

Sales Forecast and Churn Prediction for the International Health Insurance Company

FINAL REPORT

Course: Data Science Continuous Mar23

Jonathan Leipold | Christian Hirning | Rumiya Al-Meri

09.11.2023

[I. Introduction 2](#_Toc1852101975)

[Background 2](#_Toc1977901508)

[Contribution 2](#_Toc1593883786)

[Objectives 3](#_Toc1744347057)

[II. Data exploration and visualisation 3](#_Toc848181416)

[Framework 4](#_Toc392449418)

[II.1. Sales prediction 4](#_Toc928019284)

[Relevance 5](#_Toc1681293684)

[Pre-processing and feature engineering 5](#_Toc176126365)

[Visualizations and Statistics 7](#_Toc318558476)

[Modelling Sales Prediction with Time Series Models 9](#_Toc1119202881)

[Model choice and optimization 10](#_Toc1209561989)

[Modelling Sales Prediction with Classification Models 10](#_Toc424786099)

[II.2. churn prediction 11](#_Toc1246720016)

[II.2.1 Data collection & Description 12](#_Toc1300707530)

[II.2.2 Pre-processing and feature engineering 13](#_Toc1544690077)

[Target variable 13](#_Toc2073883734)

[Handling Missing Values 15](#_Toc817881025)

[Outliers 16](#_Toc94627923)

[II.2.3 Visualization & Dependencies 17](#_Toc1215845523)

[Correlations between features: 18](#_Toc1708322208)

[Correlations with target variable: 18](#_Toc1124397716)

[Distributions 19](#_Toc241120553)

[II.2.4 Encoding & Normalizing 20](#_Toc1020657405)

[Encoding 20](#_Toc466176208)

[Scaling 21](#_Toc1010412566)

[II.2.5 Final preprocessing parameters 21](#_Toc1168398853)

[III. Modelling 22](#_Toc1615753133)

[III.2 Churn Prediction modelling 23](#_Toc780445007)

[III.2.1 Classification of the problem 23](#_Toc1337747173)

[Model choice and optimization 23](#_Toc698514707)

[Interpretation of results 24](#_Toc1280459134)

[IV. Conclusions 24](#_Toc2028506451)

[challenges 25](#_Toc1739943598)

[Report 26](#_Toc68488215)

[further steps 26](#_Toc2043341338)

[Bibliography 26](#_Toc1014307711)

[Appendices 27](#_Toc81469519)

# Introduction

About the company and the product

[BDAE](https://www.bdae.com/en/bdae-group/about-bdae-group) is a specialist for international health insurance products. It is an insurance broker in co-working with the big health insurance companies. BDAE has its’ own sales & product development as well as claims handling service. Main target group are expatriates with the longer international assignments (from at least 1 year).

The company offers own health insurance products. There are several products depending on multiple factors like the insurance time period, home country and the area where the insured country of stay is situated. For this project it was initially decided on considering only one product (insurance type) to reduce the complexity. The product price is a premium amount which is paid by a client on a monthly, quarterly or a yearly basis.

## Background

The sales forecast is inevitable for the company. Based on it BDAE takes decisions on new products development and clients’ acquisition to ensure a continuous growth and avoid existential threat. Quick access to forecast by product allows quick decision-making and resources optimization which in turn reduce costs and increase overall profit. Churn predictions are not made yet and would be a novelty in the company’s data analysis. In general, there is no implementation of python or ML models yet in the business.

All data is collected in the ERP-system (SAP) based on a SQL-database. The forecast of future sales is still partly based on manual estimations. The sums and ratios of historical sales data are used combined with manual estimations of summed sales amounts to predict future sales. Data is grouped by (insurance-)product / category, time and some other variables in Power Pivot and Excel. Churn predictions are not yet implemented into the forecast procedure by the company. The goal of this project from a technical point of view is to support the manual estimations and decisions with the reliable ML-models.

From a scientific point of view there are several challenges to overcome by finding and purifying useful data as well as choosing the best ML-model for sales forecast. It is described later in details in the chapter IV. Since there is no information about future behaviour such as the number of contracts is unknown as well as the profile of future clients, it is challenging to find the right and reliable ML-model to predict future sales.

## Contribution

In this project group Johnathan Leipold is a representative of the BDAE company, an industry expert, the data owner and the initiator of a current project. He has a mathematical background and little programming and Deep Learning knowledge from a student job and private courses but no experience in ML projects. Johnathan was consulting on the data during the project, set up the initial data base and mostly all pre-processing steps as well as new feature collection from the ERP-System and feature engineering in SQL for churn prediction. Christian Hirning and Rumiya Al-Meri have no experience in the insurance industry but rather in statistics and Christian as well a deeper knowledge in programming. Raphael Kassel (DataScientest) contributed as a project tutor.

Jonathan regularly consulted with a company IT expert with basic statistics and ML skills about useful modelling options, and with another colleague about the quality and meaning of the feature content. No data scientists or ML experts were involved.

## Objectives

The **main objective** was to create **the best performing model for sales predictions**, in particularly prediction of premium amounts per month. Due to the big variety of product characteristics, only transactions concerning one main product type were considered to build a prototype.

Initially 2 main goals were defined:

1. Find the best model for forecasting / predicting the premium amount
2. Find out how premium adjustments impact the value of premium amount

During the project, the project group faced the problem of a limited number of features which are known for the future. Therefore, it was decided on project extension with the further objective, namely **churn predictions**. The contracts’ data for all products was taken and enriched by additional, information from the ERP-System.

Withing this sub-project another 2 goals were defined:

1. Identify main features that have an impact on customers’ termination behaviour
2. Find active contracts that are more likely to get terminated by the customer

# Data exploration and visualisation

## Framework

The data is owned by BDAE Group and not available to the public. Each group member signed the confidentiality agreement with BDAE. Personal information like name and address were not collected, others like ContractID and ProductName were replaced with pseudonymised values in SQL.

All data comes from the ERP-System SAP Business One. It was collected and joined on a Microsoft SQL Server as views. Some features were created by calculations in SQL.

In the end 4 views were exported as csv files for import and modelling:

1. SalesData-Example.csv  
   Sales Data in form of transactions, in total about 230 000 transactions. Each transaction belongs to a specific contract and time period and represents incoming and outgoing cash flow. This data only includes transactions belonging to one main product and the period of 2014-2023 YTD. The Sales Data dataset includes as well 36 variables (e.g., birthday, policy StartDate, premium Amount, Contract Id and Fee Rate, Zone Desc).

1. premium\_adjustments\_example.csv  
   This data contains information about adjustments of premium amounts of the main product, in total 58 lines. Each line belongs to a specific adjustment date, ZoneModel and product group. It is merged with the sales data later via time and product code.

For the second sub-project 2 more datasets were created:

1. BDAE\_DataMining\_Policies.csv  
     
   This data contains information belonging to one specific contract. Some data comes directly from the contract information, other variables were calculated as sums, ratios etc. of the transaction lines used above as SalesData. It includes all contracts created after 01.01.2017. In total about 20 000 lines. The file was updated with more recent data as well as corrections and additional features from SQL in another file BDAE\_DataMining\_Policies\_v2.csv
2. BDAE\_DataMining\_Products.csv

Describes special characteristics of products like category, max. duration, etc. They can be merged with the contracts’ information via the unique product code. In total about 300 lines.

The project is splited in two sub-projects: sales (premium amount) prediction and churn prediction and will be presented in this part by sub-project for a better understandability.

## II.1. Sales prediction

## Relevance

The development of the premium amount over the years 2014 - 2023 is most relevant for the sales prediction. Therefore, the sum of premium amounts per month was taken as a target variable. Monthly grouping was chosen from the business perspective as BDAE forecast is done by month.

During the project it turned out that the zone model has changed in 2018. Germany as a separate zone was now included into one of the new zones (see Fig.1). The replacement of old zone model with the new one explains the drop in 2019

Ein Bild, das Text, Diagramm, Reihe, Schrift enthält.

Automatisch generierte Beschreibung

Figure 1: Zone models by sum of premium amounts and number of unique contract IDs

Because to this change, the old zone model and the new zone model were considered and analysed separately. This presented a greater challenge. After analysing the data, no relevant features could be generated to predict the future development of the premium amount. This limited our ability to select additional features for modelling at a later stage. Despite the limitations, we analysed the premium amount using time series models (see chapter xy) and classification models (see chapter xy).

## Pre-processing and feature engineering

After initial exploratory analysis of the data, missing values were deleted or replaced, and the variables were converted to the correct data type.

Looking at the premium amounts over time by month, there is a dip in 2019 (see Fig.2). Ein Bild, das Text, Diagramm, Reihe, Zahl enthält.

Automatisch generierte Beschreibung

Figure 22: Sum of premium amounts and number of unique contract IDs over time by month

As mentioned earlier, a change in the zone model was identified. A new variable Zone\_United was created to include this zone conversion (see Figure 3). Moreover, optional parameters for ZoneModel are included in the preprocessing function.

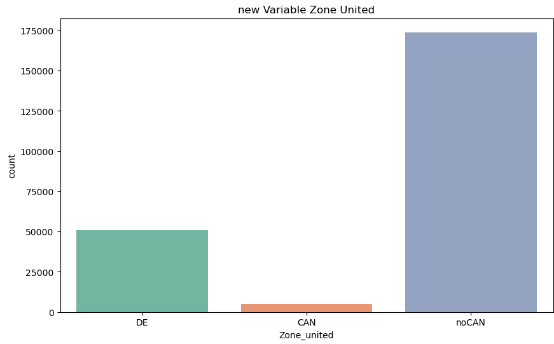
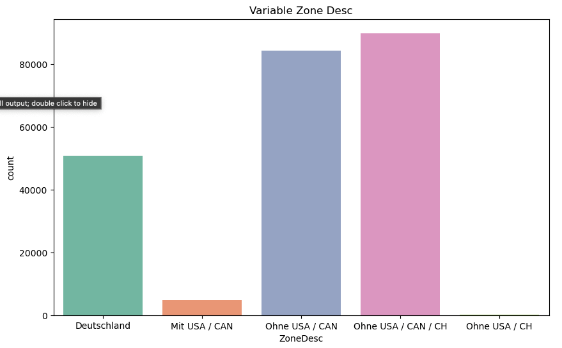


Figure 33: Count of unique Contract IDs by Zone: New Zone model vs. Old Zone model

This adjustment allowed to identify two different trajectories for the two zone models (see Figure 4). Based on this finding, the further procedure was adjusted.

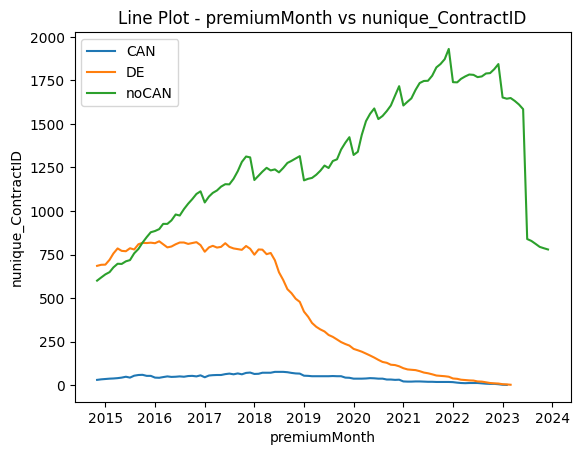


Figure 4: Count of unique Contract IDs by Zone: New Zone model vs. Old Zone model

Dummies were created for the categorical variables and other variables such as the number of days since the last adjustment. It turned out that these variables are not suitable for the Time Series models.

For a classification model some features were created for a modelling. Due to a limited number of variables which can be grouped by month to fit to the target variable, the following three features were created: mean of the Age at premium and mean of the policy age in months. Later, the rolling mean of the last 12 months for a sum of premium amount was added. From the business perspective, the first two features do not make much sense. The rolling mean as will be shown later has the maximum relevance in this case.

## Visualizations and Statistics

* Have you identified relationships between different variables? Between explanatory variables? and between your explanatory variables and the target(s)?
* Describe the distribution of these data, distribution, outliers.. (pre/post processing if necessary)
* Present the statistical analyzes used to confirm the information present on the graphs.
* Draw conclusions from the elements noted above allowing them to project themselves into the modeling part

### Correlations between features:

A strong correlation was found between the number of contracts and the premium amount (see Figure 1). Also, a positive correlation of the premium amount and the last adjustment of the premium amount.

The distribution of the variables used for a feature engineering is shown below.

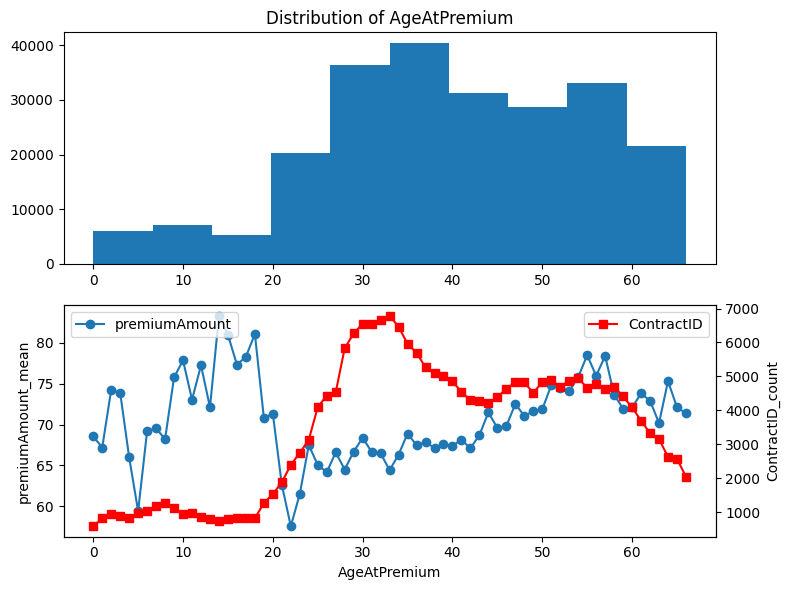


Figure 5: Count of unique Contract IDs by Zone: New Zone model vs. Old Zone model

Most clients are between 25 and 58 years old. No obvious correlation between Age at premium and premium amount but still it can be seen that the premium amount increases with the age after 25 years.

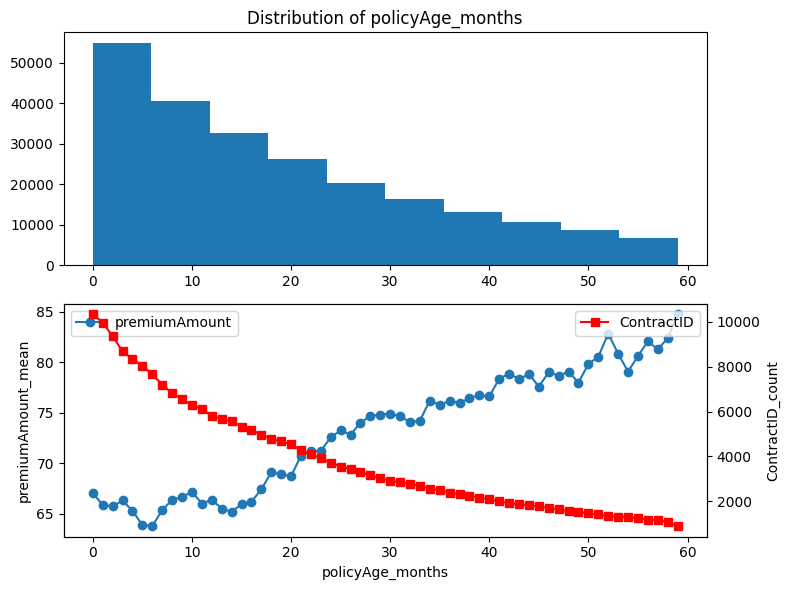


Figure 6: Count of unique Contract IDs by Zone: New Zone model vs. Old Zone model

The maximum contract duration is 5 years. The number of Contract ID is decreasing with the policy age increase. The premium amount, on the contrary, is growing which is obvious.

Both presented variables are taken as features for a classification model. Correlation between the created features and the target variable is shown below (see Figure 7).

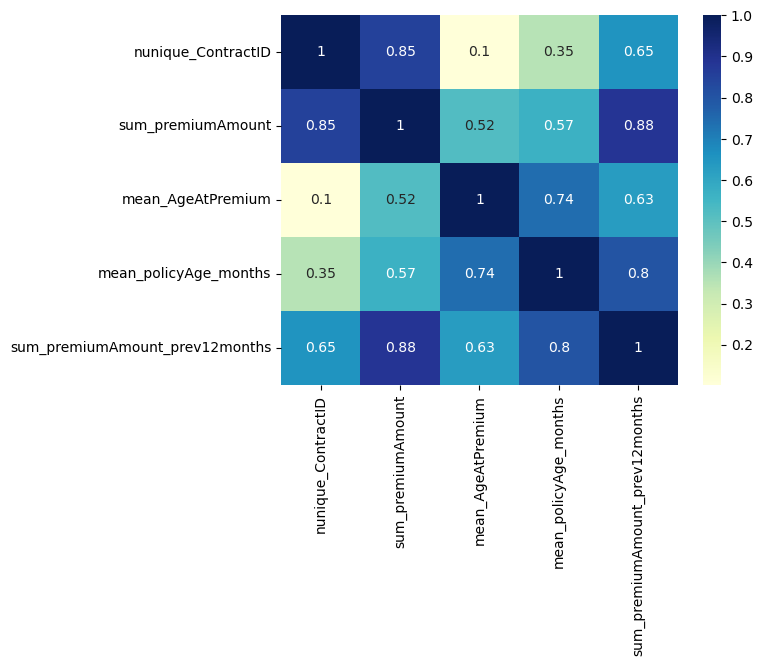


Figure 7: Correlation between policy Age, Age at premium and lag premium amount over the last twelwe months.

Obviously premium amount is mostly correlated with the rolling premium amount over the last twelwe months and the number of unique contract IDs. As was

It was decided to model the prediction of the premium amount in two different ways. On the one hand by classical time series models like SARIMA, regression models and ML models (Random Forest Regressor, XGBoost). The Jupyter Notebook by Rob Mulla https://www.kaggle.com/code/robikscube/tutorial-time-series-forecasting-with-xgboost was used as a basis and adapted. On the other hand, the premium amount was divided into classes and predicted with classification models.

Finally, we had a dataset grouped by month with lag features over the last twelve months. In addition, temporal factors such as the year, quarter were also included. Later in this report in more details by model.

## II.2. Churn prediction

## II.2.1 Data collection & Description

Data was collected directly from the contracts in SAP. This includes features like nationality, age, start- & end dates, as well as the termination date, which is filled, if the customer decided to end the contract:Ein Bild, das Text, Screenshot, Software, Webseite enthält.

Automatisch generierte Beschreibung

Figure 4: Target & Feature variables in the ERP System

This information is used for the **target variable** and set to 1 if it is filled. It’s important to know, that this is not always a sign of dissatisfaction but can be caused by several reasons like the return to home country for example. That’s why we will later also look at termination reasons (Beendigungsgrund) to select only contracts who got terminated due to specific reasons as an **alternative target variable**.

To add more information about past customer behaviour information from the transaction lines where added. They contain information about paid premiums as well es claimed customer invoices and are grouped and joined by their contractID in SQL:

Ein Bild, das Text, Software, Webseite, Computersymbol enthält.

Automatisch generierte Beschreibung

Figure 5: Claim & Premium variables in the ERP System

↓

Ein Bild, das Text, Screenshot, Schrift, Software enthält.

Automatisch generierte Beschreibung

Figure 6: Feature Engineering of premium- & claims-data

↓

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Figure 7: Example of premium- & claims-data in Jupyter Notebook

More information about the collection and preprocessing in SQL can be found in the sql-files in github.

I expected especially the claims related columns like mean\_payoutDays or the ratio of claimed and paid out invoices to be a major feature. But unfortunately, these data include **a lot of missing values**. One reason that caused a massive preprocessing work. See section below.

## II.2.2 Pre-processing and feature engineering

### Target variable

The target variable got defined as 1, if the contract contains a terminationDate, which means that the customer whished to end the contract before an eventual endDate.

If you look only at ended contracts the ratio of terminated contracts is extremely high:

Ein Bild, das Text, Kreis, Screenshot, Diagramm enthält.

Automatisch generierte Beschreibung

Figure 8: Target distribution

After internal feedback it got clear that this can be related to several reasons and that a terminated contracts isn’t always a result of disappointed customers.

That’s why an alternative target variable got created after looking at the specific termination reasons and their distribution:

Ein Bild, das Text, Screenshot, Farbigkeit, Schrift enthält.

Automatisch generierte Beschreibung

Figure 9: Termination reasons and their distribution

The variable ‘ds\_terminated’ gets created within the preprocessing function and is set to 1 only for specific termination reasons. It can be chosen within the train-test-split function as alternative target.

To choose this variable as target results – depending on the chosen reasons – mostly results in highly imbalanced data as it is shown below for the default values:

Ein Bild, das Kreis, Diagramm, Screenshot, Text enthält.

Automatisch generierte Beschreibung

Figure 10: distribution of alternative target variable

### Handling Missing Values

Especially variables containing information about last year did contain a lot of missing values:

Ein Bild, das Text, Schrift, Reihe, Screenshot enthält.

Automatisch generierte Beschreibung

Figure 11: Top NULL-Columns

Which makes sense, since contracts who got terminated more than one year ago won’t contain data here. That’s why more features were created in SQL to make a connection between active and ended contracts:

1. A reference date was created to compare the last active year of active and ended contracts.
2. Sums and averages over the last active year were created.

LastYear-columns were dropped in the preprocessing steps in python and replaced by lastActivYear columns.

In the preprocessing function other NULL values were depending of their content and dtype either

* Replaced by mean (e.g. mean\_payoutDays)
* Replaced by 0 (e.g. sum\_payout)
* Certain string values to avoid errors (e.g. ‘XX’ for countries and ‘None’ for terminationReasons)
* Column dropped (e.g. product\_code)

In some cases, company internal feedback was needed to clarify, if the amount of missing values make sense and how to handle them meaningful.

### Outliers

Due to the large number of features the columns got split by their datatypes for outlier detection. If outliers where detected the preprocessing function got adjusted accordingly.

Depending on the columns content:

1. The column got dropped, because it is redundant due to new features (e.g. lastYear-columns after adding lastActivYear-columns)
2. The column got replaced by another column to avoid high correlations (e.g. sum\_payout\_lastActivYear by payout\_ratio\_lastActivYear)
3. Outliers were dropped (e.g. sum\_claimed-columns)

This way the top 10 numerous outlier columns got preprocessed like this:

Ein Bild, das Text, Zahl, Schrift, parallel enthält.

Automatisch generierte Beschreibung

Figure 12: Top 10 initial outlier columns

Ein Bild, das Text, Zahl, parallel, Schrift enthält.

Automatisch generierte Beschreibung

Figure 13: Top 10 initial outlier columns after preprocessing

## II.2.3 Visualization & Dependencies

### Correlations between features:

High correlations was detected especially between:

1. Dates around policy\_startDate
2. Product specific characteristics
3. Claimed & payout sums

Ein Bild, das Text, Screenshot, Diagramm, parallel enthält.

Automatisch generierte Beschreibung

Figure 14: Top feature correlations

This was put into the preprocessing function in the way that:

1. Apply- & SignDate were dropped. Instead, date difference between Apply- & startDate gets considered.
2. Product columns can be selected to be dropped if product information get merged.
3. Redundant claims columns were dropped: e.g., ‘sum\_retained’ (= sum\_claimed – payout\_sum) &

Instead of absolute values, ratios of payout to claimed sum gets calculated to avoid high correlation of payout- & claimed amount:

This way the amount of highly correlated columns could be reduced.

### Correlations with target variable:

To compare all features with the target variable some modifications on the df needed to be done:

* Encode string/ categorical columns
* Convert datetime columns to int by keeping their year only

In the end the main correlating features are those wo directly give information about the current state of the contract (active / ended):

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Figure 15: Top correlations with target variable

Additionally, correlations of the initial and preprocessed df were compared. More variations of correlation comparison can be found in the notebook.

### Distributions

The high correlations with the target variable can also be visualized when looking at the distribution of these values with hue=’terminated’:

Ein Bild, das Diagramm, Reihe, Screenshot enthält.

Automatisch generierte Beschreibung

Figure 16: effEndDate distribution

There are a lot of contracts with endDate in 2100 - a internal date for “infinit” - which have a lot of products. To give the option to reduce these “outliers” 2 more parameters got included into the preprocessing function to optionally cut the effEndDatea to a certain date.

This way you can have a closer look on the distribution of the main columns:

Ein Bild, das Diagramm, Reihe, Screenshot, Design enthält.

Automatisch generierte Beschreibung

Figure 17: effEndDate distribution after cutting max to 31-12-2023 (default)

It can be seen clearly that the effEndDate has high influence on the target variable, since (almost) all terminated contracts have an effEndDate in the past. This correlation gets much lower when choosing the alternative target variable (see target distribution).

Another interesting distribution is the MainProductCode:

Ein Bild, das Diagramm, Screenshot enthält.

Automatisch generierte Beschreibung

Figure 18: MainProduct distribution

While the most frequent MainProducts have a balanced distribution of the target variable, there are common MainProducts with a positive (G014) or negative (G004) impact on the termination status.

## II.2.4 Encoding & Normalizing

### Encoding

To create numerical data for the use of ML Models the data can be encoded with the help of predefined functions.

2 Encoder were chosen:

1. CountFrequency:

This encoder was chosen to reduce the feature amount. To avoid data leakage this encoder will take place in the process / after train-test-split. To re-transform the encoded data an encoder will be returned by the create\_train\_test function.

Disadvantage of this encoder is, that it can give false impression because nominal data will be interpreted as ordinal data. That’s way another semi-manual encoding was created:

1. Dummy Encoding

This Encoder can also be used before the creation of train and test data and avoids data leakage and keeps nominal data. [[1]](#footnote-2)

Disadvantage is the creation of a large amount of features. To reduce this some additional things were done:

* Values of categorical columns with only little amount of rows get dropped (e.g. MainProducts with less than 5 contracts).
* Countries and Nationalities are grouped by their regions with the help of a REST Countries API inside of the country\_to\_region\_mapping function.

### Scaling

Due to big differences in the features distributions the data gets adjusted.

As scalaer **MinMaxScaler** was chosen as an easy to use and understandable Scaling function that keeps the ratios and should work well in almost all cases. Since the distributions above did show enough examples without Gaussian distribution (e.g. policy\_effEndDate, MainProductCode) the use of the Standardization is waived.[[2]](#footnote-3)

## II.2.5 Final preprocessing parameters

Finally, some preprocessing parameters got defined. Depending on the values of these parameters the predefined functions are opted in/out and filled with input parameters:

* year\_only: bool, default=False
  + Set to True to keep only year of all datetime features and convert them to int. Otherwise year and month will be separated and kept.
* Drop\_cols: list of strings, default = []
  + Inserted strings will be tried to be dropped. If col name doesn’t exist or already got dropped within the regular preprocessing process a corresponding message will be printed.
* claim\_ratios: bool, default=True
  + If set to True, some claims related columns will be dropped or replaced and cleaned to minimize correlating columns. In specific: retained columns get dropped, payout amount columns cleaned and replaced by a ratio of claimed amount.
* cut\_effEnd: bool, default=False
  + Set to True to cut policy\_effEndDate values at a specific cut\_date to tighten distribution.
* cut\_date: datetime, default = ‘2030-12-31’
  + Optional, if cut\_effEnd == True. All policy\_effEndDate values > cut\_date will be replaced by this value.
* add\_products: bool
  + Set to True to merge product characteristics to the contracts df based on the product\_code.
* product\_drop\_cols: list of strings, default = []
  + Optional, if add\_products == True, product columns can be selected to be dropped before merge.
* save\_csv: bool
  + Set to True to save the preprocessed df as csv-file in the 'preprocessed’ subfolder.
* filename: string, default = ‘contracts\_preprocessed'
  + Optional, if save\_csv == True. Filename can be set to name it by chosen parameter options.

This parameters will be given as preprocessing options in streamlit to create different transformations of the initial df and use them for modelling.

# Modelling

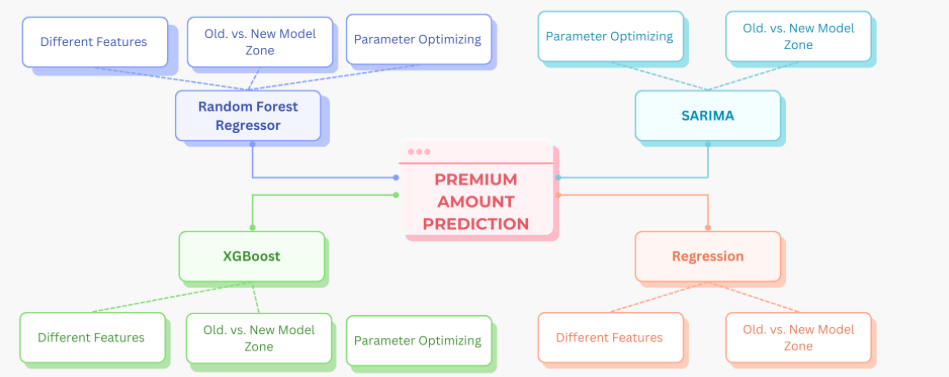
# III.1 Sales prediction with Time Series modelling

# 

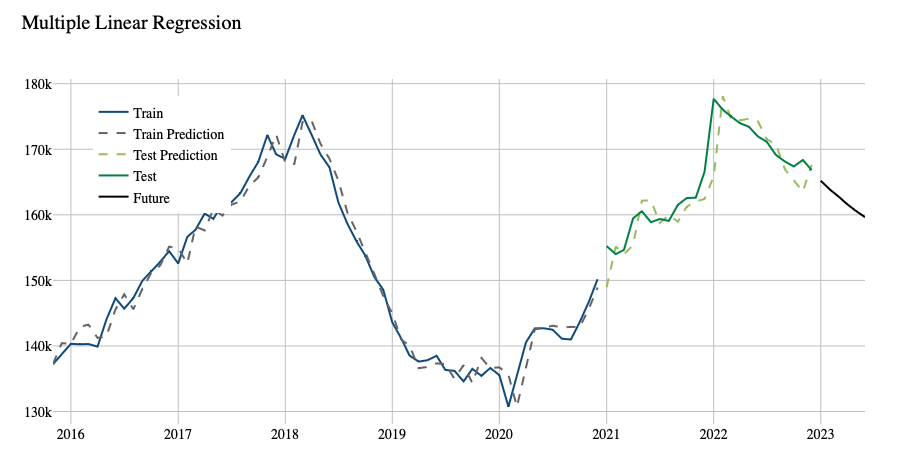
Four different models were calculated for the two zone models. The R^2 and RSME were used as metrics. The models were optimized using Time Series Split and Grid Search CV. Furthermore, a different number of lags was tested.

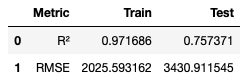
The Forecast presented a special challenge. A recursive approach was chosen. The premium amount of the following month is estimated and transferred back to the fitted model.

Advanced Models were not used.



Interpretation of the results is difficult. The two zone models produced very different results. The Random Forest Regressor and XGBoost models were clearly better for the strongly varying courses in the old zone model. For the linear course of the new zone model "noCAN" the regression models were better. Time series splits also produced strongly varying results, which further complicate the interpretation.





* What kind of machine learning problem is your project like? (classification, regression, clustering, etc)
* What task does your project relate to? (fraud detection, facial recognition, sentiment analysis, etc)?
* What is the main performance metric used to compare your models? Why this one?
* Did you use other qualitative or quantitative performance metrics? If yes, detail it.

## Model choice and optimization

* What algorithms have you tried?
* Describe which one(s) you selected and why?
* Did you use parameter optimization techniques such as Grid Search and Cross Validation?
* Have you tested advanced models? Bagging, Boosting, Deep Learning… Why?

# III.2 Sales prediction with Classification modelling

# III.2 Churn prediction modelling

## III.2.1 Classification of the problem

* What kind of machine learning problem is your project like? (classification, regression, clustering, etc)
* What task does your project relate to? (fraud detection, facial recognition, sentiment analysis, etc)?
* What is the main performance metric used to compare your models? Why this one?
* Did you use other qualitative or quantitative performance metrics? If yes, detail it.

Churn prediction is a binary classification problem. Supervised learning methods where chose to make predictions. The main goal initially was to predict (the probability) if a currently active contract will be terminated by the customer soon. From a business perspective this would mean to recognize contracts with a risk to be terminated in order to be able to counteract this in time and minimise the termination rates.

Over time, it became apparent that it was difficult to define such a target variable with a temporal component for the historical data. Instead, it was decided to use the existing variable to predict whether a contract got terminated by the client or not, based on the features. Combined with highlighting main global and individual features leading to a positive prediction with the help of Shap, this can still create insights in detecting active contracts with a higher risk to be terminated.

The Outcomes of Modelling can be visualized by a Confusion Matrix:

|  |  |  |  |
| --- | --- | --- | --- |
|  | | **PREDICTED** | |
| **Contract Terminated (1)** | **Contract Non-terminated (0)** |
| **TRUE** |  | TRUE POSITIVE (TP) | FALSE NEGATIVE (FN) |
| **Contract Terminated (1)** | Model correctly predicts that the contract got terminated by the customer. | Model predicts that the contract is still active or ended naturally but it got terminated. |
|  | FALSE POSITIVE (FP) | TRUE NEGATIVE (TN) |
| **Contract Non-terminated (0)** | Model predicts that the contract got terminated but it’s still active or ended naturally. | Model correctly predicts that the contract is still active or ended naturally. |

Table 1: Confusion Matrix description

A FP Error means that the model falsely predicts a contract to be terminated by the customer. This would result in a “false warning”. On the other hand a FN Error would result in missing a terminated contract and therefore a missed chance to counteract the termination. So, in the first place it’s chosen to **minimize the FN Rate** (Type 2 error) to maximize the number of detected terminations.

So, while a good Precision score within the positive class (1) is important to not waste capacity on controlling too many false alarms in practice, Recall of the positive class should be ranked even slightly higher to avoid undiscovered terminations.

To keep track of both – precision and recall - the main metric chosen is the **f1-score**. Alternative metric could be f2-score to put the emphasis more onto recall.[[3]](#footnote-4)

## II.2.2 Model choice and optimization

Initially DecisionTree was chosen as first option to start with a easily interpretable model. Discovering Shap as Interpreter it got replaced by **XGBClassifier** algorithm to maximize the performance.

Without setting any parameters it resulted from the beginning in solid results on the ‘terminated’ target.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Figure 19: First results of xgboost on 'terminated' and 'ds\_terminated'

For the alternative target variable first results were terrible. F1-Score on the test data was 0. Adjustments of the preprocessing parameters only resulted in unsignificant changes. Main reason must be the high imbalance of data between class 0 and 1.

It was tried to handle the imbalance using RandomOverSampler() with GridSearch inside a pipeline with the XGBClassifier as well as a GridSearch on the ‘scale\_pos\_weight’ parameter of the XGBClassifier but both didn’t result in a higher F1 score.

A self-written function instead got created to rebalanced the df by a input factor:Ein Bild, das Text, Screenshot, Reihe, Zahl enthält.

Automatisch generierte Beschreibung

Figure 20: Resampling ds\_terminated

Looping over different resampling factors resulted in an increase of F1 from 0(!) to at least 0.34 on the test set:

Ein Bild, das Text, Reihe, Diagramm, Screenshot enthält.

Automatisch generierte Beschreibung

Figure 21: Finding best Resampling ratio to increase F1

Using resampled data for parameter optimaization with GridSearch and CV resulted in further improvements up to a F1 Score of 0.5 on the test data.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Figure 22: Best XGBClassifier on resampled data

## Model choice and optimization

* What algorithms have you tried?
* Describe which one(s) you selected and why?
* Did you use parameter optimization techniques such as Grid Search and Cross Validation?
* Have you tested advanced models? Bagging, Boosting, Deep Learning… Why?

## Interpretation of results

* Have you analyzed the errors in your model?
* Did this contribute to his improvement? If yes, describe.
* Have you used interpretability techniques such as SHAP, LIME, Skater… (Grad-CAM for Deep Learning…)
* What has (or not) generated a significant improvement in your performance?

**Assessment methods:**

**Professional scenario: based on a proposed solution, the candidate will have to produce a summary report including: the explanation of the choices of AI solutions implemented, the interpretation of the results, the evaluation of the reliability of the algorithms and an optimization proposal.**

# Conclusions

## challenges

* What was the main scientific obstacle encountered during this project?
* For each of the following points, if you encountered difficulties, detail how they slowed you down in setting up your project.
* Forecast: tasks that took longer than expected, etc.
* Datasets: acquisition, volumetry, processing, aggregation, etc.
* Technical/theoretical skills: timing of skill acquisition, skill not offered in training, etc.
* One of the difficulties
* Relevance: of the approach, model, data, etc.
* IT: storage power, computational power, etc.
* Other

The main challenge was the fact that the project group worked with real data from a system with which real people work (& make mistakes from time to time). As well the ERP System was changed through the period when the data was collected, which made some older data less trustworthy. Therefore, the validity of the data could never be 100% relied upon. Jonathan, in particular, therefore spent most of his time explaining and preparing the data in both SQL and Python. This time was particularly lacking later in the modelling part.

In addition a lack of knowledge about possibilities of ML Modelling at the beginning of the project lead to unspecific objectives where it has become clear that they cannot be reached for the most part with the existing data. At the sales project it was a lack of known feature information from the future. For the churn part only a target value in the sense of “currently terminated” could be predicted instead of “will be terminated”.

Another difficulty faced by the project group was the lack of skills related to working in a remote group on a data science project. The group underestimated the time needed to divide the tasks, especially in the first part of a project: data exploration and visualisation, and had no theoretical background on how task delegation works best in such projects. It is highly recommended to add the data science project management course as a mandatory module to provide the in-depth skills in this area. For the structure, the group personally would have been helped by a deeper familiarisation with the project and an assessment of the objectives on the part of DataScientest, as well as an actual control of the progress and a demand for the interim reports at the required time to keep the project group on track.

## Report

* Detail what was your main contribution to achieving the project's goals.
* Have you changed the model since the last iteration? If yes, provide details.
* Present the results obtained and compare them to the benchmark
* For each of the project's goals, detail how they were achieved or not.
* If they have been reached, in which process(es) can your model fit? Detail.

## further steps

* What avenues for improvement do you suggest to increase the performance of your model?
* How has your project contributed to an increase in scientific knowledge?

# Bibliography

* What bibliographical elements (research articles, blog, books, etc.) did you rely on to carry out your project?
* <https://www.mckinsey.com/capabilities/operations/our-insights/ai-driven-operations-forecasting-in-data-light-environments>
* <https://thecleverprogrammer.com/2021/05/19/sales-prediction-with-machine-learning/>
* <https://towardsdatascience.com/5-machine-learning-techniques-for-sales-forecasting-598e4984b109>
* <https://medium.com/aiskunks/categorical-data-encoding-techniques-d6296697a40f#:~:text=It%20refers%20to%20the%20process,with%20text%20or%20categorical%20variables>
* <https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6>
* <https://dev.to/balapriya/cross-validation-and-hyperparameter-search-in-scikit-learn-a-complete-guide-5ed8>
* <https://www.simplilearn.com/normalization-vs-standardization-article>
* <https://neptune.ai/blog/evaluation-metrics-binary-classification>

# Appendices

* Gantt diagram.
* Description of code files.
* See github repo: <https://github.com/JonathanPablo/DataScientest_Sales-Churn_Project>
* (will be further filled, structured, put together and cleaned up)

1. See: <https://medium.com/aiskunks/categorical-data-encoding-techniques-d6296697a40f#:~:text=It%20refers%20to%20the%20process,with%20text%20or%20categorical%20variables> [↑](#footnote-ref-2)
2. See: <https://www.simplilearn.com/normalization-vs-standardization-article> [↑](#footnote-ref-3)
3. See: <https://neptune.ai/blog/evaluation-metrics-binary-classification> [↑](#footnote-ref-4)