2.0993, p < .001) Sessions resulting in *Revenue* are associated with a 2.10 increase in *log(PageValues + 1)* compared to non-purchase sessions. This highlights the strong relationship between *Revenue* and *PageValues*.

R-squared (0.452) About 45.2% of the variance in the log-transformed *PageValues* is explained by the model. While many predictors contribute significantly to the explanatory power of the model, there remains over 50% of unexplained variance.

Residual Standard Error (0.9407) The Residual Standard Error (RSE) indicates how far off the predictions of our model are from the actual log-transformed *PageValues*. The RSE of 0.941 suggests a decent model fit given the log transformation and skewness of the original *PageValues*.

To assess each predictors individual contribution to the model and potentially simplify it by retaining only the variables that significantly improve the prediction of *PageValues*, we used the *drop1()* function with an F-test.

```
drop1(lm.page.values, test = "F")
```

```
## Single term deletions
##
## Model:
## log(PageValues + 1) ~ Administrative + Administrative_Duration +
##     Informational + Informational_Duration + ProductRelated +
##     ProductRelated_Duration + BounceRates + ExitRates + SpecialDay +
##     Month + OperatingSystems + Browser + Region + TrafficType +
##     VisitorType + Weekend + Revenue
##                         Df Sum of Sq   RSS     AIC  F value    Pr(>F)
## <none>                               10851 -1439.7
## Administrative           1     167.3 11018 -1253.0 189.0178 < 2.2e-16 ***
## Administrative_Duration  1       0.2 10851 -1441.4   0.2070  0.649150
## Informational            1       6.9 10858 -1433.8   7.7785  0.005295 **
## Informational_Duration   1       1.2 10852 -1440.3   1.3310  0.248657
## ProductRelated           1       7.7 10859 -1432.9   8.6900  0.003206 **
## ProductRelated_Duration  1       1.9 10853 -1439.5   2.1508  0.142524
## BounceRates              1      67.3 10918 -1365.5  76.0100 < 2.2e-16 ***
## ExitRates               1     130.9 10982 -1293.8  147.9084 < 2.2e-16 ***
## SpecialDay              1      16.7 10868 -1422.7   18.8436  1.43e-05 ***
## Month                   9      95.6 10946 -1349.5   12.0082 < 2.2e-16 ***
## OperatingSystems        6      38.7 10890 -1407.8    7.2839  8.74e-08 ***
## Browser                11       9.4 10860 -1451.0    0.9648  0.476355    
## Region                  8      12.1 10863 -1441.9    1.7057  0.091599 .  
## TrafficType            19      32.5 10883 -1440.7    1.9347  0.008573 ** 
## VisitorType             2      11.7 10863 -1430.4    6.6083  0.001354 ** 
## Weekend                 1       0.1 10851 -1441.6    0.0650  0.798738    
## Revenue                 1    6397.4 17248  4272.9 7229.3440 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Key Observations The predictors *Administrative*, *Informational*, *ProductRelated*, *BounceRates*, *ExitRates*, *SpecialDay*, *TrafficType*, *VisitorType* and *Revenue* are statistically significant and contribute to the model since removing either of them would result in an increase in the residual sum of squares (RSS).

The predictor *Month* is statistically significant as well, yet our linear regression model showed that out of its nine levels, only May was significant. Despite the increase in the RSS, the model performance does not change dramatically. The predictor *OperatingSystems* is also significant, in the linear model only the *OperatingSystems* 2, 4 and 6 seem to make a statistically significant difference.

The remaining variables seem to not have much of an impact on the model performance showing only a minimal increase in the RSS and p-values greater than 0.05.

Valérie Lüthi took the lead in the Linear Model section.

5 Generalised Additive Models

Using all possible predictors to predict *Revenue*, a full generalized additive model (GAM) was fit allowing for smoothing in all numerical variables. As only a part of the variables were significant, a smaller model was fit containing only the significant variables while still allowing for smoothing in all numerical variables. As the effect of the smoothing term for *BounceRates* is almost significant, we keep it in the reduced model, where its effect becomes significant.

▶ *Click to see the full GAM model*

▶ *Click to see the reduced GAM model*

Looking at the estimated degree of freedom for the smoothing terms in the smaller model, *Administrative* displayed an almost linear effect and was thus modeled with a linear term in the final GAM model. As only one level of *Browser* showed a significant effect in the GAM, we decided to remove it for a simpler model. This decision was supported by a lower AIC for the simple GAM model.

▶ *Click to see the simple GAM model*

```
## 
## Family: binomial 
## Link function: logit 
```

```
##
## Formula:
## Revenue ~ Administrative + s(ProductRelated_Duration) + s(BounceRates) +
##     s(ExitRates) + s(PageValues) + Month + TrafficType + VisitorType
##
## Parametric coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.824e+00  2.130e-01  -8.567  < 2e-16 ***
## Administrative               -5.657e-02  1.033e-02  -5.476 4.36e-08 ***
## MonthDec                     -7.569e-01  2.016e-01  -3.754 0.000174 ***
## MonthFeb                     -1.688e+00  6.547e-01  -2.578 0.009932 **
## MonthJul                      7.999e-02  2.431e-01   0.329 0.742101
## MonthJune                    -3.742e-01  3.068e-01  -1.220 0.222566
## MonthMar                     -6.447e-01  2.017e-01  -3.197 0.001389 **
## MonthMay                     -7.940e-01  1.870e-01  -4.245 2.19e-05 ***
## MonthNov                      4.813e-01  1.845e-01   2.609 0.009076 **
## MonthOct                     -1.769e-01  2.247e-01  -0.787 0.431272
## MonthSep                     -8.858e-02  2.336e-01  -0.379 0.704574
## TrafficType2                  2.528e-01  1.029e-01   2.456 0.014052 *
## TrafficType3                 -6.525e-02  1.333e-01  -0.490 0.624342
## TrafficType4                  1.294e-01  1.505e-01   0.859 0.390096
## TrafficType5                  3.943e-01  2.366e-01   1.667 0.095548 .
## TrafficType6                 -8.729e-02  2.108e-01  -0.414 0.678775
## TrafficType7                  3.236e-01  4.985e-01   0.649 0.516267
## TrafficType8                  7.099e-01  1.950e-01   3.640 0.000272 ***
## TrafficType9                  2.073e-01  6.787e-01   0.305 0.760012
## TrafficType10                 4.916e-01  1.822e-01   2.698 0.006979 **
## TrafficType11                 6.248e-01  2.293e-01   2.725 0.006433 **
## TrafficType12                -4.478e+01  6.711e+07   0.000 0.999999
## TrafficType13                -4.856e-01  2.108e-01  -2.303 0.021263 *
## TrafficType14                 4.437e-01  9.857e-01   0.450 0.652615
## TrafficType15                -4.464e+01  1.089e+07   0.000 0.999997
## TrafficType16                 2.766e+00  1.239e+00   2.232 0.025647 *
## TrafficType17                -4.430e+01  6.711e+07   0.000 0.999999
## TrafficType18                -4.450e+01  2.122e+07   0.000 0.999998
## TrafficType19                -2.413e-01  1.374e+00  -0.176 0.860638
## TrafficType20                 6.466e-01  2.795e-01   2.313 0.020697 *
## VisitorTypeOther             -4.218e-01  4.950e-01  -0.852 0.394124
## VisitorTypeReturning_Visitor -5.571e-01  1.021e-01  -5.454 4.93e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                              edf Ref.df   Chi.sq p-value
## s(ProductRelated_Duration) 4.247  5.334   66.827 < 2e-16 ***
## s(BounceRates)             2.500  3.165    9.364 0.03368 *
## s(ExitRates)               5.272  6.367   20.340 0.00293 **
## s(PageValues)              8.861  8.982 2114.511 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## R-sq.(adj) =  0.423   Deviance explained = 42.4%
## UBRE = -0.49469  Scale est. = 1         n = 12330
```

5.1 Interpretation

Our GAM model is a logistic regression and thus predicts changes in log-odds of the outcome which are difficult to interpret. Taking the exponentiation of the estimates transforms the log-odds into odds ratio, which can be expressed as percentages and are more intuitive to understand. The significant predictors of the model can be interpreted as follows while holding all other variables constant.

▶ *Click to see the significant effects in percentage*

Intercept (Estimate: -1.8244, p < .001) The intercept indicates that if all continuous predictors are set to zero, the expected odds of *Revenue* would be on average 14% in *August* for a *New_Visitor* with *TrafficType1*.

Administrative (Estimate: -0.0566, p < .001) For each additional unit of *Administrative*, the odds of *Revenue* decrease on average by 5.5%. This suggests that increased engagement with administrative pages is associated with slightly lower odds of generating revenue.

Month Compared to *August*, the odds of *Revenue* are on average 53% lower in *December*, 82% lower in *February*, 48% lower in *March*, 55% lower in *May*, and 62% higher in *November.* This indicates that relative to August, November is associated with the highest odds of generating revenue and February with the lowest.

TrafficType Compared to *TrafficType1*, the odds of *Revenue* are on average 29% higher for *TrafficType2*, 103% higher for *TrafficType8*, 64% higher for *TrafficType10*, 87% higher for *TrafficType11*, 39% lower for *TrafficType13*, 1489% higher for *TrafficType16* and 91% higher for *TrafficType20*. Looking at the data, there are only three observations for *TrafficType16* with a percentage of revenue of 33.3%, which is the highest among all traffic types (with an average of 14%). Thus, this result should be interpreted with caution.

VisitorType Compared to a *New_Visitor*, the odds of *Revenue* are on average 43% lower for a *Returning_Visitor*. This suggests that returning visitors are significantly less likely to make a purchase compared to new visitors.

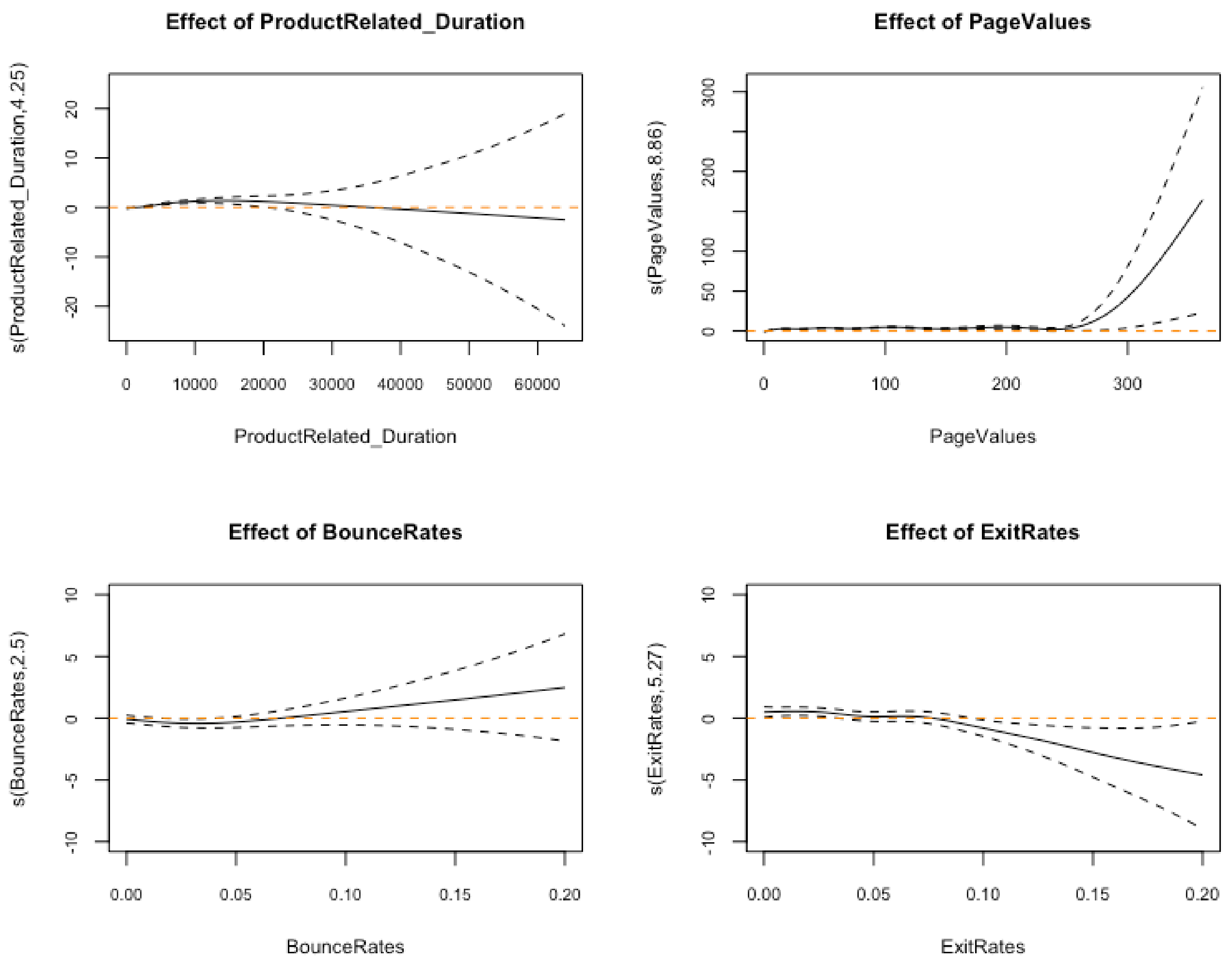
s(ProductRelated_Duration) (edf: 4.247, p < .001) *ProductRelated_Duration* has a strong, complex non-linear effect on *Revenue*. There is a positive effect on the log-odds of revenue for a duration of up to about 35’000 seconds with a peak at around 15’000 seconds. After 35’000 seconds, the effect on the log-odds of revenue is negative and continues to decrease gradually as duration increases. This suggests that visitors who spend a moderate amount of time on product pages are more likely to make a purchase, while a very long duration may reflect hesitation to make a purchase.

s(PageValues) (edf: 8.861, p < .001) *PageValues* has a strong, very complex non-linear relationship with *Revenue*. Page values up to around 250, have a slightly positive and relatively stable effect on the log-odds of generating revenue. For higher page values, the effect increases sharply. This suggests that users visiting pages with higher value are significantly more likely to make a purchase.

s(BounceRates) (edf: 2.500, p < .05) *BounceRates* has a moderate non-linear effect on *Revenue*. At low bounce rates below around 0.07, the log-odds of generating revenue are slightly negative. This may reflect visitors who browse multiple pages without a strong intent to buy. As bounce rates increase further, the log-odds of generating revenue also increase. This suggests that visitors intending to make a purchase leave the website quickly after landing on product or checkout pages and completing the purchase.

s(ExitRates) (edf: 5.272, p < .01) *ExitRates* has a complex non-linear relationship with *Revenue*. Low exit rates until around 0.08 have a slightly positive effect on the log-odds of revenue. However, as the average exit rate of pages increases, the log-odds of generating revenue decline. This suggests that visitors who tend to view pages with high exit rates are less likely to complete a purchase.

▶ *Click to see the code for the plots of the smooth terms*



5.2 Model Performance

The AIC (Akaike Information Criterion) balances model fit with model complexity, penalizing models that include more parameters without substantial improvement in explanatory power. The simpler GAM is preferred, since it has a lower AIC compared to the full and reduced model. The R-squared value indicates that the simple GAM can explain around 42.3% of the variance in the log-odds of revenue.

In terms of predictive performance, the simple GAM achieves a high overall accuracy of approximately 89.5%, just below the full GAM with 89.6%. The model is quite good at predicting sessions that do not lead to revenue, as reflected by its high specificity of 95.4%. However, it is not so good at predicting sessions that lead to revenue, as reflected by a moderate sensitivity of 57.3%.

▶ *Click to see the code for the GAM performance metrics*

Model	Accuracy	Sensitivity	Specificity	AUC	AIC	R.squared
GAM full	0.896	0.576	0.954	0.918	6260	0.424
GAM reduced	0.895	0.574	0.954	0.918	6238	0.424
GAM simple	0.895	0.573	0.954	0.917	6230	0.423

Fabienne Bölsterli took the lead in the GAM section.

6 Generalised Linear Models: Binomial

Using all possible predictors to predict *Revenue*, a full generalized linear model (GLM) for binomial data was fit. As only a part of the variables were significant, a smaller model was fit containing only the significant variables.

▶ *Click to see the full binomial GLM*

▶ *Click to see the reduced binomial GLM*

As only one level of *Browser* showed a significant effect in the binomial GLM, we decided to remove it for a simpler model. The simple model has a smaller AIC, which supports our decision.

▶ *Click to see the simple binomial GLM model*

```
##
## Call:
## glm(formula = Revenue ~ ProductRelated_Duration + ExitRates +
##     PageValues + Month + TrafficType + VisitorType, family = binomial,
##     data = df)
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -1.763e+00  1.929e-01  -9.140  < 2e-16 ***
## ProductRelated_Duration       1.074e-04  1.356e-05   7.922 2.33e-15 ***
## ExitRates                    -1.682e+01  1.663e+00 -10.116  < 2e-16 ***
## PageValues                    8.151e-02  2.396e-03  34.021  < 2e-16 ***
## MonthDec                     -7.293e-01  1.861e-01  -3.918 8.92e-05 ***
## MonthFeb                     -1.801e+00  6.425e-01  -2.802 0.005073 **
## MonthJul                      1.045e-01  2.188e-01   0.478 0.632779
## MonthJune                    -3.407e-01  2.770e-01  -1.230 0.218711
## MonthMar                     -5.973e-01  1.842e-01  -3.244 0.001180 **
## MonthMay                     -5.671e-01  1.707e-01  -3.322 0.000895 ***
## MonthNov                      4.398e-01  1.666e-01   2.640 0.008289 **
## MonthOct                     -6.962e-02  2.038e-01  -0.342 0.732606
## MonthSep                     -2.185e-02  2.125e-01  -0.103 0.918121    
## TrafficType2                  1.808e-01  9.387e-02   1.926 0.054096 .  
## TrafficType3                 -2.483e-01  1.229e-01  -2.021 0.043275 *  
## TrafficType4                  3.842e-02  1.395e-01   0.275 0.782973    
## TrafficType5                  2.359e-01  2.128e-01   1.109 0.267547    
## TrafficType6                 -9.339e-02  1.972e-01  -0.474 0.635819    
## TrafficType7                  3.413e-01  4.719e-01   0.723 0.469516    
## TrafficType8                  5.737e-01  1.771e-01   3.238 0.001202 ** 
## TrafficType9                 -2.821e-02  6.693e-01  -0.042 0.966378    
## TrafficType10                 3.485e-01  1.651e-01   2.111 0.034793 *  
## TrafficType11                 4.024e-01  2.097e-01   1.919 0.054954 .  
## TrafficType12                -1.189e+01  1.455e+03  -0.008 0.993484    
## TrafficType13                -6.040e-01  1.979e-01  -3.052 0.002277 ** 
## TrafficType14                -4.996e-01  1.097e+00  -0.456 0.648749    
## TrafficType15                -1.234e+01  2.200e+02  -0.056 0.955258    
## TrafficType16                 1.927e+00  1.235e+00   1.561 0.118554    
## TrafficType17                -1.178e+01  1.455e+03  -0.008 0.993541    
## TrafficType18                -1.251e+01  4.444e+02  -0.028 0.977549    
## TrafficType19                -1.128e+00  1.423e+00  -0.793 0.427732    
## TrafficType20                 4.503e-01  2.606e-01   1.728 0.083911 .  
## VisitorTypeOther             -5.875e-01  5.453e-01  -1.077 0.281337    
## VisitorTypeReturning_Visitor -1.999e-01  8.992e-02  -2.223 0.026201 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 10624.8  on 12329  degrees of freedom
## Residual deviance:  7116.1  on 12296  degrees of freedom
## AIC: 7184.1
## 
## Number of Fisher Scoring iterations: 14
```

6.1 Interpretation

The binomial GLM is again a logistic regression, predicting changes in log-odds of the outcome which are difficult to interpret. Taking the exponentiation of the estimates transforms the log-odds into odds ratio, which can be expressed as percentages and are more intuitive to understand. The significant predictors of the model can be interpreted as follows while holding all other variables constant.

▶ *Click to see the significant effects in percentage*

Intercept (Estimate: -1.7627, p < .001) The intercept indicates that if all continuous predictors are set to zero, the expected odds of *Revenue* would be on average 14.7% in *August* for a *New_Visitor* with *TrafficType1*.

ProductRelated_Duration (Estimate: 0.0001, p < .001) For each additional unit of *ProductRelated_Duration*, the odds of *Revenue* increase on average by 0.01%. This indicates that a longer duration on product-related web pages is associated with increased odds of generating revenue, it increases by 0.01% per second.

ExitRates (Estimate: -16.8234, p < .001) For each additional unit of *ExitRate*, the odds of *Revenue* decrease dramatically by almost 100% This suggests that visitors who visit pages with high exit rates are extremely unlikely to generate revenue.

PageValues (Estimate: 0.0815, p < .001) For every additional unit of *PageValues*, the odds of *Revenue* increase on average by 8.5%. This suggests that visiting more high-value pages is associated with increased odds of generating revenue.

Month Compared to *August*, the odds of *Revenue* are on average 52% lower in *December*, 83% lower in *February*, 45% lower in *March*, 43% lower in *May*, and 55% higher in *November*. This indicates that relative to August, November is associated with the highest odds of generating revenue and February with the lowest.

TrafficType Compared to *TrafficType1*, the odds of *Revenue* are on average 22% lower for *TrafficType3*, 77% higher for *TrafficType8*, 42% higher for *TrafficType10* and 45% lower for *TrafficType13.*

VisitorType Compared to a *New_Visitor*, the odds of *Revenue* are on average 18.1% lower for a *Returning_Visitor.* This suggests that returning visitors are significantly less likely to generate revenue compared to new visitors.

6.2 Model Performance

Comparing the models using AIC, the simple binomial GLM has the lowest AIC. This indicates a better balance of fit and complexity compared to the full and reduced binomial GLM model.

In terms of predictive performance, the simple binomial GLM achieves a high overall accuracy of approximately 88.5%, which is only slightly below that of the full binomial GLM with 88.6%. The model is quite good at predicting sessions that do not lead to revenue, as reflected by its very high specificity of 97.7%. However, it is bad at predicting sessions that lead to revenue, as reflected by a low sensitivity of 38.7%.

▶ *Click to see the code for the binomial GLM performance metrics*

Model	Accuracy	Sensitivity	Specificity	AUC	AIC
GLM Binomial full	0.886	0.390	0.976	0.900	7215
GLM Binomial reduced	0.885	0.387	0.976	0.899	7193
GLM Binomial simple	0.885	0.387	0.977	0.898	7184

Fabienne Bölsterli took the lead in the GLM Binomial section.

7 Generalised Linear Models: Poisson

For a generalized linear model following the Poisson Regression, we need output variables that are non negative. Hence, only the variables *Administrative*, *Informational* and *ProductRelated* can be considered as they are the only true count variables that do not violate the Poisson conditions:

$$Y \in \{0, 1, 2, \ldots\}, \quad \mathrm{Var}(Y) = \mathbb{E}(Y).$$

In a first step, we look at how the variance of those three variables compares to their mean, as the a Poisson model assumes they are equal.

$$\mathbb{E}(Y) = \lambda, \quad \mathrm{Var}(Y) = \lambda$$

Dispersion

```
##          Administrative Informational ProductRelated
## Mean           2.315166     0.5035685       31.73147
## Variance      11.034250     1.6132973     1978.07039
```

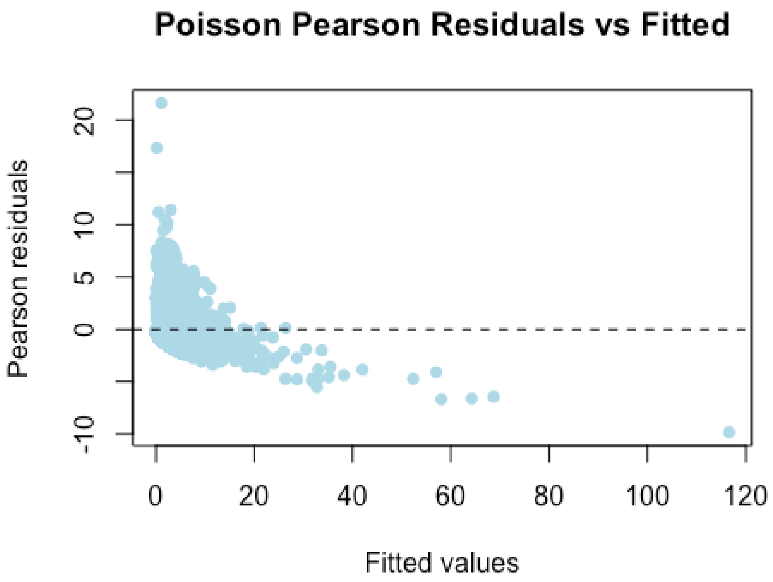
Clearly, there is overdispersion across all three variables (*Administrative*: 11 > 2.3, *Informational*: 1.6 > 0.5, *ProductRelated*: 1978.1 > 31.7), indicating that there are additional factors or patterns that a Poisson model cannot account for.

However, for the purpose of this project, we continue with performing a GLM Poisson using *Administrative* as the outcome variable. This decision is based on the fact that *ProductRelated* shows the highest level of overdispersion by far and *Informational*, although less dispersed, takes the value 0 in nearly 80% of observations, which could lead to a zero-inflated model. *Administrative*, on the other hand, has a moderate level of dispersion and a relatively balanced count distribution.

▶ *Click to see the Poisson GLM*

The GLM Poisson on the *Administrative* count showed highly significant predictors such as *Administrative_Duration*, *Informational*, *ProductRelated*, *BounceRates* and *ExitRates*. However, it also shows strong overdispersion (deviance/df $\approx$ 2.51; Pearson χ^2/df $\approx$ 2.58), violating the Poisson assumption that *Var(Y) = E(Y)*.

▶ *Click to see the code to plot the residuals against fitted values*



A visualization of the model predictions against the actual data clearly highlights the overdispersion and the poor fit of the Poisson model. The data show too much variability and a few extreme outliers for the Poisson model to handle. However, since overdispersion is the rule rather than the exception when dealing with count data, we allow the model to account for it by estimating an additional dispersion parameter using a quasi-Poisson GLM.

7.1 GLM Quasi-Poisson

To address this extra variability, we fit a quasi-Poisson model, which keeps the same structure but scales the variance by an estimated dispersion parameter (2.58), widening the margin of error in our results. This should reflect the fact that the observed counts vary more than a standard Poisson model assumes (*Var(Y)/E(Y) = 1*).

▶ *Click to see quasi-Poisson GLM*

Under the quasi-Poisson adjustments, only a few key predictors remain highly significant, confirming that we have removed all those predictors whose significance was only due to underestimated variability and which are no longer significant once we account for the overdispersion (2.58).

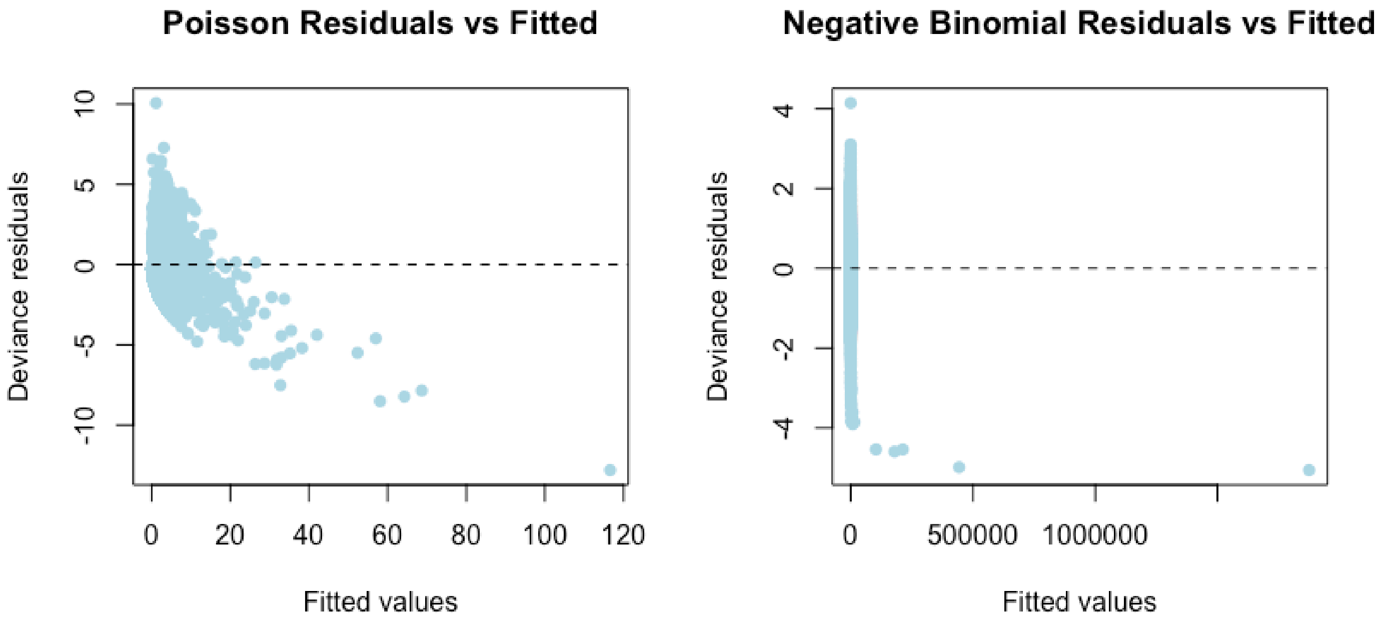
7.2 GLM Negative Binomial

Finally, we fit a Negative Binomial GLM, which extends the Poisson model by introducing a dispersion parameter θ to explicitly model the variance beyond the Poisson assumption. Thus, we shift the variance function from the linear form used by the quasi-Poisson model to the quadratic form used by the Negative Binomial model.

$$\mathrm{Var}(Y) = \phi\,\mu \qquad \longrightarrow \qquad \mathrm{Var}(Y) = \mu + \frac{\mu^2}{\theta}$$

▶ *Click to see the negative-binomial GLM*

▶ *Click to see the code to plot the residuals against fitted values*



Under the Negative Binomial GLM, several coefficients (*OperatingSystems*, *VisitorType*, *Weekend*, etc.) lose significance, reflecting its ability to reduce outlying predictors. The relatively low dispersion parameter θ (around 1.21) indicates substantial overdispersion, meaning that variance increases quadratically rather than linearly with the mean. Comparing the AIC scores of the Poisson GLM and the Negative Binomial GLM, we clearly see an improvement in balancing model fit and complexity indicated by a lower AIC. Hence, the Negative Binomial GLM (AIC = 41'558) is the preferred count model for *Administrative*. This is also visible when plotting the predicted versus actual values for both models; the Negative Binomial GLM has no funnel form and therefore fewer outliers than the Poisson GLM.

7.2.1 Interpretation

While several coefficients are statistically significant and each additional unit is associated with an increase or decrease in the *Administrative* page count (e.g. +0.0041 for *Administrative_Duration*, +0.1017 for *Informational*, and +0.0058 for *ProductRelated*), the most interesting variables are *BounceRates* and *ExitRates*.

BounceRates (Estimate: 4.6428, p < .001) The model suggests that sessions with higher bounce rates tend to have more views of administrative pages. This may imply that when customers encounter friction (e.g. broken link or unclear checkout), they do not immediately leave the website. Instead, they might navigate to help, FAQ, or support pages (i.e. *Administrative* pages) trying to resolve the issue.

ExitRates (Estimate: –21.8016, p < .001) The negative coefficient for *ExitRates* indicates that sessions with higher *ExitRates* tend to have fewer views of administrative pages. This suggests that once visitors decide to leave, they exit the site directly without navigating to administrative pages. From a business perspective, this could be a warning sign as we may be losing customers before they reach any help resources.

David Gerner took the lead in the GLM Poisson section.

8 Support Vector Machines (SVM)

Next, we want to predict *Revenue* using a Support Vector Machine model. Prior to modeling, the data needs to be prepared. First, we transform the binary target variable *Revenue* into a factor with the levels "No" and "Yes" to indicate to the SVM that this is a classification task. Then, we dummy encode all categorical predictor variables and then center and scale all variables. This ensures that predictors measured on larger scales do not dominate the model.

▶ *Click to see the data preparation*

Finally, we set up an unbiased test by splitting the data so that 80% is used for model fitting and 20% for testing the model. This split is stratified in order to maintain the proportional distribution of the "No" and "Yes" classes in both the training and testing data set.

▶ *Click to see the test and training split*

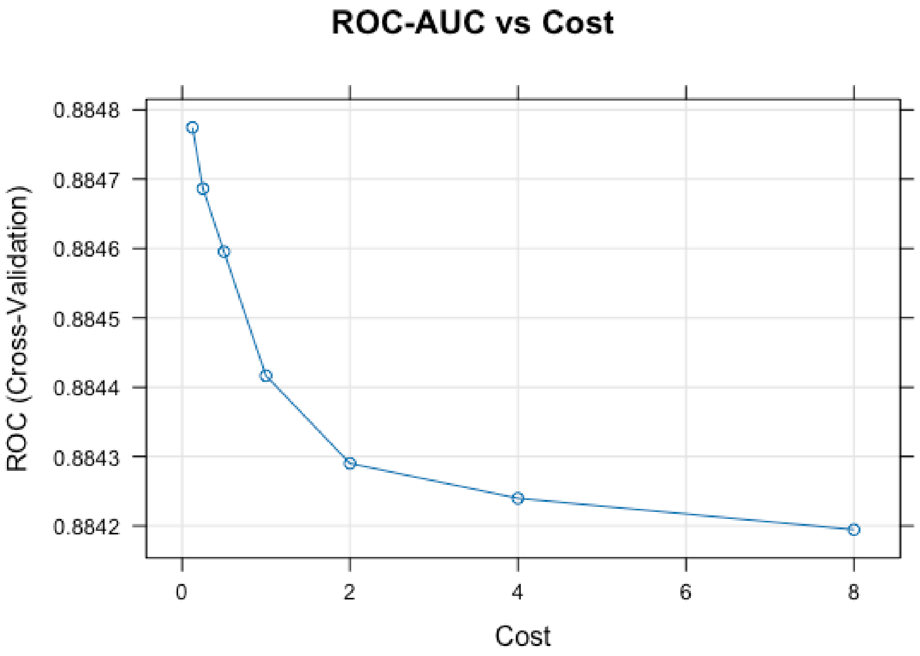
8.1 SVM Model

First, we define a 3-fold cross-validation to smooth out variability and optimize the ROC-AUC metric. Due to time constraints, the SVM only runs three resamples, from which the cost value with the highest average ROC-AUC is selected. We train the SVM model on the training data and corresponding labels using the `caret::train()` function with a linear kernel and the previously defined hyperparameter tuning settings.

▶ *Click to see the hyperparameter setting and model training*

From the cross-validation results, we extract the ROC-AUC scores for all cost values and visualize them. A cost value of 0.125 seems to give the best separation between buyers and non-buyers with a ROC-AUC of approximately 0.8848. We then retrain the SVM model on the entire 80% training set using this optimal cost value and predict the class probabilities on the held-out 20% test set.

▶ *Click to see the code for the ROC curve*



8.2 Model Performance

8.2.1 Confusion Matrix

The confusion matrix provides a clear overview of the model performance. Overall, the model shows a strong accuracy by correctly classifying about 89% of all sessions. The model is quite bad at correctly identifying buyers, as indicated by a low sensitivity of about 36%, meaning it mislabels abound 64% of the buyers as non-buyers. On the other hand, the model is very good at detecting non-buyers as indicated by a very high specificity of around 98.7%.

When the model predicts that a session has revenue, it is accurate about 83.4% of the time, as indicated by the positive predictive value. Similarly, when it predicts that a session will not result in revenue, it is correct about 89.4% of the time as indicated by the negative predictive value.

The model has a moderate balanced accuracy of about 67.2%, reflecting its performance across both groups by averaging how well it detects buyers (sensitivity) and non-buyers (specificity).

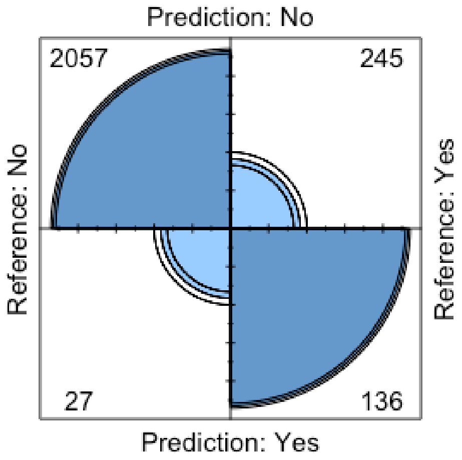
▶ *Click to see the code for the confusion matrix*

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   No  Yes
##        No  2057  245
##        Yes   27  136
##                                           
##                Accuracy : 0.8897          
##                  95% CI : (0.8766, 0.9018)
##     No Information Rate : 0.8454          
##     P-Value [Acc > NIR] : 1.404e-10       
##                                           
##                   Kappa : 0.449           
##                                           
##  Mcnemar's Test P-Value : < 2.2e-16       
##                                           
##             Sensitivity : 0.35696         
##             Specificity : 0.98704         
##          Pos Pred Value : 0.83436         
##          Neg Pred Value : 0.89357         
##              Prevalence : 0.15456         
##          Detection Rate : 0.05517         
##    Detection Prevalence : 0.06613         
##       Balanced Accuracy : 0.67200         
##                                           
##        'Positive' Class : Yes             
## 
```

3.2.2 Fourfold-Plot

The fourfold plot provides a visual overview on how many buyers and non-buyers were correctly identified, how many buyers were missed, and how many non-buyers were wrongly labeled as buyers. It makes it easy to see how often the model predictions were correct versus incorrect or both groups.

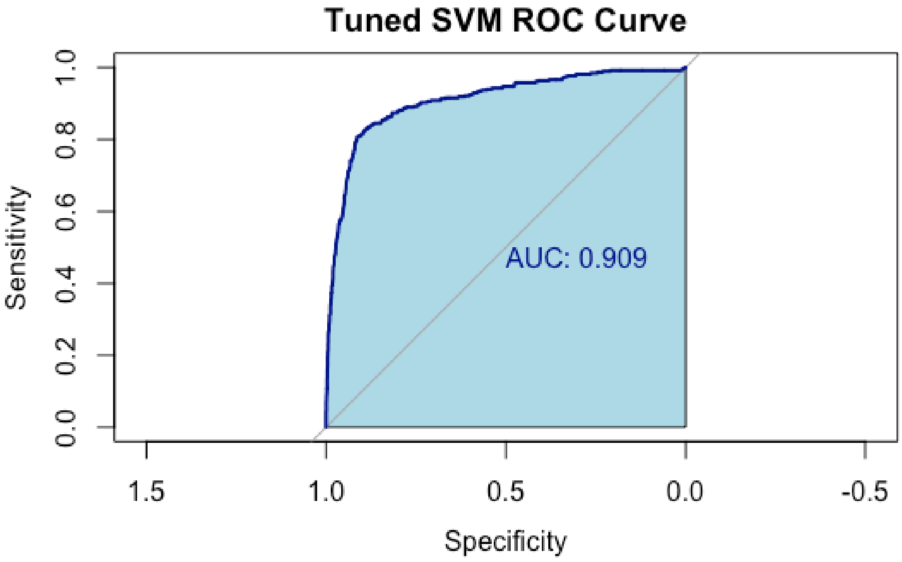
▶ *Click to see the code for the fourfold plot*



8.2.3 ROC Curve

Other than the confusion matrix, we also evaluate our SVM based on how confident it needs to be before predicting someone as a buyer, and look at the trade-off between correctly identifying buyers and producing false positives. The area under the ROC curve (AUC) is 0.909, meaning there is a 90.9% chance that a randomly selected purchaser will receive a higher predicted probability than a randomly selected non-purchaser. This indicates that our tuned linear SVM with a cost parameter of 0.125 separates buyers from non-buyers very effectively on new data.

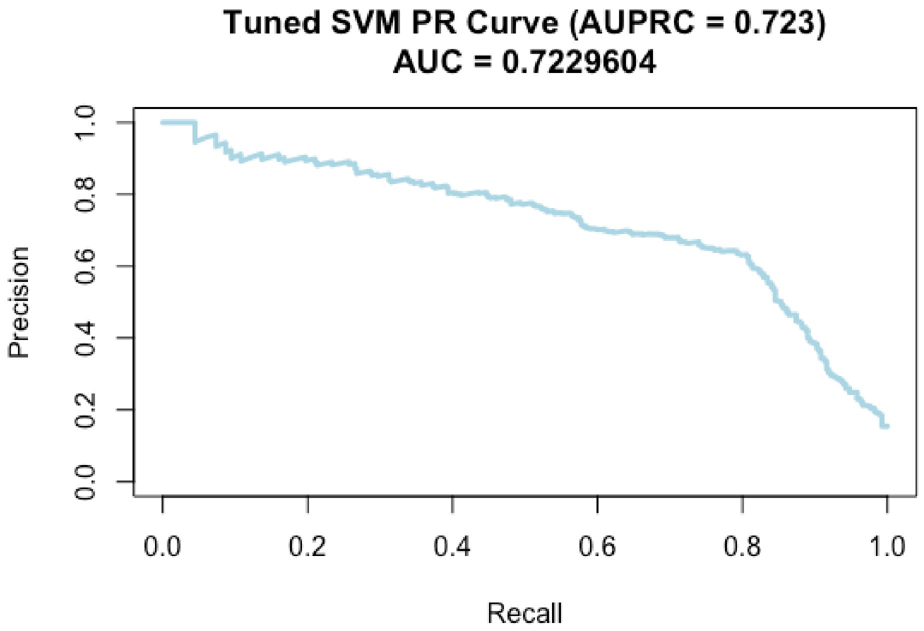
▶ *Click to see the code for the ROC Curve*



8.2.4 Precision–Recall Curve

Because buyers are the smaller group in our imbalanced data set, we shift the focus to the ability of detecting actual buyers using the Precision-Recall (PR) curve. The area under the PR curve (AUPRC) is 0.72. This means that the model correctly identifies real buyers 72% of the time on average. This gives a more realistic picture of how well the model works when buyers are rare.

▶ *Click to see the code for the PR Curve*



David Gerner took the lead in the Support Vector Machines section.

9 Neural Network

We now want to predict *Revenue* by using a neural network. The class imbalance of our target variable might cause the neural network to be biased towards the majority class without *Revenue*. In a first step, we include all predictors in the neural network. Since neural networks cannot handle factors directly, we converted all categorical variables into numerical variables using one-hot encoding.

▶ *Click to see the code for the one-hot encoding*

In a second step, we divided our data set into training and testing sets, using a 80% train and 20% test split. We used stratified sampling to ensure that the distribution of our target variable *Revenue* in both the training and testing set matched our original data set.

▶ *Click to see the code to split the training and testing data*

As the next step, we trained and evaluated the neural network on the training data set. The model consists of one hidden layer with 16 neurons.

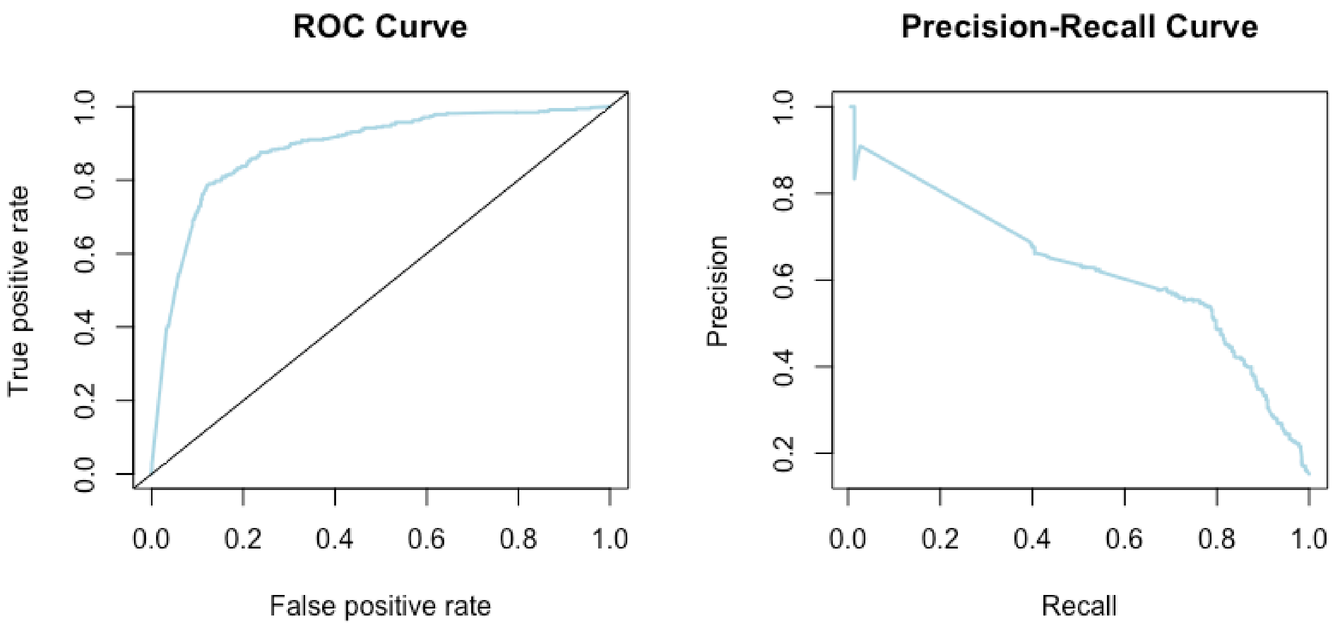
▶ *Click to see the code to train the neural network*

We then used the trained neural network to make predictions on the testing data set. To analyse the model performance, we compared its predictions with the actual values using a confusion matrix.

▶ *Click to see code for the model predictions and confusion matrix*

The confusion matrix shows that our model has an overall accuracy of 88%, representing the percentage of correct predictions. The sensitivity is only 53.7%, indicating that our model is not very effective at detecting positive cases of *Revenue*. In contrast, the specificity is high with 94.2%, meaning our model is very effective at identifying cases where there is no *Revenue*. The high specificity and low sensitivity suggest that the model is biased towards predicting non-buyers correctly at the expense of missed buyers. This is likely due to the class imbalance in the sample.

▶ *Click to see code for the ROC and PR curves*



To asses the quality of our model, we use the Receiver Operating Characteristic (ROC) and the Precision-Recall (PR) curve. For the ROC curve we plot the true positive rate (sensitivity) against the false positive rate (1 - specificity). The plot shows that a high true positive rate (around 0.8) can be achieved when lowering the classification threshold which would mean accepting more false positives.

The PR curve, on the other hand, shows the trade-off between precision and recall for the binary classifier *Revenue* at different threshold settings. The curve peaks around 0.75 precision at low recall values. This means the model can be very precise when it is conservative about making positive predictions. As recall increases, the precision decreases.

Finally, we look at the AUC value of the model. The AUC is 0.886 which indicates a good discrimination ability of the model between classes. Yet the current threshold (low false positive rate) prioritizes cost control (e.g. not wasting marketing efforts on unlikely purchasers) over leveraging potential revenue opportunities. Assuming the cost of marketing to be relatively low relative to the potential value of additional revenue, lowering the threshold seems a valid option. This could increase recall significantly while maintaining reasonable precision.

▶ *Click to see code for the AUC*

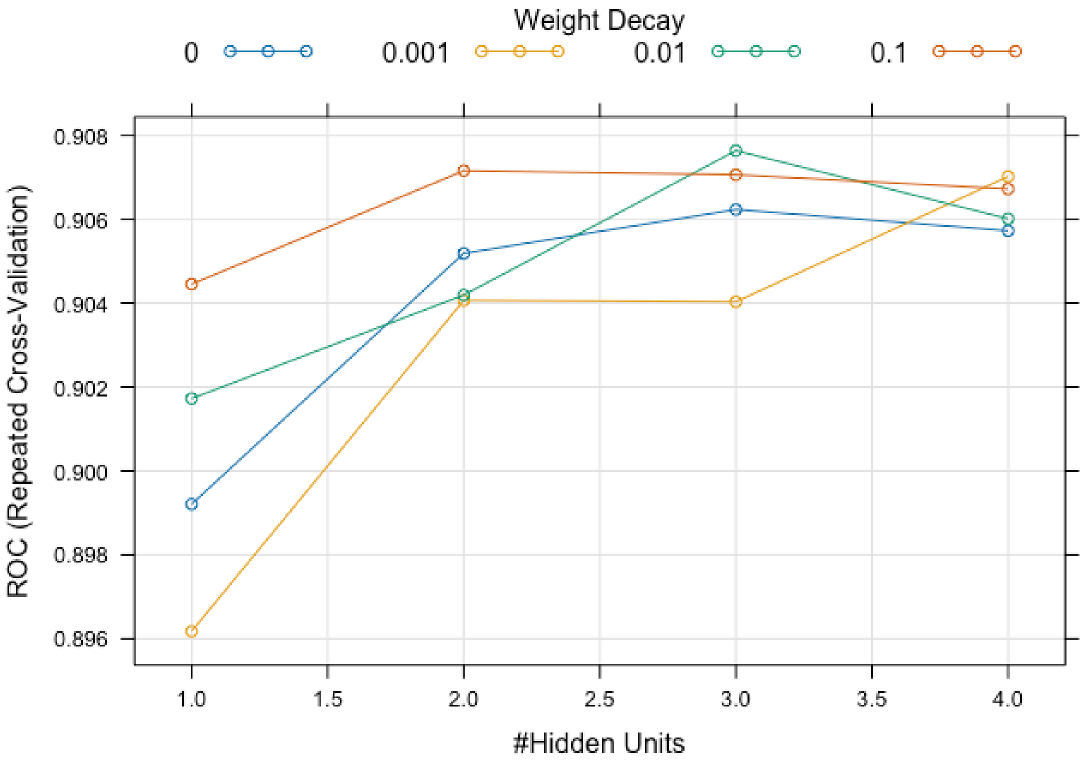
9.1 Neural Network Optimization and Cross-Validation

To further evaluate the neural network’s performance and reduce the risk of overfitting, we use a 5-fold cross-validation repeated 10 times. This means that the data set was split into five parts, with the model trained and validated on different combinations of these splits multiple times. To optimize the model, two hyperparameters were tuned: the size of the hidden layer and weight decay. A tuning grid was defined to explore multiple values for each hyperparameter, allowing the training process to identify the combination that results in the best performance based on the ROC metric. To handle class imbalance in the target variable, the Synthetic Minority Over-Sampling Technique (SMOTE) from the `{themis}` package was used. SMOTE artificially increases the number of minority class examples during training, helping the model to better learn patterns from both classes. To apply SMOTE, *Revenue* had to be transformed into a factor rather than binary variable.

▶ *Click to see the code for the variable encoding*

▶ *Click to see the code for the cross-validation and optimization of the neural network*

The following plot displays the results of the hyperparameter tuning. Each line represents a different value of weight decay. The x-axis displays the number of hidden units, while the y-axis shows the mean ROC score across resamples. The best performance is achieved with 3 hidden units and a weight decay value of 0.01, resulting in a ROC score close to 0.908.



▶ *Click to see the results for all models*

```
##    size decay       ROC      Sens      Spec       ROCSD     SensSD     SpecSD
## 11    3  0.01 0.9076386 0.7974439 0.8673187 0.008428907 0.02457735 0.01211504
```

We now use the final model to make predictions on the test data set and compare its performance to the initial neural network. The AUC has improved to 0.923 compared to 0.886 in the initial model. The confusion matrix shows a slightly lower overall accuracy of 86.17% compared to 88.04%. While sensitivity is significantly better with 81.6% compared to 53.7%, the specificity is lower with 87% compared to 94.2%. Since our main goal is to identify customers who generate revenue, the substantial improvement in sensitivity is worth the small drop in specificity.

▶ *Click to see the confusion matrix and AUC*

Valérie Lüthi took the lead in the neural network section.

10 Conclusion

▶ *Click to see the functions that were used to create the comparison table*

Model	Target.Variable	Accuracy	Sensitivity	Specificity	AUC	AIC	R.squared
Linear Model	log(PageValues)	•	•	•	•	33553	0.452
GLM Poisson	Administrative	•	•	•	•	51051	•
GLM Negative Binomial	Administrative	•	•	•	•	41558	•
GAM (simple)	Revenue	0.895	0.573	0.954	0.917	6230	0.423
GLM Binomial (simple)	Revenue	0.885	0.387	0.977	0.898	7184	•
Support Vector Machine	Revenue	0.89	0.357	0.987	0.909	•	•
Neural Network	Revenue	0.88	0.537	0.942	0.886	•	•
Neural Network optimized	Revenue	0.862	0.816	0.87	0.922	•	•

Since our target variable *Revenue* was binary we had to choose alternative continuous variables for our linear and Poisson models. Therefore, those models are not directly comparable to the classification models predicting *Revenue*.

Among the classification models, the simple GAM achieved good overall accuracy and specificity, with moderate sensitivity. This illustrates that a simpler and more interpretable model can perform well. The GLM Binomial, SVM and Neural Network all showed very high specificity at the cost of very low sensitivity, making them conservative in predicting buyers. This was likely due to the class imbalance in *Revenue*. To address this issue, we optimized the neural network using SMOTE to balance the classes. This led to a significant improvement in sensitivity, helping to better identify potential buyers, while only slightly reducing specificity.

From a business perspective, high sensitivity is crucial to accurately identify online buyers to target marketing efforts effectively and maximize *Revenue*, even if this means to accept some false positives. The optimized neural network offers a balanced approach to successfully predict *Revenue* in online-shopping sessions.

11 Use of Generative AI

In this assignment, we used generative AI in the form of ChatGPT as a supportive tool for coding and debugging. The AI has been particularly helpful in explaining functions that we were not familiar with and in helping to debug coding errors.

We noticed that the specificity of our prompts had a big impact on the quality of the responses and that it is essential to evaluate the AI-generated answers thoroughly.

For this project, we worked with publicly available data. Working in a professional setting as a data scientist, this is often not the case. This makes it crucial to be extremely cautious about what information is shared with generative AI tools.

References

1. Sakar, C. O., Polat, S. O., Katircioglu, M. A., & Kastro, Y. (2019). Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks. *Neural Computing and Applications*, *31*(10), 6893–6908. https://doi.org/10.1007/s00521-018-3523-0
2. Sakar, C., & Kastro, Y. (2018). *Online Shoppers Purchasing Intention Dataset*. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset