Feature Engineering

IMPORTANCE OF FEATURE ENGINEERING TO IMPROVE MODEL PERFORMANCE

STUDENT: PAULO EDUARDO DA SILVA JUNIOR

Summary

1. Why is Feature Engineering Important;
2. Learned vs. Created Characteristics;
3. Common Feature Engineering Operations;
4. Data Leaks and How to Avoid Them;
5. Importance and Generalization of Features;
6. Best Practices for Feature Engineering;
7. Conclusion.

Why is Feature Engineering Important

1. Main reasons: performance impact, practical examples where feature selection outperforms algorithm tuning.

Learned vs. Created Characteristics

1. Deep learning (automatic feature learning);
2. Traditional methods (manual definition);
3. Examples of text processing:
   - Stopword removal
   - Lemmatization
   - Contraction
   - Punctuation
   - Lowercase
   - Tokenization
   - N-gram

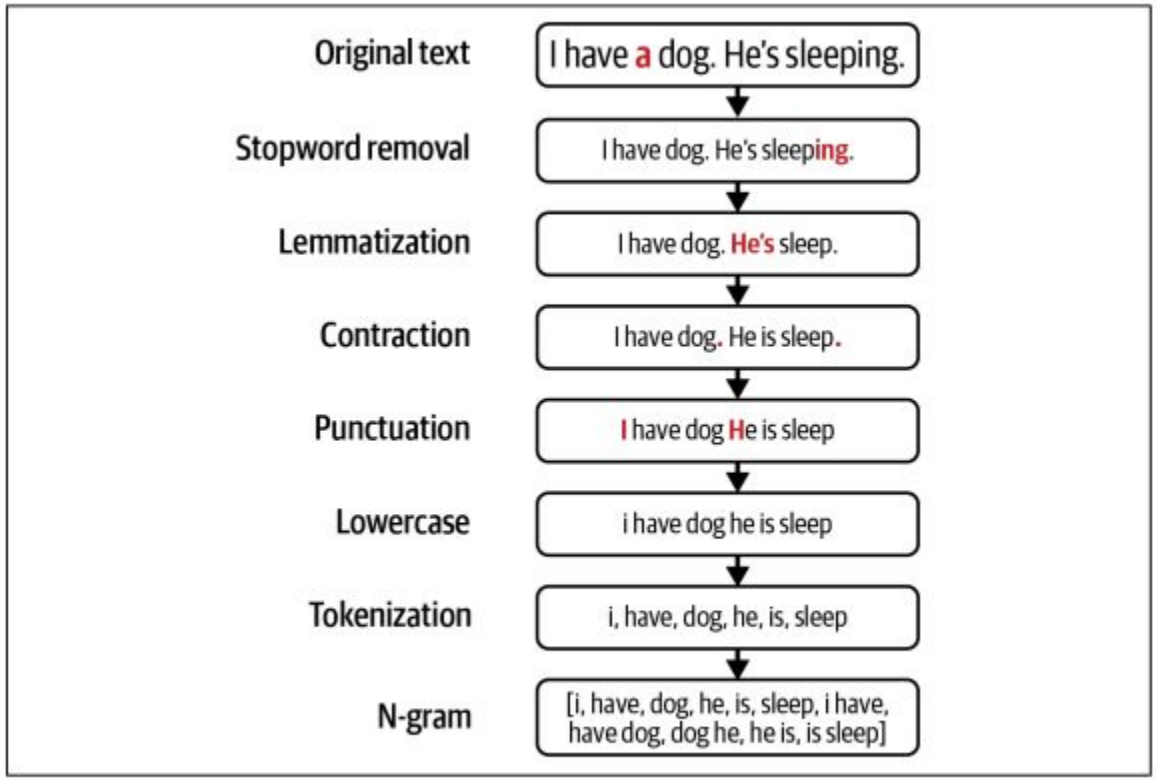

Figure 5-1. An example of techniques that you can use to handcraft n-gram features for your text

Common Feature Engineering Operations

1. Handling missing values;
2. Scaling;
3. Discretization;
4. Encoding of categorical variables;
5. Combining features.

Table 5-2. Example data for predicting house buying in the next 12 months

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

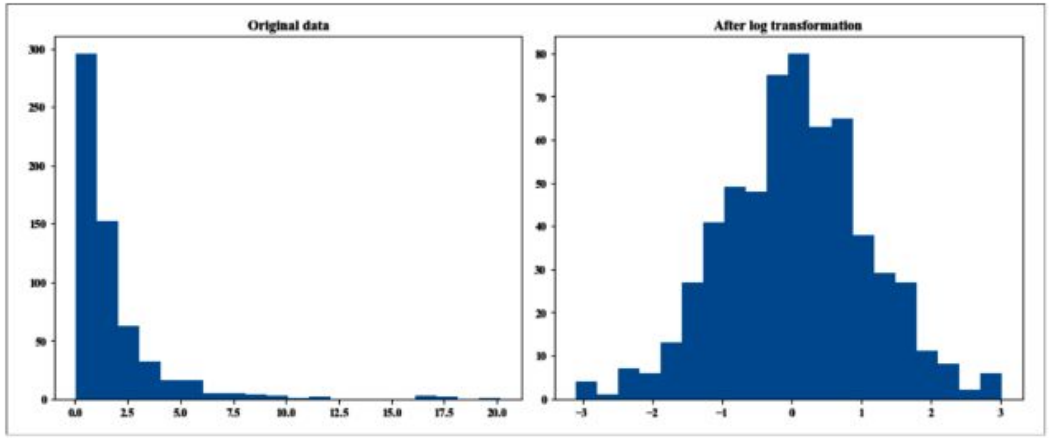

Figure 5-3. In many cases, the log transformation can help reduce the skewness of your data

Data Leaks and How to Avoid Them

1. Leakage occurs when information from the target variable “leaks” into the model during training, leading to incorrect predictions in production;
2. Common causes include:
   - Incorrect splitting of data in time;
   - Scaling before splitting;
   - Data duplication.
3. **Leakage Detection**: Testing the importance of each attribute and performing ablation studies can help identify and prevent leaks.

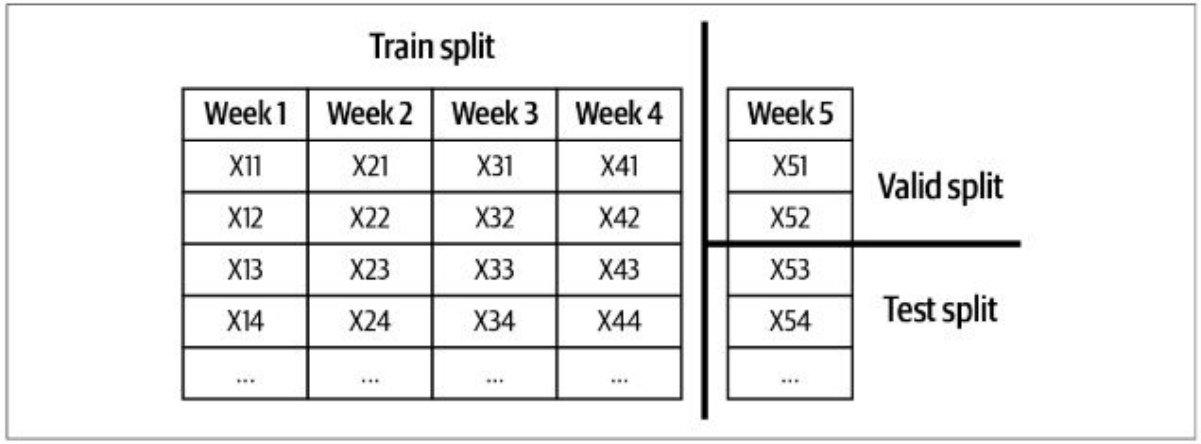

Figure 5-7. Split data by time to prevent future information from leaking into the training process

Importance and Generalization of Features

1. Feature Importance:
   - SHAP;
   - XGBoost;
2. Generalizability.

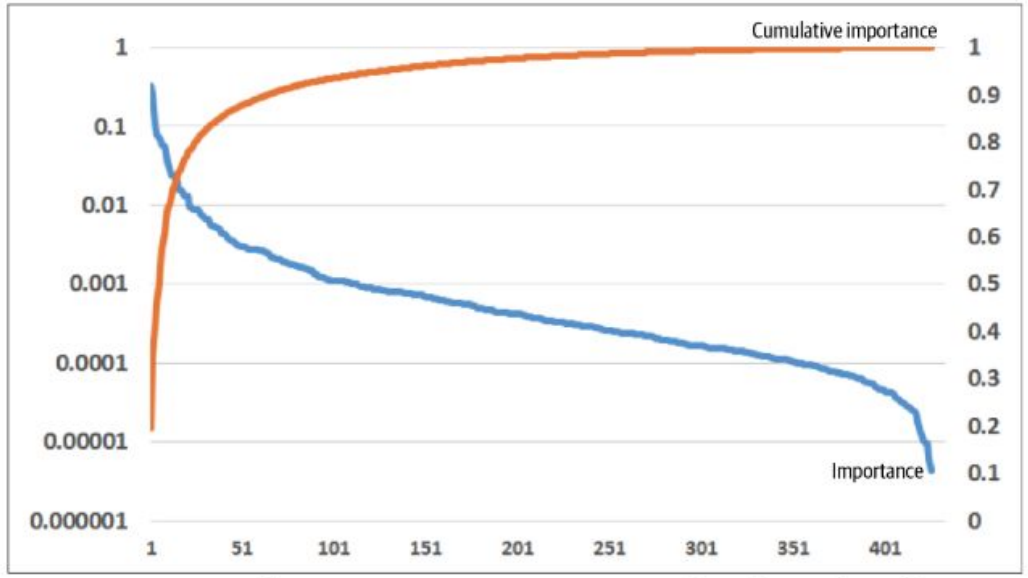

Figure 5-10. Boosting feature importance. X-axis corresponds to the number of features. Feature importance is in log scale. Source: He et al.

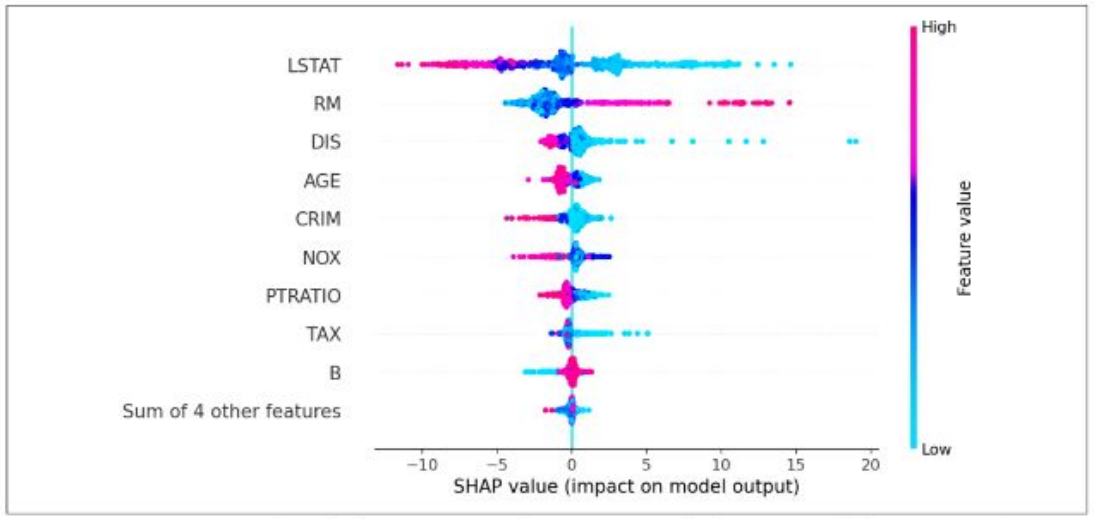

Figure 5-9. How much each feature contributes to a model, measured by SHAP. The feature LSTAT has the highest importance. Source: Scott Lundberg

Best Practices for Feature Engineering

1. These are the best practices listed in the book:
   - Split data by time into train/valid/test splits instead of doing it randomly.
   - If you oversample your data, do it after splitting.
   - Scale and normalize your data after splitting to avoid data leakage.
   - Use statistics from only the train split, instead of the entire data, to scale your
   - features and handle missing values.
   - Understand how your data is generated, collected, and processed. Involve
   - domain experts if possible.
   - Keep track of your data's lineage.
   - Understand feature importance to your model.
   - Use features that generalize well.
   - Remove no longer useful features from your models.

Conclusion

Feature engineering is vital to the success of ML projects;

The best way to learn is through experience: trying out different features and observing how they affect your models' performance.