



# Customer Satisfaction of the German Energy Supply Chain

Predicting star ratings and company responses on Trustpilot

*Rendering 2*

Data Science Graduation Project @DataScientest, Paris  
Jan 26, 2024

Matthias Isele - Stefanie Arlt

## Outline

<b>Summary of Rendering 2.....</b>	<b>2</b>
<b>Data selection and preparation.....</b>	<b>4</b>
<b>Rating prediction based on customer feedback.....</b>	<b>6</b>
<i>Modeling problem and performance metric .....</i>	<i>6</i>
<i>Model choice and optimization .....</i>	<i>6</i>
<i>Sentiment analysis.....</i>	<i>9</i>
<i>Interpretation of results .....</i>	<i>9</i>
<i>Key word analysis .....</i>	<i>10</i>
<b>Prediction of number of words of response .....</b>	<b>13</b>
<i>Classification of the modeling problem .....</i>	<i>13</i>
<i>Model choice and optimization .....</i>	<i>13</i>
<i>Interpretation of results .....</i>	<i>15</i>
<b>Outlook .....</b>	<b>16</b>
<b>Appendix .....</b>	<b>17</b>
<i>Bibliography .....</i>	<i>17</i>
<i>List of figures .....</i>	<i>20</i>
<i>List of tables .....</i>	<i>20</i>
<i>Data Files and Contributions .....</i>	<i>21</i>
<i>Additional Figures.....</i>	<i>21</i>

## Abbreviations

- i.e. = that is
- e.g. = for example
- etc. = and others
- max. = maximum
- min. = minimum
- IQR = inter quartile range
- ML = machine learning
- PCA = principal component analysis
- MAE = Mean Absolute Error
- MSE = Mean Squared Error
- RMSE = Root Mean Squared Error
- $R^2$  = R-squared

## Summary of Rendering 2

We started from the data set as explored and explained in rendering 1 . The goal was to predict the star rating of customer posts from customer comments and to predict the length of company answers to customer posts, respectively the full answers, if possible.

After applying simple ML models for **rating prediction based on number of words in customer feedback** and its headline, we could reach not over 70% in accuracy. The support vector machine classifier and Random Forest classifier were applied both untuned and tuned, with SVM classifier models performing better but both lacking severely in the middle classes of this imbalanced data set.

Not only were these features insufficient but also the data set was too big and reached the limit of available computing capacity. Leveraging business practice and everyday use cases, the data was reduced in a step-by-step approach: using data with comments, then with comments and headlines, lastly exploring data only from one supplier. In addition, the multiclass target was transformed into a binary classification, dividing ratings 4 and 5 into the dominant class 1 and ratings 1 to 3 into class 0.

A significant improvement could be reached after applying **sentiment analysis** on the customer comment field, and then training the same models again. While we saw overfitting for both models, the results were overall much improved since both classes were predicted over 75 % correctly, for the dominant class even up to 90%, with accuracy reaching up to 88% overall.

Closely related to sentiment analysis is the **key word analysis**, which leverages the same techniques to extract meaningful comparing positive and negative comments between suppliers for reporting and visualization of the main interest points of the customers such as billing, service and metering. As the same keywords are used both for negative and positive comments, these could be used as input for large language model prompting to create individual responses.

The second task was to **predict the length of company answers or even the full answers, if possible**. As company answers are highly dependent on the answer policy of the company, we selected two sub data sets, the first one containing data for E.ON Energy, the second one for Octopus Energy. Predicting answers of E.ON Energy turned out to be a binary classification problem. There were only two distinct answers, up to trivial modifications like spaces and captions. Using a logistic regression on the numerical variables of comment length, headline length and star rating resulted in an astonishing accuracy of 99.82% on the total data set. This is due to a very strong correlation of the star rating to the company answer. Predicting the answer based on **sentiment analysis** on the comments could not improve the result, as we let go of the star rating variable. Still we reached a good accuracy of 85%.

In contrast, the answers of Octopus Energy were highly personalized which made it necessary to target the answer length instead of the full answers. Training several models on the numeric variables of comment length, headline length and star rating we reached RMSE's ranging from 0.47 to 0.55, the best model being linear regression and the worst model being decision tree regressor. Predicting the answer lengths based on **sentiment analysis** on the comment variable ranged in between with an RMSE of 0.52. Although, we expect sentiment analysis to be superior if more elaborate language models are used.

The immediate next step is model refinement. Sentiment analysis can be improved with better language comprehension models. In the case of modeling answers, a parameter search can be implemented.

Further, one can try to improve the modelling with advanced models like deep learning.

For the next project, one could integrate a pre-trained Language Model with an API to generate company answers to customer posts. One could give the model keywords that we identified in our keyword analysis as prompts to generate answers. The true company answers could work as a library for suggestion.

Other prompts could be length of answer, name of customer, focus words, key values of the company.

The answers of the model could be checked by a comparison sentiment analysis on both the generated and the true answers.

## Data selection and preparation

### Feature selection

We started from the data set as explored and explained in rendering 1<sup>1</sup>. The goal was

1. to predict the star rating of customer posts from customer comments and
2. to predict the length of company answers to customer posts, respectively the full answers.

For the first case it was necessary to restrict the data set to rows, where a comment exists, i.e. *Comment\_TF* = 1, for the second case, in addition also answers were required, i.e. *Comment\_TF* = 1 & *Answer\_TF* = 1.

Due to unique answer policies applied by some companies, it was helpful to select specific companies by the variable *Company*.

### Feature preparation

As explained in Rendering 1<sup>2</sup>, we applied the natural logarithm on the features word counts of comments, headlines and answers, building the engineered variables *log\_Words\_Comment*, *log\_Words\_Headline*, *log\_Words\_Answer*.

Furthermore, the min-max-scaler was applied to the target variable *Stars* creating the variable *Stars\_min\_max\_scaled*.

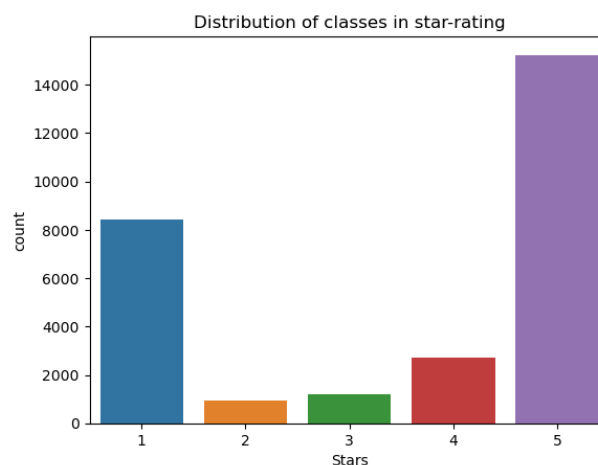


Figure 1: Class distribution in target variable

As seen in the chart above, the target variable 'Stars' is not evenly distributed in the dataset: Over 50% of the customer comments had 5-star ratings, followed by around 30% of 1-star ratings and around 10% of 4-star labels. Less than 5% each went to 3 and 2-star ratings.

Following the classical use case in business reports and analysis papers, where the focus is usually on the very best and the very worst, to define controls and measures for improvement, we grouped the star ratings into 'good' (*Stars*  $\geq$  4) and 'bad' (*Stars*  $<$  4) ratings<sup>3</sup>, consolidated in the Boolean variable *Stars\_geq4\_TF*.

---

<sup>1</sup> For details see Rendering 1, p. 6.

<sup>2</sup> Compare Rendering 1, p.2.

<sup>3</sup> Alternative solution would be to remove ratings 2-4 stars for a perfect training set, but the platform would deliver continuously new ratings with 2 to 4 stars, which makes this approach too theoretical.

## Dataset for machine learning

The following features were used for testing our ML models, a subset only for keyword analysis.

Label	Description	Appearance	Modelling Case 1	Modelling Case 2	Keyword Analysis
Headline	Headline of post	object	explanatory	explanatory	explanatory
Comment	Comment of post	object	explanatory	explanatory	explanatory
Answer	Answer to post	object	not used	target	explanatory
Company	Name of Company	object	selector	selector	selector
Comment_TF	Checks, if there is a comment, Boolean	Int64	= 1	= 1	selector
Answer_TF	Checks, if there is an answer, Boolean	Int64	Not used	= 1	selector
log_Words_Headline	Natural logarithm of word count of headline	float64	explanatory	explanatory	not used
log_Words_Comment	Natural logarithm of word count of comment	float64	explanatory	explanatory	not used
log_Words_Answer	Natural logarithm of word count of Answer	float64	not used	target	not used
Stars	Rating number of stars 1 - 5	Int64	not used	not used	selector
Stars_min_max_scaled	Min-max-scaled star rating	float64	target	explanatory	not used
Stars_geq4_TF	Checks for Stars > 3, Boolean.	Int64	target	explanatory	not used

Table 1: Feature description ML dataset

## Outlier handling

In the distribution of the main variables, there were some data points with extreme values, which were investigated with KDE and box plots, looking into distribution and interquartile range. The most extreme values were capped.<sup>4</sup>

## Dataset size and selection

To develop the most practical and appropriate approach, simple models were applied first on the complete data set, but due to long run times and differences in suppliers' answer policies, the following subsets were built, dividing the dataset down to a more manageable size while preserving logical cohesion.



Figure 2: Refinement of dataset size

This method corresponds to business practices where the target is usually limited to the company's own customer feedback, or a limited number of competitors in a benchmark analysis.

Other methods to reduce the dataset into more manageable portions could have been:

- Selection by date, e.g. only data younger than 2 years
- Limitation to a fixed line count supported by star ratings to maintain target class distribution.

<sup>4</sup> In most cases we had skewed distributions, so we looked at the inter-quartile range (IQR): Data points below  $Q1 - 1.5 * IQR$  or above  $Q3 + 1.5 * IQR$  can be considered as outliers. See also Dey, Akash: How to handle outliers, Feb 2022: <https://www.kaggle.com/code/aimack/how-to-handle-outliers/notebook>.

# Rating prediction based on customer feedback

## Modeling problem and performance metric

As a target of this ML problem, we will try to predict the star rating based on customer feedback, as available and engineered in the steps described earlier.

The rating prediction based on customer feedback was a **classification problem**, with **accuracy** and **F1-score** as the **main performance metric**.

“Precision measures the extent of error caused by False Positives (FPs) whereas recall measures the extent of error caused by False Negatives (FNs).”<sup>5</sup> Recall measures the model’s ability to detect positive samples, which does not consider other classes. Even though precision includes all classes for measuring how reliable the model is classifying one class, both metrics are focusing on one class of interest, and we must consider here more than one class.

Accuracy describes how the model performs over all classes as it calculates the ratio between the number of correct predictions to the total number of predictions. In other words, the focus is here on penalizing the FPs and FNs.<sup>6</sup> In our case, we need to concentrate on True Positives (TPs) and True Negatives (TNs), so we decided on accuracy as main performance metric but also not discarding F1-score for comparing different models.

“F1-score is balancing precision and recall on the positive class while accuracy looks at correctly classified observations of both positive and negative.”<sup>7</sup>

## Model choice and optimization

Support vector machine (SVM) from scikit-learn ensemble was chosen first because we have limited features, and we were looking for a simple but effective model which would also support multi-class classification.

Preparation consisted only of standardization and the data set must only consist of numerical features.<sup>8</sup>

As second model we tried RandomForestClassifier also from scikit-learn ensemble which is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. With this model we kept all numerical features and did not need scaling, as it is a tree-based model.<sup>9</sup>

---

<sup>5</sup> See Zeya LT: Essential things you need to know about F1-score, Towards Data Science, Nov 2021: <https://towardsdatascience.com/essential-things-you-need-to-know-about-f1-score-dbd973bf1a3>

<sup>6</sup> Compare Huilgol, Purva: Accuracy vs. F1-Score, Medium Aug 2019: <https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2>.

<sup>7</sup> See Czakob, Jakub: F1 Score vs ROC AUC vs Accuracy vs PR AUC: Which Evaluation Metric Should I Choose?, Neptune.ai Blog Sep 2023: <https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc#:~:text=F1%20score%20vs%20Accuracy,observations%20both%20positive%20and%20negative>

<sup>8</sup> Compare user guide support vector machines: <https://scikit-learn.org/stable/modules/svm.html>; cited as Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

<sup>9</sup> Compare Sklearn documentation on RandomForestClassifier: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>.

## Model training and observations

Applying `train_test_split` from `sklearn.model_selection` we specified a test set of 20%.

The first simple ML models applied for rating predictions for all 5-star ratings showed poor results since we faced a multiclass qualification problem with an imbalanced data set. Neither application of hyperparameter tuning and cross validation did improve the results, nor adding more features for model training.

In addition, run times were very long and handling of the code impractical, due to the initially very large size of the data set.


Model	Dataset	Performance metric before tuning		Performance metric after tuning <sup>10</sup>	
		accuracy	F1-score	accuracy	F1-score
SVM Classifier	All suppliers, only data with answers; Words_Comment	0.69	Class 1: 0.69 Class 2 to 4: 0.00 Class 5: 0.79	0.59	Class 1: 0.65 Class 2: 0.10 Class 3: 0.09 Class 4: 0.03 Class 5: 0.76
		No overfitting		No overfitting	
SVM Classifier	All suppliers, only data with answers; Words_Comment, Words_Headline	0.70	Class 1: 0.69 Class 2 to 4: 0.00 Class 5: 0.79	0.54	Class 1: 0.66 Class 2: 0.09 Class 3: 0.10 Class 4: 0.13 Class 5: 0.70
		No overfitting		No overfitting	
Random Forest Classifier	All suppliers, but only data with answers; Words_Comment, Words_Headline	0.67	Class 1: 0.67 Class 2: 0.00 Class 3: 0.01 Class 4: 0.00 Class 5: 0.78	0.67	Class 1: 0.73 Class 2: 0.00 Class 3: 0.01 Class 4: 0.00 Class 5: 0.78
		Slight overfitting		Notable overfitting	

Table 2: Performance metrics for multi-class prediction of star ratings (simple models)

Hyperparameter tuning, cross validation and boot strapping were considered with the tuning step, considering the multiclass classification and the imbalanced data set.

As we can see, the 1- and 5-star ratings were predicted best. With hyperparameter tuning we saw a small improvement on the middle ratings while dominant classes performed a little less. RandomForestClassifier did better while we encountered an overfitting issue with the SVM classifier models. However, overall performance was not much over 50% for the dominant ratings 1 and 5 which is very close to random reliability.

The greater the imbalance between classes, the less successful the classical models will be in predicting the minority class. Therefore, applying undersampling was considered to increase the number of observations of the minority classes, as the data set was sizable enough.

However, the focus for customer service and customer satisfaction KPIs lies in the extreme opinions: “Is the customer really happy or really dissatisfied?” With this guideline for improvement, the middle star ratings, which were quantitatively low, could be neglected.

<sup>10</sup> Tuning for imbalanced data set and multi-class decision for SVM. Tuning with GridSearchCV with 3 fold crossvalidation for RandomForestClassifier.



To improve performance, the following changes were implemented:

- The size of the dataset was decreased as explained above<sup>11</sup>. Following everyday use cases, the data set for prediction would most likely contain only customer feedback from one supplier.
- As star ratings were almost binary distributed, we simplified the modelling to a binary classification problem targeting the *Stars\_geq4\_TF* variable.
- Because we expect the star rating of the customer to be closely tied to the sentiment of the customer's comment, we apply a sentiment analysis to the explanatory variable *Comment*.

In a first step, model performance was checked with a reduced data set limited to E.ON customer feedback, also applying natural logarithm to the word count features before capping the extreme values. The distribution of classes between positive (1) and negative (0) classes was still imbalanced, with nearly 70 % of the customer comments deriving from 4 or 5-star ratings and 30% going to 1- to 3-star ratings.

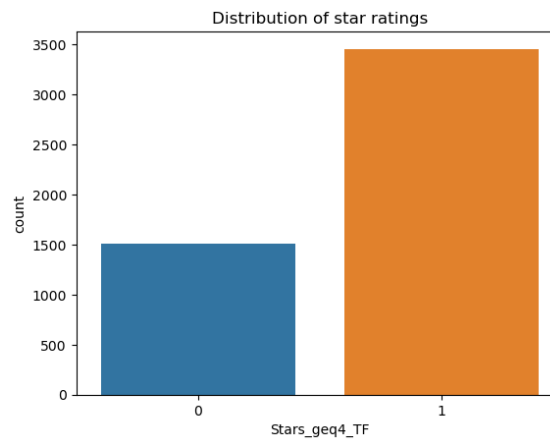


Figure 3: Class distribution in E.ON customer feedback

The limited and for one supplier specialized data was much easier to predict, reaching an accuracy around 80%.

Even when adding the information from word count of headline to the comment feature, we could see only small improvement for the non-dominant class 0 to the detriment of class 1, as the imbalance could not be overcome. Again the SVM classifier showed the best result, while we saw some overfitting with the Random Forest model.

Model	Dataset	Performance metric before tuning		Performance metric after tuning <sup>12</sup>	
		accuracy	F1-score	accuracy	F1-score
<b>SVM Classifier</b>	E.ON all data; <i>log_Words_Comment</i>	0.79 No overfitting	Class 0: 0.65 Class 1: 0.88	0.79 No overfitting	Class 1: 0.69 Class 5: 0.84
<b>SVM Classifier</b>	E.ON all data; <i>log_Words_Comment</i> , <i>Log_Words_Headline</i>	0.82 No overfitting	Class 0: 0.63 Class 1: 0.88	0.79 No overfitting	Class 1: 0.70 Class 5: 0.84
<b>Random Forest Classifier</b>	E.ON all data; <i>log_Words_Comment</i> , <i>Log_Words_Headline</i>	0.76 Slight overfitting	Class 0: 0.59 Class 1: 0.84	0.77 Slight overfitting	Class 1: 0.60 Class 5: 0.84

Table 3: Performance metrics for binary prediction of star ratings (simple models)

<sup>11</sup> See description and figure 2 in chapter Dataset size and selection.

<sup>12</sup> Tuning for imbalanced data set and multi-class decision for SVM. Tuning with GridSearchCV with 3 fold crossvalidation for RandomForestClassifier.

## Sentiment analysis

Most customers included in *Headline* only a few words or a summary sentence but wrote extensively about problems and good experiences in the *Comment* field. So, we chose the feature *Comment* to undergo sentiment analysis because it contained more detailed and therefore more meaningful content than the field *Headline* with its character limitation on the Trustpilot platform.

All texts in *Comment* were prepared with the following steps:

- With regex, all non-letters in the comments were replaced by spaces, consecutively removing words of length 2 or less.
- Each comment was converted to lowercase.
- Most common fill words were removed by `stop_words` function with German settings

Comment	Comment_alpha	Comment_no_stopwords
Korrekte Auflistung des Zählerstandes und Verb...	korrekte auflistung des zählerstandes und verb...	korrekte auflistung zählerstandes verbrauchs g...
Leichte Eingabe der Daten und schneller Wechse...	leichte eingabe der daten und schneller wechse...	leichte eingabe daten schneller wechsel abschl...
Hallo,Leider muss man sich mehr Fach mit dem S...	hallo leider muss man sich mehr fach mit dem s...	hallo leider mehr fach service wenden geschätz...
Ich bin rundum zufrieden mit e-on. Umzug mit Ü...	ich bin rundum zufrieden mit umzug mit übersch...	rundum zufrieden umzug überschneidung geklappt...
Alles korrekt. Allerdings erscheint bei uns n...	alles korrekt allerdings erscheint bei uns nur...	korrekt allerdings erscheint frochkönigweg pc...

Figure 4: Text transformation during sentiment analysis

After creating a test and training set, the filtered customer comments were converted to numerical columns with `CountVectorizer` from `sklearn.feature_extraction`. This new feature set was then applied to train SVM classifier and Random Forest classifier, again in two steps with and without tuning.

While we saw overfitting for both models, the results were overall much improved and far away from random guesses. Both classes were predicted over 75 % correctly, for the dominant class even up to 90%, with accuracy reaching up to 88% overall.

Model	Dataset	Performance metric before tuning		Performance metric after tuning <sup>13</sup>	
		accuracy	F1-score	accuracy	F1-score
<b>SVM Classifier</b>	E.ON all data;	0.88	Class 1: 0.78	0.88	Class 1: 0.81
	<i>Comment_no_stopwords</i>	Some overfitting	Class 5: 0.92	Some overfitting	Class 5: 0.92
<b>Random Forest Classifier</b>	E.ON all data;	0.88	Class 1: 0.77	0.88	Class 1: 0.79
	<i>Comment_no_stopwords</i>	Abundant overfitting	Class 5: 0.91	Abundant overfitting	Class 5: 0.92

Table 4: Performance metrics for Prediction of star ratings (sentiment analysis)

## Interpretation of results

The reduction of the data set to one supplier and setting the target feature up as a binary problem helped significantly in improving performance of the simple machine learning models.

As expected, sentiment analysis on the *Comment* feature was the correct approach to predict star ratings. In case of not having entries in this column, the mandatory field *Headline* would be a good substitute, as we found here in most cases the content being a summary of the *Comment* content.

<sup>13</sup> Tuning for imbalanced data set and multi-class decision for SVM. Tuning with `GridSearchCV` with 3 fold crossvalidation for `RandomForestClassifier`.

## Key word analysis

Closely related to sentiment analysis is the key word analysis, which leverages the same techniques to extract meaningful content which can support customer satisfaction reportings and deliver input for improvement initiatives.

To identify the most common topics in customer feedback and ratings, we were looking for key words in the headlines, comments, and supplier responses. In this case it made sense to focus on one supplier only, not only to keep the data volume manageable but also due to individual approaches to replying to customer feedback: We looked at E.ON Energie Deutschland GmbH first with nearly 5,000 lines, followed by Octopus Energy Germany with 6,000 lines of feedback.

The process started always with checking duplicates for each feature and creating a string containing the concatenation of all entries in the text column of the data set, inserting thereby a space between each line. Applying a function to streamline the process, the following steps were performed:

- Converting the text to lowercase characters only with the casefold method
- Cleaning a text of special characters, numbers etc. by regex and character mapping
- Tokenizing the content of the string by the TweetTokenizer from nltk.tokenize library
- Filtering the word token list with the stop\_words file specified for each supplier and German language setting.

Separating the comments by star rating allowed to compare 10586 positive and 2480 negative words of feedback, which showed similar key words for both features *Headline* and *Comment*.



Figure 5: Positive and negative feedback for E.ON (feature Headline)

For more advanced visualizations, the function was expanded to create a data frame listing every word with its number of occurrences, sorted by word count in descending order. With help of a color dictionary leveraging matplotlib color palettes, this allowed to compare directly key words and cluster information.<sup>14</sup>

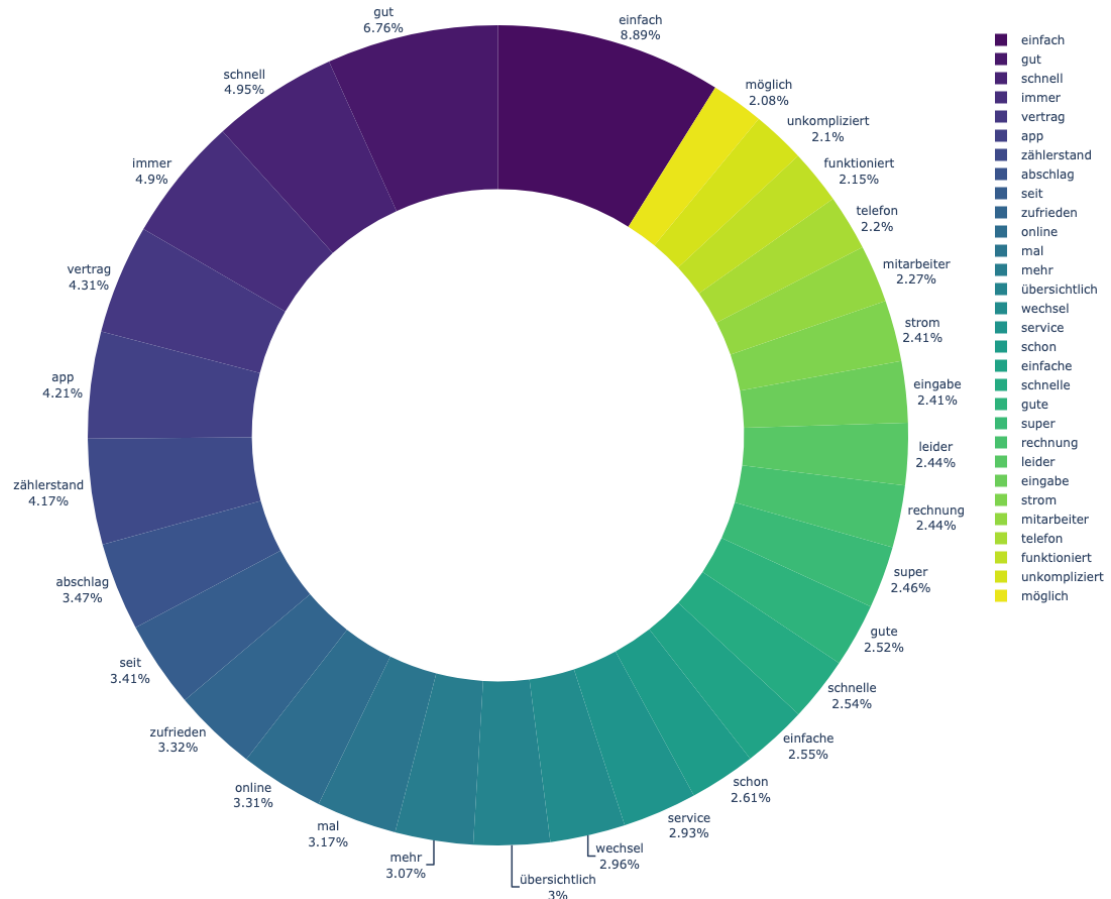


Figure 6: 30 most mentioned words in E.ON customer comments

Compared to classic word clouds the content is more quantified and systematic. This is especially helpful when comparing positive and negative comments between suppliers, as in the chart below, where 63178 key words in customer comments have been separated into 45867 positive and 4719 negative tokens.

Customers praise the easy (einfach, einfacher), uncomplicated (unkompliziert, reibungslos, problemlos, problemloser) and fast (schnell, schneller) transactions like change of supplier (Anbieterwechsel, Abwicklung, Ablauf), also applicable for contact with Customer service (Kundenservice, Service, Kommunikation, Bearbeitung). Adjectives like good (gut, toll, super), friendly (freundlich), to be recommended (empfehlenswert) and top describe very positive feelings; also the price (Preis) is mentioned.

<sup>14</sup> For color dictionary and visualization ideas, see Boriarn, K.: Beyond the Cloud: 4 Visualizations with Python to use instead of Word Cloud, Towards Data Science Jul 2022: <https://towardsdatascience.com/beyond-the-cloud-4-visualizations-to-use-instead-of-word-cloud-960dd516f215>

Whereas on the complaint side, there have been issues with longer waiting times (wochen, monat, monate), concerning contract (Vertrag, Grundversorgung, gekündigt, wechsel), invoice (abrechnung) and final billing (endabrechnung, jahresabrechnung). The impression from the headline section is confirmed; in addition to the headlines, we see here also an issue with the email communication(mails, mail, email).

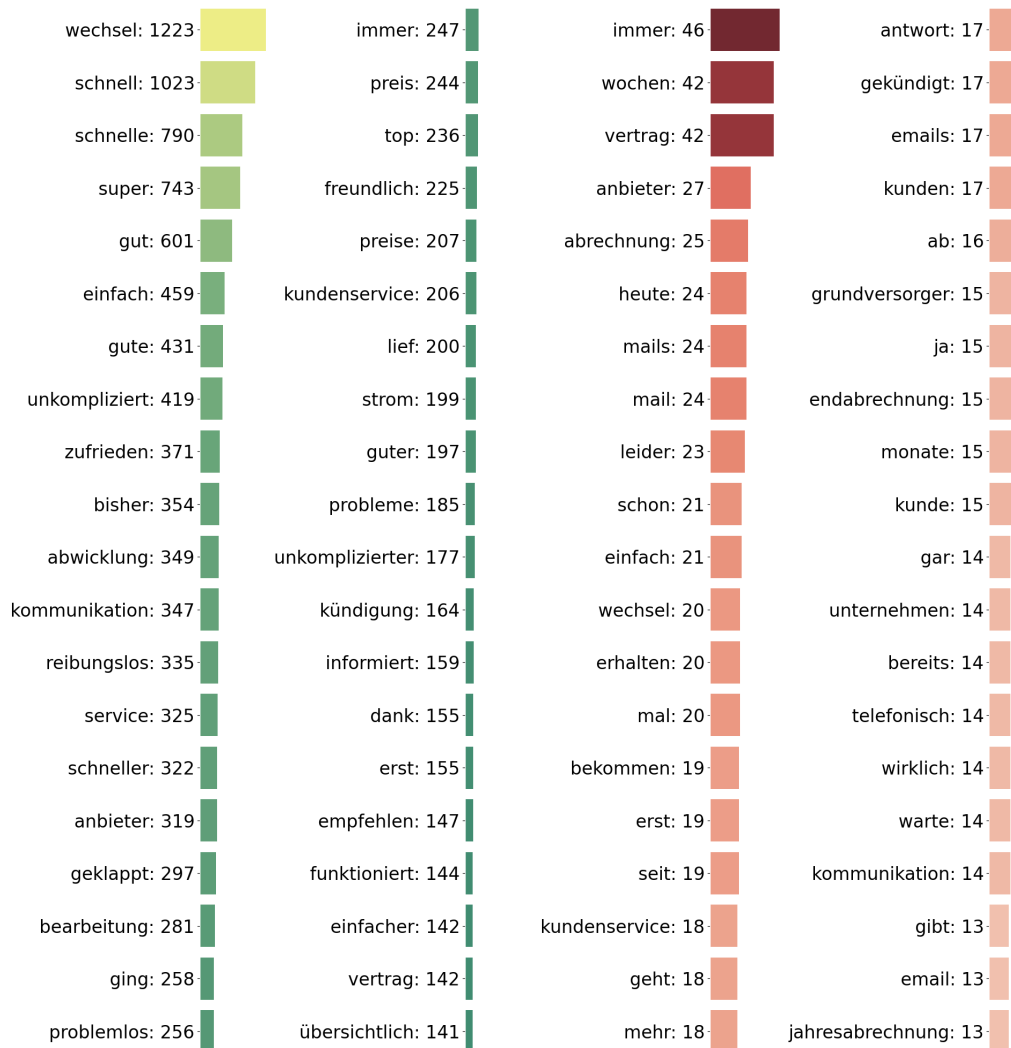


Figure 7: Word count of keywords in positive and negative customer comments for Octopus Energy

## Prediction of number of words of response

### Classification of the modeling problem

The goal is to predict the length of company answers to customer posts, respectively the full answers, if possible. Hence, we restrict the data set to posts (rows) where a comment and an answer exists, i.e.  $Comment\_TF = 1$  &  $Answer\_TF = 1$ . Recall that the answer policy is company specific (Rendering 1, p.13). Therefore, we choose the companies E.ON Energy and Octopus Energy Germany for further investigation, creating a data set for each using the *Company* variable. These are the companies with the most entries, 4965 and 8059, respectively. It turns out that these companies require rather contrary modelling approaches.

#### E.ON Energy

The E.ON data set surprisingly turns out to be a binary classification problem. There are only two pairwise unique answers in the *Answer* variable, up to trivial modifications like spaces and captions. (In total there are six pairwise unique answers.) This makes it possible to predict the full *Answer* variable, i.e. there is no need to simplify the target by considering *log\_Words\_Answer*.

For E.ON energy, the prediction of answers is a **binary classification problem**, with **accuracy** and **F1-score** as the **main performance metrics**.

We want to maximize the True Positives (TPs) and True Negatives (TNs), so we decided on accuracy as main performance metric but also not discarding F1-score for comparing different models. For a deeper look into classification performance metrics see p. 6.

#### Octopus Energy

The answers of Octopus Energy Germany are highly personalized. Company answers reference user names directly. In the 5- and 4-star regime there seem to be standard answers (up to user names), but below, answers are personalized to a very high degree. To simplify the modelling, we choose the target variable *log\_Words\_Answer*.

For Octopus Energy, the prediction of *log\_Words\_Answer* is a **regression problem**, with **root mean squared error (RMSE)** as the **main performance metric**.

The RMSE is sensitive to outliers. However, this is not a problem as outliers are tamed by the natural logarithm. The square root accounts for the errors being in the same order of magnitude as the data.

### Model choice and optimization

#### E.ON Energy

First, we replace the two possible answers in the *Answer* variable by 0 and 1, converting the Dtype of *Answer* to int64. The now binary target *Answer* is closely tied to the star rating. The variable *Stars\_geq4\_TF* is Pearson-correlated to *Answer* by 0.9957. In fact, there are only 9 cases out of 4965 where *Stars\_geq4\_TF* and *Answer* do not match. These cases occur when people confuse the star rating (5 is the best) with the German grading system (1 is the best), leading to comments which are contrary to the ratings. E.ON Energy gave the correct answers to the sentiment of the comment, not to the star rating. This means they either use a strong sentiment

analysis model on the comments for automatized answers, or a human assigns the two standard answers manually.

In a first step we try to capture the strong relationship of the target variable to the star rating using logistic regression on the numeric columns *log\_Words\_Comment*, *log\_Words\_Headline*, *Stars\_geq4\_TF*. The test set is 20% of the total population. The model with default hyper parameters suppresses the first two numeric columns. It reduces to be a copy of *Stars\_geq4\_TF*. It is exactly the 9 cases discussed that it cannot predict correctly, neither on the training, nor on the test set. Still, we reach an astonishing accuracy on the total data set of  $1 - 9/4965 = 99.82\%$ .

We tried to improve the model by projecting the three explanatory variables on a 2- and 1-dimensional subspace using principal component analysis, which failed.

Our last try is sentiment analysis on the *Comment* variable. Using regex, we replace all non-letters by spaces, consecutively removing words of length 2 or less. Each comment is converted to lowercase. We filter for german stop words and replace the special german characters ä, ö, ü, ß, by ae, oe, ue, ss. We convert the column to numerical columns with CountVectorizer from sklearn.feature\_extraction.text. These numerical columns derived from *Comment* are used to train a GradientBoostingClassifier with respect to a test set size of 20% and hyperparameters *n\_estimators*=100, *learning\_rate*=1, *max\_depth*=1. On the test set we obtain an accuracy of  $(188+643)/(51+111+188+643) = 85\%$ .

#### Octopus Energy

The first approach is to predict answer lengths *log\_Words\_Answer* on the numerical variables *log\_Words\_Comment*, *log\_Words\_Headline*, *Stars\_min\_max\_scaled*. A standard scaler is applied on the two logarithmic variables. We will check the performance of several models (default hyperparameters) on a test set of test size 20%. This includes a custom model defined as follows: On the training set, compute the averages of *log\_Words\_Answer* grouped by *Stars\_min\_max\_scaled*. On the test set, the predictions are defined as the computed averages (learned from the test set) rise to *Stars\_min\_max\_scaled*. The performance metric is the square root of the mean squared error, i.e. the cartesian distance of the prediction- to the test-vector. The results are collected in the following table.

Model	Performance metric: RMSE
XGBRegressor	0.5006
RandomForestRegressor	0.5036
DecisionTreeRegressor	0.5516
LinearRegression	0.4732
Custom Model	0.4972

Table 5: Performance metrics for Prediction of number of words of response

We try to improve the situation with sentiment analysis. Following the same procedure on the *Comment* column as in the last section, but with a GradientBoostingRegressor, we get a performance of 0.5211.

## Interpretation of results

### E.ON Energy

The star rating variable *Stars\_geq4\_TF* is already a really strong indicator for the company answer with an accuracy of 99.82% on the total data set. A logistic regression model is able to mimic the variable *Stars\_geq4\_TF*, reducing to the identity after training. Using sentiment analysis with CountVectorizer and GradientBoostingClassifier, the accuracy 85% is worse, but still good. Language comprehension models that are more sophisticated should improve the accuracy. It seems that E.ON Energy either uses a really advanced language model or a human decides which of the two standard answers are replied.

### Octopus Energy

The model that performed the best to predict *log\_Words\_Answer* from numeric columns is LinearRegression with an RMSE of 0.47. A similar sentiment analysis but with a GradientBoostingRegressor leads to an RMSE of 0.52. This is slightly worse, though we assume that sentiment analysis should beat LinearRegression if a more sophisticated model than CountVectorizer is used.



## Outlook

Monitoring customer feedback and regularly compare the situation are key objectives in customer service management. With the analysis and predictive modeling in this scraped data set of Trustpilot ratings and comments for Energy Suppliers in Germany, we have simulated two typical use cases:

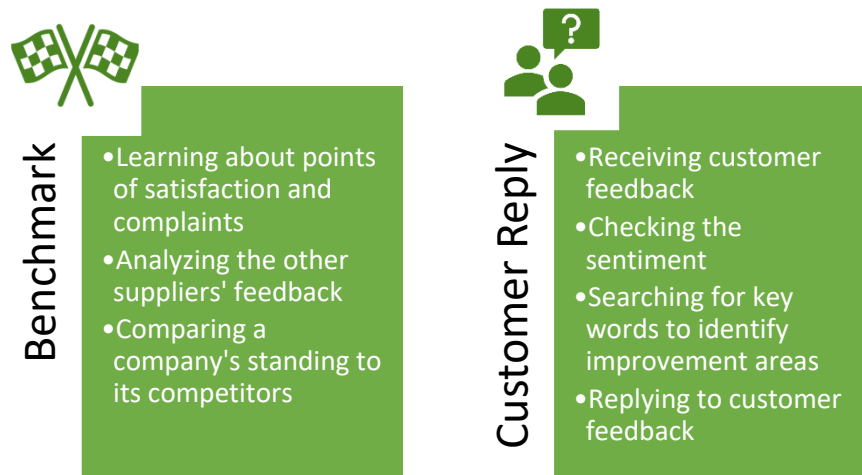


Figure 8: Use Cases Customer Management

For both uses cases named above, double-checking the rating with the sentiment of the written comments is important, as some customers might have misunderstood the numbers for grades instead as rating points, and the written feedback should match rather the comments than the rating itself.

Looking for expected or important key words is the basis for the next shared step: Identifying points of interest to compare keywords or take them as input for a response to the customer.

We have learned that it makes sense to limit the data to one supplier, to accommodate for a supplier's specific product portfolio but also to compare within the same reference system. Consequently, the ML models needed to be trained for one supplier very efficiently and achieve high and reliable performance in rating prediction, e.g. as part of a sentiment analysis.

The next steps for Customer Reply could be to collect the previous answers as a library to design new and individualized answers, which could be prompted to a large language model, together with expected length, customer name, focus words and company values such as #gerneperdu, i.e. personal address, or eco-friendly marketing messages.

For Benchmarking, we have shown already that some keywords are shared for both positive and negative feedback. Nevertheless, in case of specific complaints accumulating, warning systems could be triggered to rectify what was crooked as soon as possible. Distribution of information and quantified basis for decision making can be achieved more easily with pre-set dashboards and visualizations in reports.

To summarize: In times of consumer markets and price comparison portals like Check24, customer expectations have evolved to personalized answers and timely feedback also on external platforms as the new normal. Therefore, the overall goal to turn happy customers into loyal customers can only be achieved by constantly checking competitors and applying AI supported customer service processes.

# Appendix

## Bibliography

This is an annotated list of literature and links supporting the research.

### Webscraping Target Site

- Trustpilot Website Search categories: <https://support.trustpilot.com/hc/de/articles/360022026634>
- Trustpilot transparency report:  
[https://assets.ctfassets.net/b7g9mrbfayuu/tHyJSsKiNJxZvAuGPr6hz/5c6a42f3719debd02ca25989a7225222/Trustpilot\\_Transparency\\_Reporting\\_Active\\_Recipients\\_under\\_the\\_Digital\\_Services\\_Act\\_-\\_21\\_March\\_2023.pdf](https://assets.ctfassets.net/b7g9mrbfayuu/tHyJSsKiNJxZvAuGPr6hz/5c6a42f3719debd02ca25989a7225222/Trustpilot_Transparency_Reporting_Active_Recipients_under_the_Digital_Services_Act_-_21_March_2023.pdf)
- Trustpilot energy suppliers: [https://de.trustpilot.com/categories/electric\\_utility\\_company](https://de.trustpilot.com/categories/electric_utility_company)
- Official overview energy market actors (Bundesnetzagentur):  
<https://www.marktstammdatenregister.de/MaStR/Akteur/Marktakteur/IndexOeffentlich>

### Outlier handling

- Grace-Martin, Karen: Outliers: to drop or not to drop, 2018:  
<https://www.theanalysisfactor.com/outliers-to-drop-or-not-to-drop/>
- Sharma, Natasha: Ways to Detect and Remove the Outliers, May 22, 2022:  
<https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba>
- Dey, Akash: How to handle outliers, Feb 2022: <https://www.kaggle.com/code/aimack/how-to-handle-outliers/notebook>
- IQR score: [https://en.wikipedia.org/wiki/Interquartile\\_range](https://en.wikipedia.org/wiki/Interquartile_range)
- Z-Score: [https://en.wikipedia.org/wiki/Standard\\_score](https://en.wikipedia.org/wiki/Standard_score)

### Performance metrics

- Javatpoint tutorial on precision and recall: <https://www.javatpoint.com/precision-and-recall-in-machine-learning>
- Zeya LT: Essential things you need to know about F1-score, Towards Data Science, Nov 2021:  
<https://towardsdatascience.com/essential-things-you-need-to-know-about-f1-score-dbd973bf1a3>
- Huilgol, Purva: Accuracy vs. F1-Score, Medium Aug 2019: <https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2>
- Czakob, Jakub: F1 Score vs ROC AUC vs Accuracy vs PR AUCH: Which Evaluation Metric Should I Choose?, Neptune.ai Blog Sep 2023: <https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc#:~:text=F1%20score%20vs%20Accuracy,observations%20both%20positive%20and%20negative>
- Brownlee, Jason: Regression Metrics for Machine Learning, Jan 20, 2021:  
<https://machinelearningmastery.com/regression-metrics-for-machine-learning/>

## Model choice and optimization

- User guide support vector machines: <https://scikit-learn.org/stable/modules/svm.html>; cited as Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
- Deepthi, A.R.: Support Vector Machines & Imbalanced Data. How does SVM work in the case of an imbalanced dataset?, Medium 2019: <https://towardsdatascience.com/support-vector-machines-imbalanced-data-feb3ecffb0e>
- Sklearn documentation on C support vector classification: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
- Sklearn documentation on RandomForestClassifier: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- Filho, Mario: Does Random Forest Need Feature Scaling or Normalization?, Forecastgy Jun 2023: <https://forecastgy.com/posts/does-random-forest-need-feature-scaling-or-normalization/#:~:text=Random%20Forest%20is%20a%20tree,can%20be%20skipped%20during%20preprocessing.>

## Parameter optimization

- Sklearn documentation on cross validation: [https://scikit-learn.org/stable/modules/cross\\_validation.html#stratified-k-fold](https://scikit-learn.org/stable/modules/cross_validation.html#stratified-k-fold)
- Imbalanced data sets: <https://medium.com/sfu-csmp/surviving-in-a-random-forest-with-imbalanced-datasets-b98b963d52eb>
- Sklearn documentation on Stratified K-Fold: [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.StratifiedKFold.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html)
- [https://en.wikipedia.org/wiki/Stratified\\_sampling](https://en.wikipedia.org/wiki/Stratified_sampling)
- HalvingGridSearchCV: <https://towardsdatascience.com/stop-using-grid-search-cross-validation-for-hyperparameter-tuning-b962160dd6ae>
- Sklearn documentation on HalvingGridSearchCV: [https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.HalvingGridSearchCV.html#sklearn.model\\_selection.HalvingGridSearchCV](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.HalvingGridSearchCV.html#sklearn.model_selection.HalvingGridSearchCV)
- Kumar, Satyam: 20x times faster Grid Search Cross-Validation. Speed-up your cross-validation workflow with Halving Grid Search, Towards Data Science May 2021: <https://towardsdatascience.com/20x-times-faster-grid-search-cross-validation-19ef01409b7c>

## Keyword analysis

- Brownlee, Jason: A Gentle Introduction to the Bag-of-Words Model, Machine Learning Mastery Aug 2019: <https://machinelearningmastery.com/gentle-introduction-bag-words-model/>
- TZJY: Natural Language Processing: Bag-Of-Words, Medium Oct 2021: <https://medium.com/@tzjy/natural-language-processing-bag-of-words-python-code-included-ed3cfe979d2>
- Boriarn, K.: Beyond the Cloud: 4 Visualizations with Python to use instead of Word Cloud, Towards Data Science Jul 2022: <https://towardsdatascience.com/beyond-the-cloud-4-visualizations-to-use-instead-of-word-cloud-960dd516f215>
- Luvsandorj, Zolzaya: Simple word cloud in Python, Towards Data Science, Jun 2020: <https://towardsdatascience.com/simple-wordcloud-in-python-2ae54a9f58e5>
- Choosing colormaps in Matplotlib: <https://matplotlib.org/3.2.1/tutorials/colors/colormaps.html>

## List of figures

Figure 1: Class distribution in target variable.....	4
Figure 2: Refinement of dataset size.....	5
Figure 3: Class distribution in E.ON customer feedback .....	8
Figure 4: Text transformation during sentiment analysis .....	9
Figure 5: Positive and negative feedback for E.ON (feature Headline).....	10
Figure 6: 30 most mentioned words in E.ON customer comments .....	11
Figure 7: Word count of keywords in positive and negative customer comments for Octopus Energy.....	12
Figure 8: Use Cases Customer Management .....	16
Figure 9: Energy suppliers on TrustPilot website .....	21
Figure 10: TrustPilot rating site "Octopus Energy" .....	22
Figure 11: Customer vote with supplier feedback on TrustPilot.....	22

## List of tables

Table 1: Feature description ML dataset.....	5
Table 2: Performance metrics for multi-class prediction of star ratings (simple models) .....	7
Table 3: Performance metrics for binary prediction of star ratings (simple models) .....	8
Table 4: Performance metrics for Prediction of star ratings (sentiment analysis).....	9
Table 5: Performance metrics for Prediction of number of words of response .....	14

## Data Files and Contributions

All data files are available for download on GitHub.

## Additional Figures

### TrustPilot Website

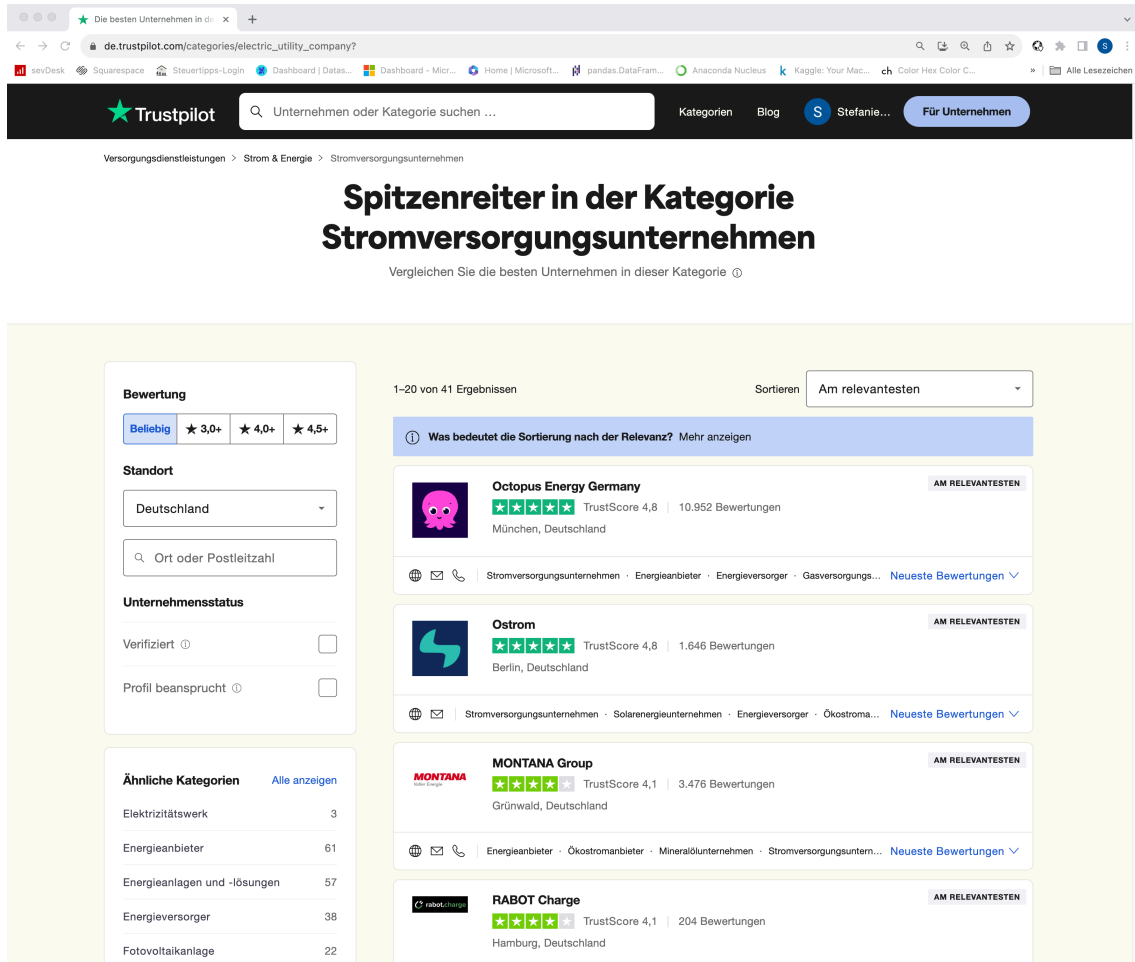


Figure 9: Energy suppliers on TrustPilot website

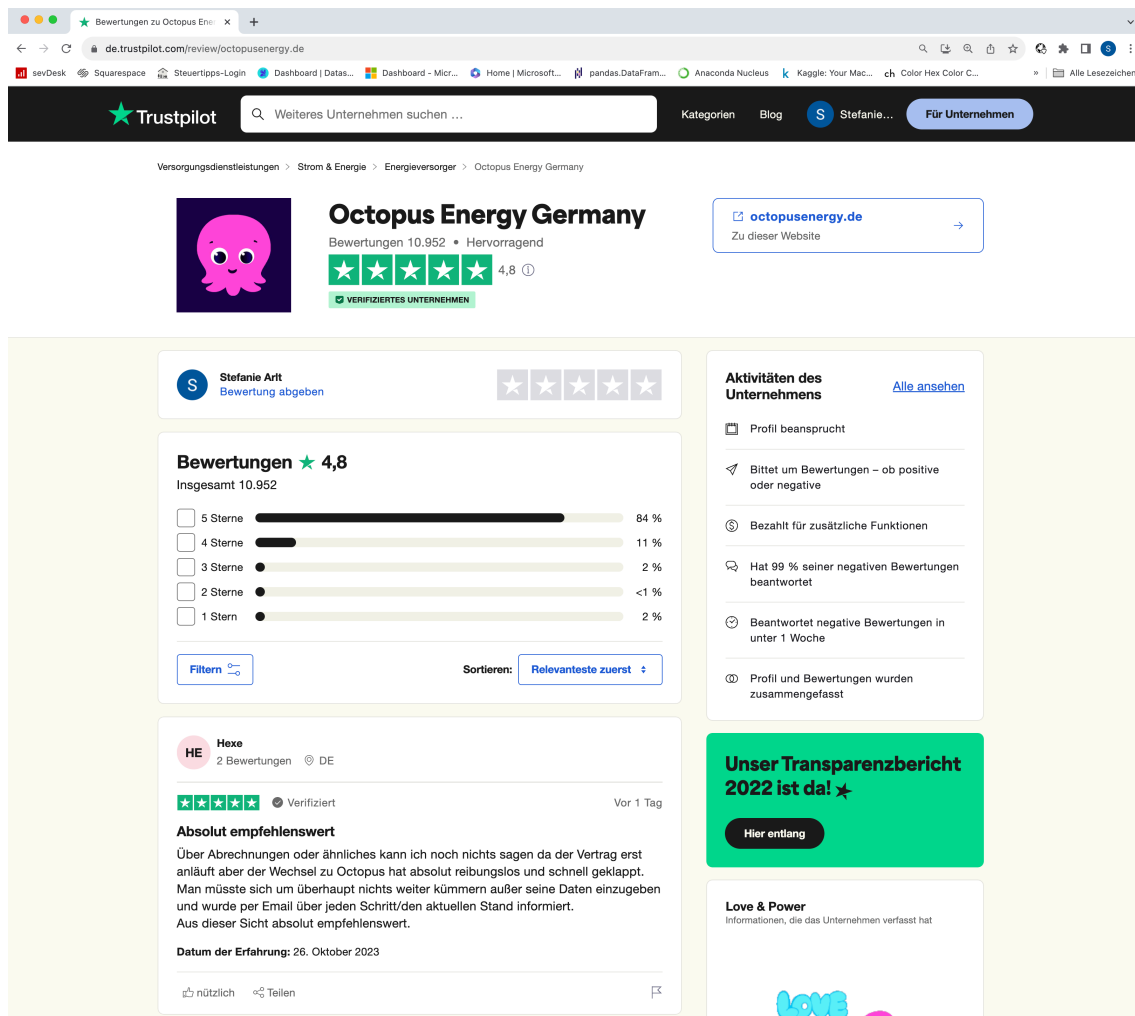


Figure 10: TrustPilot rating site "Octopus Energy"

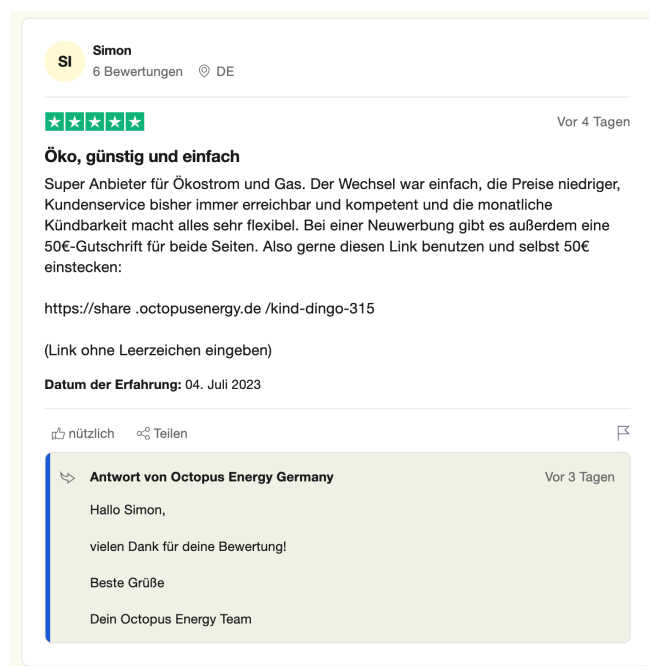


Figure 11: Customer vote with supplier feedback on TrustPilot