Exploratívna analýza odchodovosti zákazníkov: Využitie vizualizačných nástrojov v Pythone

Exploratory Analysis of Customer Churn: Utilizing Visualization Tools Available for Python

Michal Bogár[1]

Abstrakt

Exploratívna dátová analýza sa ukázala byť dôležitým nástrojom pri snahe lepšie pochopiť odchodovosť zákazníkov telekomunikačného operátora, a to skúmaním údajov o viac ako 7000 zákazníkoch, pričom boli využité nástroje z ekosystému Python, najmä Jupyter Lab, spolu s knižnicami ako pandas, NumPy, Seaborn a Plotly. Tento článok vychádza z myšlienok, ktoré koncipoval John Tukey, a zdôrazňuje význam vizualizácie údajov na odhalenie skrytých vzorov a vzťahov v dátach, ktoré ovplyvňujú správanie zákazníkov. Postupnou analýzou rôznych zákazníckych atribútov, od tých demografických, až po premenné, ktoré popisujú ich predplatené služby sme identifikovali trendy vedúce k odchodovosti, vrátane výrazného vplyvu dĺžky viazanosti, rodinného stavu zákazníka a citlivosti na cenu produktov. Získané výsledky poukazujú na to, že zákazníci so zmluvami bez viazanosti a zákazníci bez partnerov alebo detí sú náchylnejší prerušiť zmluvu. Vytvorené interaktívne grafy poskytujú nielen intuitívne prehľady, ale pomáhajú aj pri hlbšom skúmaní dát, čím vytvárajú solídny základ pre prediktívne modelovanie. Táto analýza zdôrazňuje dôležitosť exploratívnej analýzy pri formulovaní účinných retenčných stratégií na udržanie zákazníkov a ponúka možnosti na pokračovanie vo výskume využívajúc pokročilé analytické techniky, ako sú strojové učenie a kohortová analýza, na predikovanie odchodovosti.

Kľúčové slová

Exploratívna analýza, EDA, Odchodovosť zákazníkov, Python

Abstract

The role of Exploratory Data Analysis (EDA) proved to be important in understanding customer churn in telecommunications by examining a dataset of over 7000 customers, utilizing tools from the Python ecosystem, particularly Jupyter Lab, along with libraries such as pandas, NumPy, Seaborn, and Plotly. Based on John Tukey's foundational concepts of EDA, this paper emphasizes the importance of visualizing data to uncover patterns and relationships that influence customer behavior. By systematically analyzing various demographic and service-related attributes, we identified significant trends, including the disproportionate impact of contract types, customer demographics or marital status, and price sensitivity on churn rates. Obtained results highlight that customers with month-to-month contracts, and those without partners or children, exhibit higher churn tendencies. The generated interactive visualizations provide not only intuitive insights but also assist in a deeper exploration of hidden anomalies and trends, setting a robust groundwork for predictive modeling. Ultimately, this analysis underlines the necessity of EDA in formulating effective customer retention strategies, offering ways for future research employing advanced analytical techniques like machine learning and cohort analysis to predict and mitigate churn.

[1] University of Economics in Bratislava, Faculty of Economic Informatics, Department of Operations Research and Econometrics, Dolnozemská cesta 1, 852 35 Bratislava, michal.bogar@euba.sk.

Keywords
Exploratory Data Analysis, EDA, Customer Churn, Python

JEL classification
L86, C88

1 Introduction

Exploratory Data Analysis (EDA) is a critical step in understanding data as stated by John W. Tukey (1977) in his influential book that introduced the concept of exploratory analysis. According to Tukey, the key to EDA is knowing where to look and using the right tools to uncover insights in data. This paper focuses on analyzing customer churn in a telecommunications dataset, that comprises over 7000 customers, including a variety of attributes ranging from demographic details to service preferences. By leveraging Jupyter Lab with Python and popular libraries such as pandas, NumPy, and Plotly, we aim to explore patterns in customer behavior and identify factors influencing churn. This analysis will provide a deeper understanding of the data by providing visually clear graphs that uncover key insights into customer behavior and churn patterns, setting the stage for potential predictive modeling.

2 The Role of Exploratory Data Analysis

The pioneer of exploratory analysis, John W. Tukey, in his 1977 book *Exploratory Data Analysis*, writes that exploratory analysis resembles detective work, where one must look in the right places using the right tools.

Nowadays, Exploratory Data Analysis became a crucial aspect of data science, developed to uncover patterns in data and to form hypotheses in research. EDA offers a framework separate from Confirmatory Data Analysis (CDA), which focuses on hypothesis testing. While CDA is critical for confirming well-defined theories, EDA helps in the initial stage where researchers can ask broader questions about their data, leading to insights that may influence subsequent analyses and models. But to clarify, even Tukey recognizes the importance of both CDA and EDA, as they complement each other, and neither would be as effective without the other one (Tukey, 1980).

At its core, EDA encourages willingness to question the data and utilizing graphical visualizations to help with the examination. Rather than holding strictly to predefined hypotheses, it helps data scientists to explore various propositions and investigate multiple potential explanations for observed patterns. This contrasts with CDA, which is more rigid and often focuses only to confirm or reject a specific hypothesis. Common EDA techniques include graphical displays such as dot plots, box plots, and kernel density estimates, which provide intuitive visual representations of data distributions. For instance, box plots offer a summary that effectively summarizes key elements of data distribution while highlighting outliers. Moreover, modern tools to create interactive graphics have emerged, allowing for dynamic exploration of datasets and enabling the identification of significant trends or anomalies (Behrens, 1997).

The role of Python in EDA is getting more prevalent, and today Python is used in a wide range of domains. Peng and colleagues (2021) focus on the use of Python in EDA for statistical modeling, where they identify the most common EDA tasks for statistical modeling. These tasks include general overview, correlation analysis, missing value analysis, univariate and bivariate analysis.

Graff and his team (2022) in turn describe the use of Python to analyze Twitter data, which allows to investigate trends in language usage and analyze events reflected in Twitter data. In addition, this also allows the analysis of the mobility of Twitter users based on the geolocation data of the tweets and the comparison of different variants of Spanish language in different countries.

Sahoo and colleagues (2019) describe a wider use of Python in EDA using libraries such as Pandas, Matplotlib and Seaborn. These libraries enable data cleaning, transformation, analysis, visualization and other tasks that are essential in the EDA process.

3 Methodology

The methodology for this paper outlines the steps and tools used to analyze customer churn. It begins by describing the dataset, its origins, and characteristics, followed by a detailed explanation of the development environment and libraries employed to process and visualize the data. The selected tools and techniques, including open-source Python libraries, notably Jupyter Lab for efficient code execution and visualization and key libraries including NumPy for handling multi-dimensional arrays, pandas for data manipulation, Seaborn for static data visualization, and Plotly for interactive charts, ensure efficiency and reliability in handling the data and generating insights.

3.1 Characteristics of the analyzed dataset

We will use a dataset compiled by a foreign technology company as the customer data of an unspecified operator that offers telephone, internet and television services. This dataset is available on kaggle.com, a website that focuses on data analytics, sharing datasets, organizing contests, and bringing together the analytics community. It contains anonymous information on more than 7000 customers, including data such as demographic characteristics, information about their subscribed products and services, monthly fees, length of their contracts, and much more. The customer base in the dataset is heterogeneous and provides an all-round view of the market segment, where we can assume a certain degree of imbalance in the dataset, as most customers usually stay with the operator (BlastChar, 2018).

3.2 Development environment and libraries

Considering all the available alternatives, open-source tools from the Python ecosystem are particularly fitting due to their usually free availability and widespread community support. Their performance and reliability are then highlighted even further by the fact that they are used by many specialists.

3.2.1 Jupyter Lab

The Jupyter project is a group of open-source programs including Jupyter Lab, a tool for writing and sharing source code or entire documents, called notebooks, suited for visualizing data and creating interactive elements such as charts and plots. Its use is widespread in the scientific and analytics community for a variety of workflows, including data cleaning, statistical analysis, and creating machine learning models. With the ability to run code in parts by dividing it into individual cells, it allows for logical separation of different parts of a project, which helps to test and demonstrate the procedures and algorithms more efficiently without having to run the entire laptop. It is primarily based on the Python programming language, but its name is based on three programming languages Julia, Python and R. In addition to these, it supports over 100 different languages such as Java, MATLAB and many others (Barba et al., 2019). In this paper, we have decided to use Jupyter Lab in combination with Python and several additional libraries to analyze the dataset, all of which are presented below.

3.2.2 Numpy

NumPy, is an open-source Python library that offers powerful tools for scientific computing. It provides tools for creating and working with multi-dimensional array objects, providing efficient operations with large volumes of data. It is therefore an integral part of the Python ecosystem of scientific libraries and serves as a foundation of many other libraries, as it supports a rich variety of mathematical operations necessary for solving complex analytical tasks (NumPy Developers).

3.2.3 pandas

The pandas library is one of the core open-source packages of the Python programming language, which is extremely helpful in data analysis and data manipulation. It provides robust data structures such as *Series*, for working with one-dimensional data structures, and *DataFrame*, for working with two-dimensional structures. This makes it an ideal tool for working with different types of datasets. More specifically, due to its abilities for handling missing data or converting data types, the pandas package became nearly ubiquitous in the field of data science (The pandas development team).

3.2.4 Seaborn

Seaborn is primarily a graphics library for creating graphs and other graphical elements, building on the foundations of the Matplotlib library and interfacing with data structures of the pandas package to simplify data visualization. It points out intuitive visualizations, which helps to explore and understand available data. Visualizing functions of the library work with data frames and arrays, performing all the required calculations and statistical aggregations internally to generate informative graphs (Waskom, 2021).

3.2.5 Plotly

Plotly is the second graphical library used in this paper, which, compared to Seaborn, can create interactive graphs. Generating graphs is done in an intuitive interface with good syntax and detailed styling options that range from size customization, text color and font size adjustments, to custom element layouts. It supports a large variety of different chart types, such as bar and pie charts, dendrograms, as well as various 3D charts. Moreover, it integrates seamlessly with scientific computing libraries such as Pandas and NumPy. The difference with most of the packages already mentioned is that this library is not based on a completely open-source license and needs to be licensed for commercial use (Di Méo, 2023).

4 Data file analysis and editing

Dataset used for the analysis was obtained from website Kaggle.com and contains 21 variables describing different customer attributes, which are represented by different types of variables. The name of the attribute, what information it describes and what values it can obtain is explained in more detail below.

- customerID - a unique customer identifier composed of both numbers and letters, formatted as an alphanumeric *string*.
- gender - identifies customer's gender, with values of "Female" for women and "Male" for men and thus represents a binary categorical variable.
- SeniorCitizen - defines whether the client is a pensioner, where "0" stands for no and "1" for yes. It acts as a binary categorical variable.
- Partner - indicates whether the customer has a partner. Possible values are "Yes" and "No", which shows that this is a binary categorical variable.

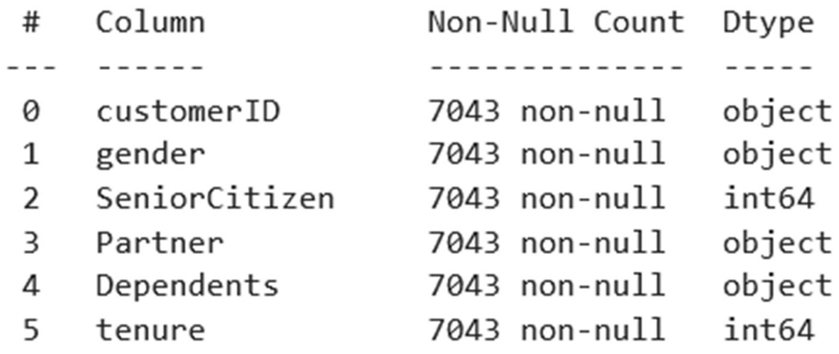
- Dependents - indicates whether the client has children. Allowed values are "Yes" and "No", and therefore it is another dichotomous variable.
- tenure - represents the number of months a customer has been with the company. Values are numeric and the data type is integer, so it is a numeric variable.
- PhoneService - determines whether the client has subscribed to a phone line. The values are either "Yes" or "No" for no service, so it is a dichotomous variable.
- MultipleLines - complements the attribute PhoneService and tells if the customer has multiple phone lines. Values can be "Yes", "No" or "No phone service", indicating that this is a nominal categorical variable.
- InternetService - defines the type of internet connection that the customer uses. The value can be "DSL", "Fiber optic" or "None". This is a nominal categorical variable with multiple options.
- OnlineSecurity - determines whether the client has subscribed to an online security service. Possible values are "Yes", "No" or "No internet service" and thus it is a nominal categorical variable.
- OnlineBackup - indicates whether the customer has an online backup service. Values are "Yes", "No" or "No internet service". This variable is a nominal categorical variable.
- DeviceProtection - describes whether the customer has subscribed to digital device protection. The values are "Yes", "No" or "No internet service". This is a nominal categorical variable.
- TechSupport - reveals whether the customer is paying for technical support. The values are "Yes", "No" or "No internet service", so it is yet another nominal categorical variable.
- StreamingTV - describes whether the customer has a paid service for streaming TV. Possible values are "Yes", "No" or "No internet service" and can be defined as a nominal categorical variable.
- StreamingMovies – similarly, tells us whether the client has a paid service for streaming movies. The values are "Yes", "No" or "No internet service". This variable is a nominal categorical variable.
- Contract - the type of contract commitment that the customer has agreed to. Values can be "Month-to-month", "One year" and "Two years". This is an ordinal categorical variable.
- PaperlessBilling - determines whether the customer prefers electronic billing. Possible values are "Yes" and "No", so this is a binary categorical variable.
- PaymentMethod - the payment method used by the client to pay invoices. Values can be "Electronic check", "Mailed check", "Bank transfer (automatic)" or "Credit card (automatic)". We can say that this is a nominal categorical variable.
- MonthlyCharges - variable contains information about the customer's monthly subscription fee in monetary units of any amount, so it is a numeric continuous variable.
- TotalCharges - this is another attribute that relates to the subscription amount. In this case, it describes the total amount of money the customer has paid so far. Again, the values in this variable are numeric and continuous.
- Churn - whether a customer left or stayed is defined by either a value of "Yes" when they terminated the contract, or "No" when they stayed.

According to this description, it is clear that the dataset contains different types of variables, seventeen of which are categorical. To go into more detail, there are seven binary variables, nine nominal variables and one ordinal variable. In addition, three numeric attributes are also present, where one is discrete and two are continuous. Finally, there is one alphanumeric variable representing the customer identifier.

4.1 Data preparation and cleanup

After importing all necessary libraries and the file containing data in 'csv' format, we can proceed to the actual analysis process. We will work with data in the Jupyter Lab environment using Python, as described in the methodology section. First, we will check how are the attributes stored and whether there are any faulty or missing observations in the dataset. After retrieving information about the dataset, we see there are exactly 7043 non-zero observations out of the total 7043, which would indicate that all records are correct. However, some variables are listed under the *object* data type, even though we stated above they should represent categorical or numeric attributes, which could cause problems later on. Figure 1 below shows output for the first five attributes.

Figure 1: Verification of the number of observations and data types

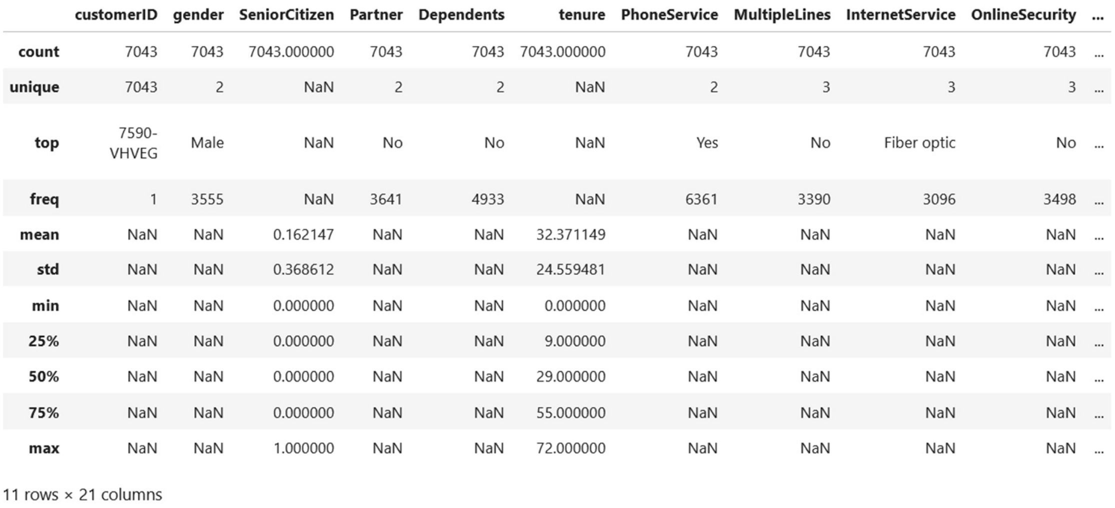


```
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   customerID      7043 non-null   object
 1   gender          7043 non-null   object
 2   SeniorCitizen   7043 non-null   int64
 3   Partner         7043 non-null   object
 4   Dependents      7043 non-null   object
 5   tenure          7043 non-null   int64
```

Source: (Own elaboration in Python)

This was confirmed when we encountered a problem with the *TotalCharges* variable, which is marked as an *object* in the dataset when in reality, it contains decimal numbers representing total costs. When we tried to change the data type of this variable to *float* (a format representing decimal numbers), an error message appeared, indicating that there was a problem, even though the NaN value check did not find any missing values. By further procedure, we found that the most frequently occurring value in this variable was " " (a blank white character), to be more precise, it occurred eleven times. From this we conclude that the missing values were stored as blank characters, giving the illusion that the data was correct. In addition, we also noticed an unusual situation for the *tenure* variable, where the minimum value reaches 0, which turned out to be related to faulty values in *TotalCharges*. After displaying only observations that have a *tenure* value equal to zero, it shows that these are exactly the eleven erroneous observations that have non-existent *TotalCharges*, as seen on Figure 2.

Figure 2: Dataset statistics in Jupyter Lab and faulty NaN observations



	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...
count	7043	7043	7043.000000	7043	7043	7043.000000	7043	7043	7043	7043	...
unique	7043	2	NaN	2	2	NaN	2	3	3	3	...
top	7590-VHVEG	Male	NaN	No	No	NaN	Yes	No	Fiber optic	No	...
freq	1	3555	NaN	3641	4933	NaN	6361	3390	3096	3498	...
mean	NaN	NaN	0.162147	NaN	NaN	32.371149	NaN	NaN	NaN	NaN	...
std	NaN	NaN	0.368612	NaN	NaN	24.559481	NaN	NaN	NaN	NaN	...
min	NaN	NaN	0.000000	NaN	NaN	0.000000	NaN	NaN	NaN	NaN	...
25%	NaN	NaN	0.000000	NaN	NaN	9.000000	NaN	NaN	NaN	NaN	...
50%	NaN	NaN	0.000000	NaN	NaN	29.000000	NaN	NaN	NaN	NaN	...
75%	NaN	NaN	0.000000	NaN	NaN	55.000000	NaN	NaN	NaN	NaN	...
max	NaN	NaN	1.000000	NaN	NaN	72.000000	NaN	NaN	NaN	NaN	...

11 rows × 21 columns

Source: (Own elaboration in Python)

A possible explanation as to why these customers had a contract with duration equal to zero is either a simple mistake or they were indeed brand-new customers who were yet to receive a first invoice. In either case, this information is not very interesting for churn analysis and given the fact that there are only 11 such customers we have decided to remove these observations.

Since all the observations were correct, we were able change the necessary data types. We have converted categorical variables into the *category* data type, which is defined in the pandas library. Numeric variables have been converted to *float* and *int* data types (a format representing integers). Finally, we can proceed to the graphical part of the exploratory analysis.

5 Exploratory analysis

Once it is clear what type of data we will be working with and that all the records are correct, we can take a closer look at the data through exploratory analysis, which tends to be one of the first steps in any data analysis. Exploratory data analysis usually involves statistical and graphical outputs and will help us better understand available data and to build new hypotheses.

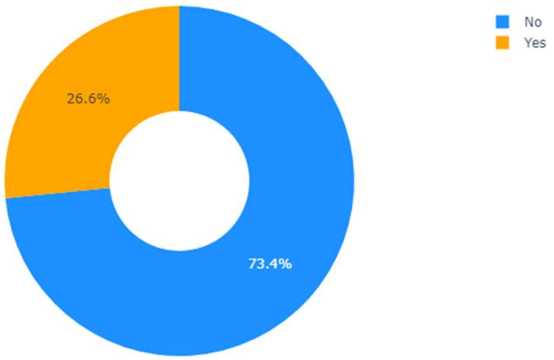
We decided to primarily use the Plotly library to create the graphs. All created graphs are interactive and after clicking any option in the legend, the graph is recalculated. There is a plethora of different data combinations to create graphs, so we will show the most interesting ones, first for categorical and then for numeric variables.

5.1 Categorical variables

5.1.1 Churn ratio

To begin with, we look at the ratio of customers leaving and staying with the telecommunication company. In the Figure 3 below, we can see that the proportion of customers who have left is approximately 26.6%, indicating that this is indeed an unbalanced dataset.

Figure 3: Customer churn ratio

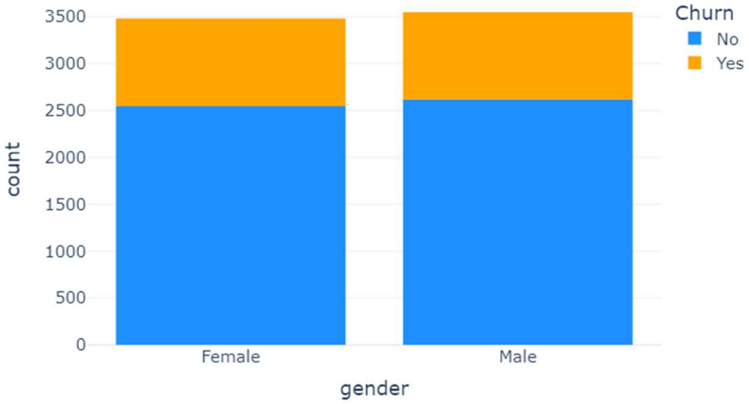


Source: (Own elaboration in Python - Plotly)

5.1.2 Male and female churn rates

The difference in the total number of male and female customers is minimal, and the same is true when comparing their churn rates. The proportion of men who have left is 26.2%, while for women it is 26.96%. These numbers suggest that gender does not have a significant impact on customers' decision to leave an operator.

Figure 4: Comparison of male and female churn rates

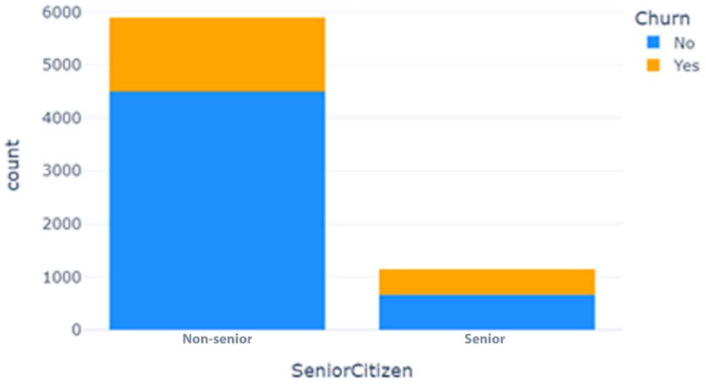


Source: (Own elaboration in Python - Plotly)

5.1.3 Senior citizen churn rate

On the contrary, the group of senior clients showed a significantly higher churn rate (41.68%) compared to the economically active population (23.65%). Despite the fact that seniors make up only about 16% of the total customer base, they account for 20% of all monthly revenues. This trend shows that seniors represent a group with good purchasing power but at the same time a risky group of customers.

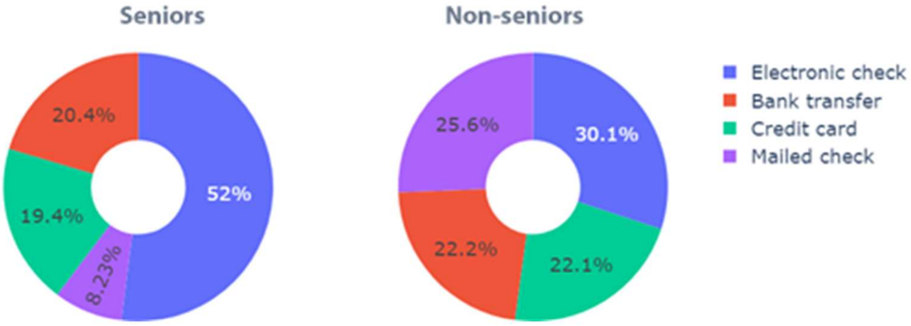
Figure 5: Churn rates for seniors and non-seniors



Source: (Own elaboration in Python - Plotly)

This is followed by pie charts in Figure 6, which compare payment method preferences between senior and non-senior customers. Seniors prefer payment by e-check, used by up to 52% of seniors, while the working population is more divided. Interestingly, for all customers who have left, i.e., not just pensioners, the primary payment method was e-check.

Figure 6: Comparison of payment method

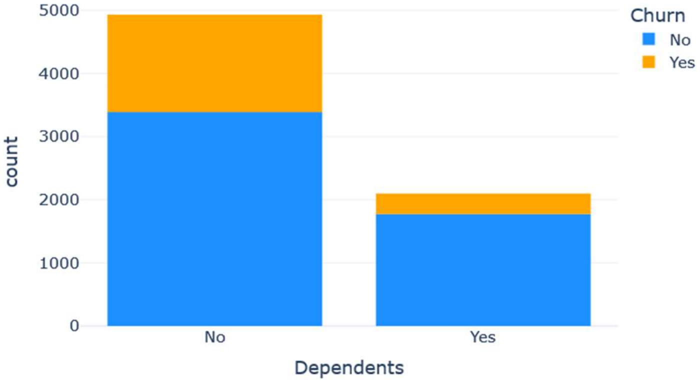


Source: (Own elaboration in Python - Plotly)

5.1.4 Marital status

When it comes to the marital status of clients, variables that describe whether clients have a partner and children can help us to better understand their decision. Clients who do not have children have seen a higher churn rate, which is as high as 31%. This appears to be a sizable difference compared to the churn rate of 15.5% for those with children, so marital status can greatly influence loyalty to a telecommunications service provider.

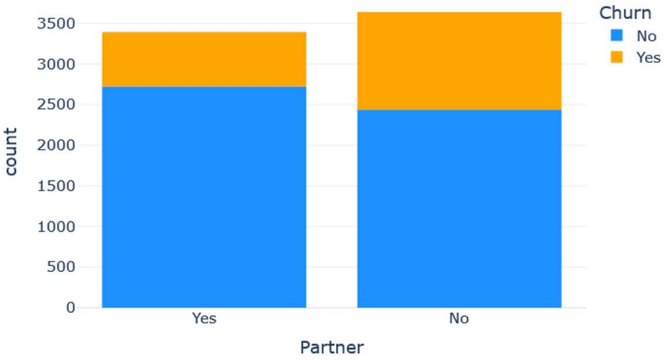
Figure 7: Churn rate and dependents



Source: (Own elaboration in Python - Plotly)

It is comparable for customers who have a partner. The churn rate for customers with a partner is around 19.7%, but for customers without a partner it is nearly 33%. Besides, both groups, unlike those with dependents, are relatively evenly matched in number of clients, with rates of around 3,000 to 3,500 clients.

Figure 8: Churn rate and partners

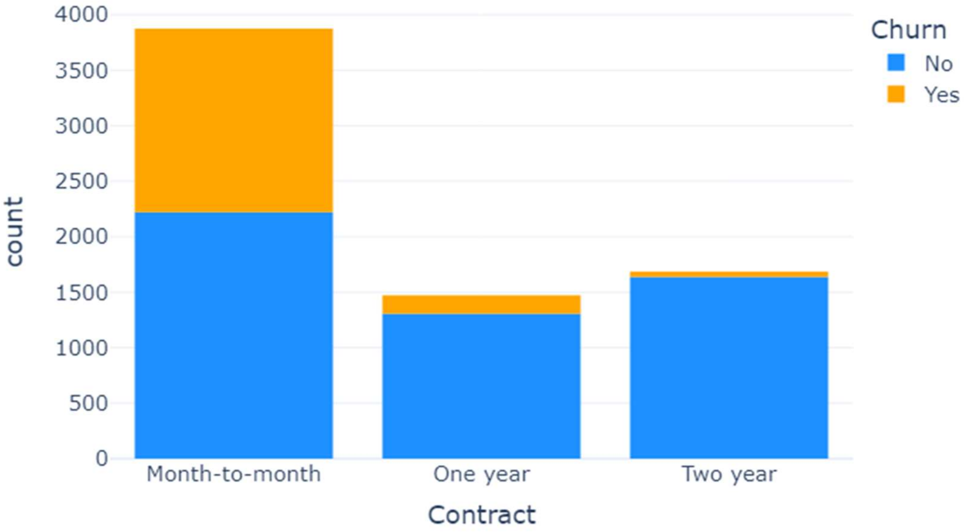


Source: (Own elaboration in Python - Plotly)

5.1.5 Contract commitment

Customers with no contract commitment are not only the largest group of customers, but also have by far the highest churn rate and are therefore considered to be a risk group. In contrast, churn rates for one- and two-year contracts are significantly lower, indicating greater loyalty among customers with longer commitment periods.

Figure 9: Comparison of churn rates by the type of contract



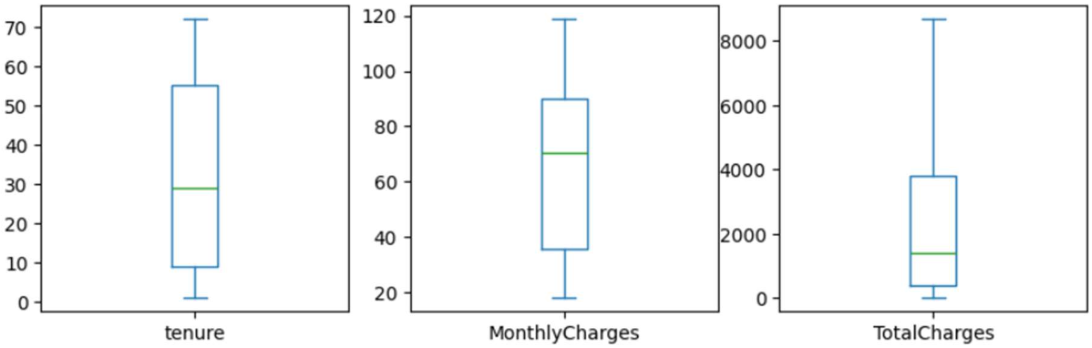
Source: (Own elaboration in Python - Plotly)

5.2 Numeric variables

5.2.1 Checking for outliers

Using boxplot plots, we identified that there are no outlier observations present in the dataset that are further than 1.5 times the interquartile range and that could bias the obtained results.

Figure 10: Boxplot for the numerical variables of the dataset

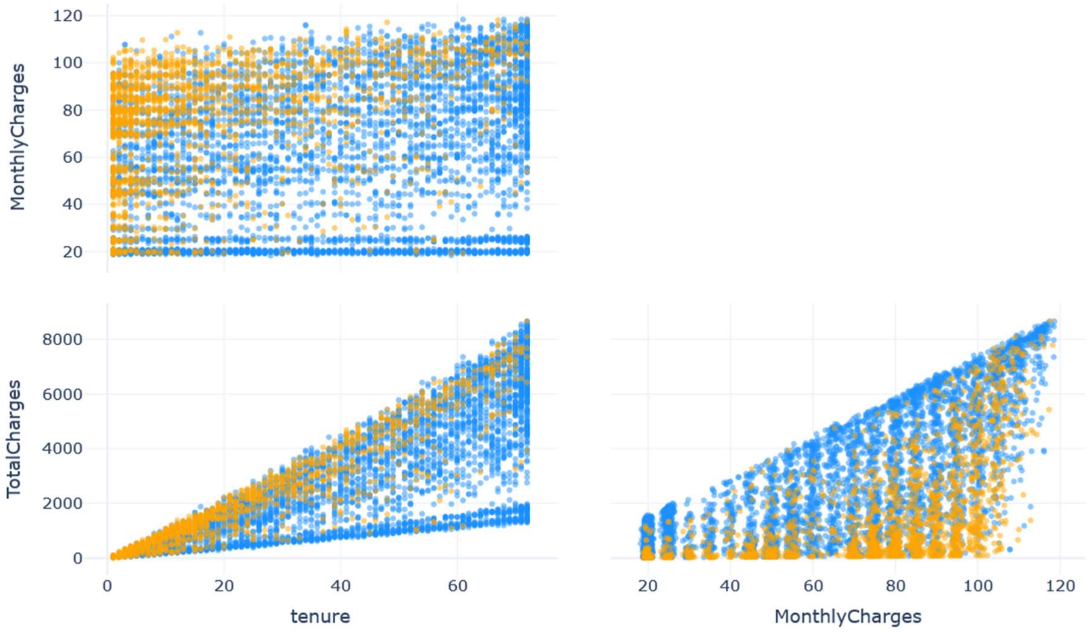


Source: (Own elaboration in Python)

5.2.2 Point charts

Scatter plots of the numerical variables on Figure 11 reveal that there is a relationship between the variables *Tenure*, *MonthlyCharges* and *TotalCharges*. Yellow colored dots represent churned customers, and blue dots represent customers who decided to stay. The relationship between the length of customer loyalty (*tenure*) and the value of monthly payments (*MonthlyCharges*) suggests that customers with shorter tenure and higher monthly payments tend to leave more often. This assumption is reinforced by the dot plot between total costs (*TotalCharges*) and *tenure*, which shows that newer customers (lower *tenure*) who have paid less in total charges are more inclined to leave. At the same time, according to the last plot, which compares monthly costs (*MonthlyCharges)* and total costs (*TotalCharges)*, it seems that those who pay higher monthly fees but have not paid that much in total are more likely to cancel their subscription plan, which would indicate some price sensitivity of customers. These results suggest that for retention, it may be appropriate to focus on customers with higher monthly expenses, especially if they have only been customers for a shorter period of time.

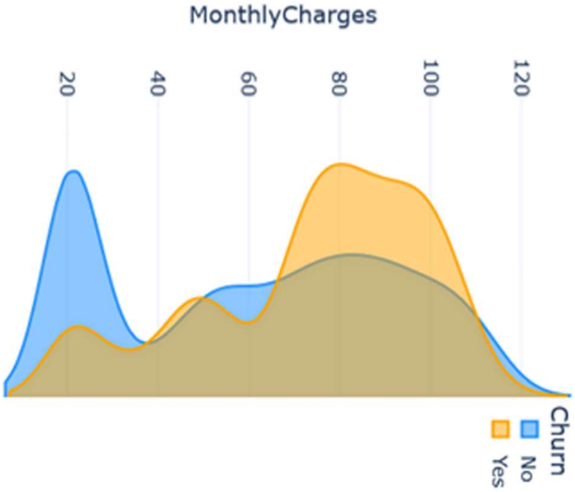
Figure 11: Scatter plots of numerical variables



Source: (Own elaboration in Python - Plotly)

We can also verify our theory about the price sensitivity of customers with a density plot that compares the height of monthly payments separately for customers who decided to leave and those who stayed. The graph below shows that when monthly payments exceed approximately 70 to 80 monetary units, customer churn rises significantly. On the contrary, customer loyalty is higher in the lower price range from 20 to 40 monetary units.

Figure 12: Comparison of churn rates at different monthly payment levels



Source: (Own elaboration in Python - Plotly)

6 Conclusion

In conclusion, utilization of various visualization tools available for Python has proven to be vital for performing exploratory data analysis on churn rates of telecommunication customers. By applying libraries such as pandas, NumPy, Seaborn, and Plotly in a Jupyter Lab environment, we were able to effectively manipulate and visualize complex data relationships, revealing insightful patterns and trends related to customer behavior.

The analysis highlighted the significance of EDA in uncovering factors that contribute to customer churn. By visualizing categorical and numerical variables, we could identify relationships and anomalies that would otherwise remain hidden. The interactive nature of the visualizations allowed us to explore the data dynamically, achieving deeper insights of different customer segments and service preferences.

Therefore, the visualizations created using graphical libraries available for Python have enabled us to uncover important trends in customer behavior, illustrating that demographic factors such as age and marital status or the type of internet service can have a considerable impact on churn rates. Particularly, customers with month-to-month contracts and customers without a partner or children displayed a higher likelihood of leaving the company.

Additionally, our graphical representations, including box plots and scatter plots, not only provided clear insights into the distribution of various customer attributes but also simplified a dynamic exploration of churn patterns, which are crucial for understanding customer retention strategies to mitigate the risk of losing customers.

Finally, this exploratory analysis not only improves our understanding of customer churn but also sets a solid foundation for possible future predictive modeling. By further leveraging the power of Python's visualization and modeling libraries, we can more precisely focus on different customer segments and build strategies to improve customer retention by predicting customer churn with the help of cohort analysis, cluster analysis, machine learning and others.

This work was supported by the Grant Agency of Slovak Republic – VEGA grant no. 1/0047/23 „*The importance of spatial spillover effects in the context of the EU's greener and carbon-free Europe priority*".

References

1. Barba, L. A., Barker, L. J., Blank, D. S., Brown, J., Zingale, M., Willing, C., Wickes, E., West, R. H., Watkins, R. R., Niemeyer, K. E., Lippert, D., Moore, J. K., Mandli, K. T., Heagy, L. J., George, T., & Downey, A. (2019). *Teaching and learning with Jupyter*. Jupyter4edu. https://jupyter4edu.github.io/jupyter-edu-book/
2. Behrens, J. T. (1997). Principles and procedures of exploratory data analysis. *Psychological Methods*, *2*(2), 131–160. https://doi.org/10.1037/1082-989x.2.2.131
3. BlastChar. (2018). *Telco customer churn*. Kaggle. https://www.kaggle.com/datasets/blastchar/telco-customer-churn
4. Graff, M., Moctezuma, D., Miranda-Jiménez, S., & Tellez, E. S. (2022). A Python Library for Exploratory Data Analysis on Twitter data based on tokens and aggregated origin–destination information. *Computers &amp; Geosciences*, *159*, 105012. https://doi.org/10.1016/j.cageo.2021.105012
5. Méo, G. D. (2023). Creating interactive visualizations with plotly. *Programming Historian*, (12). https://doi.org/10.46430/phen0115
6. NumPy Developers. (n.d.). *What is numpy?*. NumPy v2.1 Manual. https://numpy.org/doc/stable/user/whatisnumpy.html
7. The pandas development team. (n.d.). *Package overview*. pandas 2.2.3 documentation. https://pandas.pydata.org/docs/getting_started/overview.html
8. Peng, J., Wu, W., Lockhart, B., Bian, S., Yan, J. N., Xu, L., Chi, Z., Rzeszotarski, J. M., & Wang, J. (2021). DataPrep.EDA: Task-centric Exploratory Data Analysis for statistical modeling in Python. *Proceedings of the 2021 International Conference on Management of Data*, 2271–2280. https://doi.org/10.1145/3448016.3457330
9. Sahoo, K., Samal, A. K., Pramanik, J., & Pani, S. K. (2019). Exploratory data analysis using Python. *International Journal of Innovative Technology and Exploring Engineering*, *8*(12), 4727–4735. https://doi.org/10.35940/ijitee.l3591.1081219
10. Tukey, J. W. (1977). *Exploratory Data Analysis*. Addison-Wesley.
11. Tukey, J. W. (1980). We need both exploratory and confirmatory. *The American Statistician*, *34*(1), 23–25. https://doi.org/10.1080/00031305.1980.10482706
12. Waskom, M. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 2021. https://doi.org/10.21105/joss.03021