Making it explicit

Comparison of prediction abilities for different AI-methods

Christoph Sonn
Julius Heitmann

November 2019
Blogpost

Key Facts

- In order to implement meaningful and high-quality data analyses, data preparation is of outstanding importance - but it is time-consuming and costly
- One task of data preparation is Feature Engineering – an optional process which helps to make implicitely inherent information of the model explicitly accessible
- Feature Engineering requires the application of domain or expert knowledge to make information explicitly available
- In this Blogpost Feature Engineering will be examined in the context of different models for information extraction of temporal data
- The consideration of domain knowledge for Feature Engineering appears to be both useful and essential for the creation of more precise analyses, even when taking into account the associated effort. However, the success of Feature Engineering depends on the algorithm used

Introduction

In times of increasing efficiency of business processes due to digitalization of information flows, artificial intelligence represents a valuable possibility to realize further efficiency gains. Naturally there is not only media attention surrounding Artificial Intelligence (AI) but also great interest from enterprises in the application of AI. An important field of application is the automation of decisions. In order to achieve this goal, data must be collected and prepared during a data analysis process in order to enable the training of a corresponding model. The process of data preparation is time-consuming and complex.

According to a study by Forbes magazine, data analysts spend about 80% of their time on data preparation, which includes the steps "data cleaning", "data preparation" and "feature engineering". While the first two elements are necessary to enable the analysis in the first place, feature engineering is an optional process that makes information inherently in the model accessible. In this blog post, we will investigate whether such a process step, which requires domain knowledge and thus skilled personnel, is worth the effort, whether current models are capable of extracting this information themselves, or whether this step is even essential for a successful data analysis project.

Feature engineering is the process of extracting information (features) from data which is already implicitly available. Domain knowledge is required for this process. By applying models based on expert knowledge, the features are extracted and explicitly added to the data set. This makes the information accessible to algorithms. A good example, since every reader of the present blog post has domain knowledge in this area, is the extraction of features from a date, which will be further investigated later in the blog post. For an algorithm, a set of data is an timely ordered set of data points. Only the information about the relative temporal position and possibly the information about day, month and year are accessible. For humans, a date contains information about, for example, the day of the week, workday, or holiday. With suitable functions, this information can be displayed in the form of an additional attribute.

In this blog post, the effectiveness of feature engineering in combination with different models of data analysis will be investigated. For this purpose, the COREai team investigated the analysis and evaluation of customer needs based on transaction data of a Brazilian payment service provider. The topic of customer needs prediction has a crucial importance in the financial environment, among others. The analysed data set was made available in the context of a public tender (Kaggle Challenge - Elo Merchant Category Recommendation).

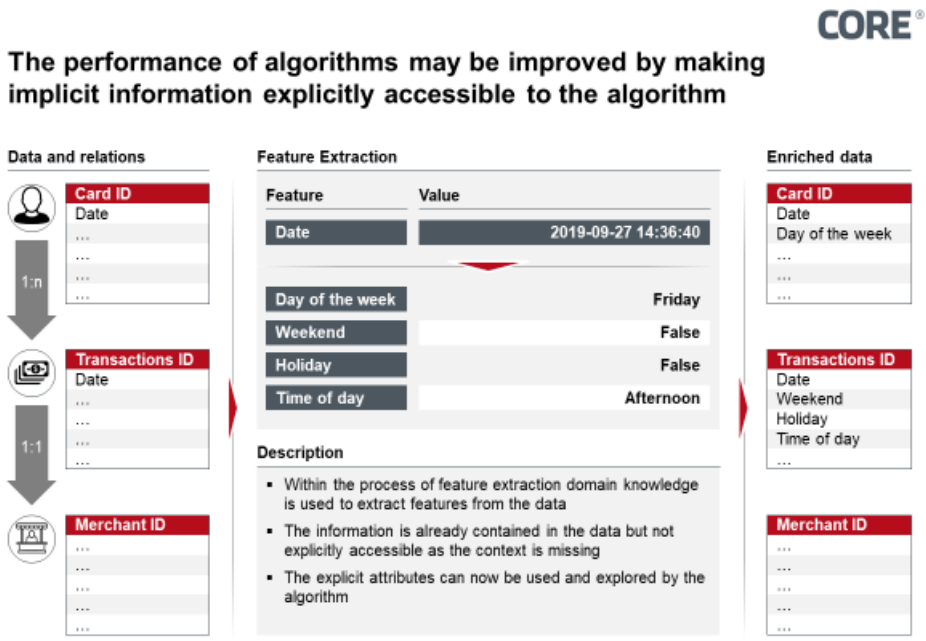

Figure 1: Feature Extraction Prozess in Elo Use Case

The Challenge was initiated by Elo, one of the largest payment transaction providers in Brazil. Elo entered into partnerships with several merchants with the medium-term goal of providing Elo credit card customers with individualized offers. For example, it would be conceivable to offer cardholders some suggestions for a restaurant that are based on their personal preferences and in addition a discount on the menu. In this case, Elo would benefit from transaction fees, among other things.

First, a selection of algorithms are trained to predict consumer willingness and thus customer needs on the basis of the data set. These trained models are then examined more closely against the background of feature engineering in order to enable a discourse on cost and benefit. Various regression models (the variable to be predicted is numerical) are conceivable for forecasting consumer willingness. In the context of this blog post, analyses with two currently very popular models - Gradient Boosting and Deep Learning - will be presented and carried out. For this purpose there will be taken at first a look at the data. The type and content of the data will be briefly explained. In a second step, the process of feature engineering will be presented in a cursory manner. In the following experiment the models Gradient Boosting and Deep Learning with and without Feature Engineering will be trained and validated on the data. In the conclusion,

the results will be reflected and thus a conclusion on quality and properties of the models with regard to the considered use case and feature engineering will be made.

Data

The analysed data set contained data on over 2 million customers, over 300 thousand dealers and almost 30 million transactions between dealers and customers. The data has been anonymized to such an extent that no conclusions can be drawn about individuals or concrete connections, why no statement about specific customer behavior can be made in the conclusion of this blog post. The data set contains a score for each customer that reflects loyalty in the form of the customer's willingness to buy. This customer-specific attribute must be predicted. The score can be seen as representative of other purchase-related attributes (for example, brand loyalty).

Feature Engineering

The process of data preparation includes the step of feature engineering, in addition to data cleansing and modeling, which are necessary for an algorithm to be able to process the data at all. Here, using domain/expert knowledge, information that is already implicit in the data is extracted and explicitly added to the data set in the form of an additional attribute. Examples could be the local arrangement of business areas (for example, area A could be located between B and C) or specific domain knowledge (for example, object A can only be purchased with object B). This information was unfortunately lost due to a strong anonymization. The form of feature engineering discussed in this blog post is the extraction of temporal features from data. Directly intuitive for us humans that the data contains not only a chronological sequence but also information about, for example, the day of the week, the beginning of the month or the working day. Especially for payment processes, additional information can be obtained here (e.g., on Sunday, can be paid less due to shop opening). The data record that contains this information in the form of additional attributes is referred to in the following as an enriched data record.

Experiment/Analysis

The data can now be used to test the different models, training neural networks/deep learning in different depths (2, 3 and 4 layers deep) and widths (layers between 50 and 100 neurons), created with the KERAS framework. For comparison, a model is trained on gradient boosting, an ensemble technique with decision trees, which is currently very popular in the data analyst community. The framework used is LightGBM.

The models are tested on the enriched and unenriched data set. This way, it will be investigated which of the models makes better use of the extracted information and thus can generate further added value. The MSE (Mean squared error - the mean square deviation from the target value) is chosen as a quality measure. Results with a higher MSE have a higher deviation from the expected value and are therefore to be considered worse.

The results of the modelling are shown below. Validation was carried out using 5-fold cross-validation.

Model\Analysis	***Enriched data set***	***Unenriched data set***
Gradient Boosting	3,654782841672	3,698093670456
DL 2-layers	3,857178126259	3,857171424530
DL 3-layers	3,857898095313	3,857640795361
DL 4-layers	3,858468548452	3,858397482541

It should be noted that the gradient boosting approach always provides the better results. In addition, this approach also showed clear advantages in learning and query performance. Furthermore, it can be seen that the performance of the neural network decreases the deeper the network gets. In this context (2 layers deep) there can be hardly spoken about "deep" learning. Finally, it can be observed that the information gained by enriching the data set with the gradient boosting approach can be better used. It is not possible to draw conclusions about the general public of model selection, but this approach seems to be better able to relate the temporal information to the searched information.

Conclusion

As in the previous blog post in which a "Churn Case" was examined with the two models mentioned above, gradient boosting shows strong results compared to deep learning for classification problems that are not about image, speech or handwriting. This has also been shown in the past in internal COREai projects as well as in the community of data analysts. Due to their flexible and potentially powerful architecture, neural networks are able to detect highly complex patterns in very complex data types (video, image or speech), but they do not necessarily take the lead with lower-dimensional data types.

The goal of the blog post was to look at both models against the background of feature engineering and to answer the question whether the involved effort is justified. Besides the fact that feature engineering has been an essential tool of data analysis in the past, allowing the algorithm to use information implicitly contained in the data, the investigation showed that the success of the work step depends also on the algorithm used. On the one hand, it seems to be useful and essential to incorporate domain knowledge into the data analysis from the beginning, even if this requires an often considerable effort on the part of the experts. The image of the lonely data analyst in basement rooms, fully indifferent of the domain he is working with, can therefore not be justified. On the other hand, it is important not to make a decision about the algorithm in the target image without the factor of feature engineering, since an increase in performance may depend on it in principle. As the number of data sources increases, which is not unusual in today's project contexts, the potential for implicitly contained information increases, since the data was most likely not intended for this purpose. The potential to be raised can be found in conscientious feature engineering.

Sources

1. **Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says**
   https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#6ca5c1016f63
2. **Elo Merchant Category Recommendation**
   https://www.kaggle.com/c/elo-merchant-category-recommendation#description)
3. **This is going to be epic... sit back, relax and enjoy!**
   https://www.kaggle.com/raddar/target-true-meaning-revealed
4. **Concepts in predictive machine learning**
   http://www.davidwind.dk/wp-content/uploads/2014/07/main.pdf
5. **Gradient Boosting vs. Deep Learning. Möglichkeiten des Einsatzes Künstlicher Intelligenz im Banking**
   https://core.se/de/techmonitor/gradient-boosting-vs-deep-learning-moeglichkeiten-des-einsatzes-kuenstlicher-intelligenz-im-banking

Christoph Sonn is Engineering Manager at CORE. He has a double degree in economics and chemistry and his professional focus is on the development of innovative products and digitalization solutions for banks, insurance companies and biotechnology companies. From strategic conception to the go-live of innovative IT transformations, Christoph develops solutions for companies to position themselves in the market.

Mail: christoph.sonn@core.se

As Expert Manager, **Julius Heitmann** supports CORE with a focus on the conception of complex IT transformations. In the automotive industry he has gained extensive experience in dealing with Big Data and Data Analytics. He applies this experience in particular in projects for the analysis and optimisation of large data volumes and develops tailor-made control systems for data processing, especially in the financial sector.

Mail: julius.heitmann@core.se

CORE SE
Am Sandwerder 21-23
14109 Berlin | Germany
https://core.se/
Phone: +49 30 263 440 20
office@core.se

COREtransform GmbH
Am Sandwerder 21-23
14109 Berlin | Germany
https://core.se/
Phone: +49 30 263 440 20
office@core.se

COREtransform GmbH
Limmatquai 1
8001 Zürich | Helvetia
https://core.se/
Phone: +41 44 261 0143
office@core.se

COREtransform Ltd.
Canary Wharf, One Canada Square
London E14 5DY | Great Britain
https://core.se/
Phone: +44 20 328 563 61
office@core.se

COREtransform MEA LLC
DIFC – 105, Currency
House, Tower 1
P.O. Box 506656
Dubai I UAE Emirates
https://core.se/
Phone: +97 14 323 0633