# Title for title page

Author 1
Author 2
Author 3

Ladislaus von Bortkiewicz Chair of Statistics
C.A.S.E. – Center for Applied Statistics
and Economics
Humboldt–Universität zu Berlin
http://lvb.wiwi.hu-berlin.de
www.case.hu-berlin.de

# Outline

1. Introduction
2. Pre-processing Steps
3. Model Selection
4. Variable Importance and Dimensionality Reduction
5. Results and Conclusion

# Formal Problem Setting

- *training set*: inputs $X = (x_1, \dots, x_n) \in \mathbb{R}^{n \times d}$ and labels $Y = (y_1, \dots, y_n) \in \mathbb{R}^n$
- *test set*: inputs $X' = (x'_1, \dots, x'_t) \in \mathbb{R}^{t \times d}$ without labels

Find a function

$$f : X \to Y \tag{1}$$

s.t. the *test set* labels are predicted as accurately as possible, i.e.

$$f(X') \approx Y' \tag{2}$$

# Outline

1. Introduction ✓
2. Pre-processing Steps
3. Model Selection
4. Variable Importance and Dimensionality Reduction
5. Results and Conclusion

# Pre-processing

Several transformations and cleaning steps needed before putting the data into an algorithm, e.g.

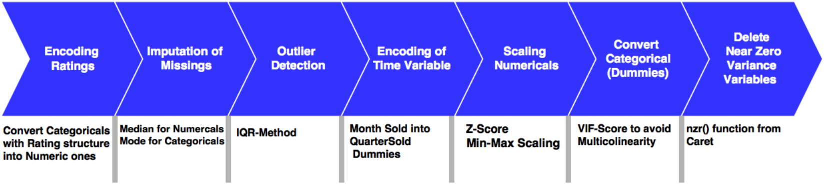


Figure 1: Workflow of Pre-Processing Steps

All transformation need to be preformed on the test set as well!

```
basic_preprocessing = function(X_com, y, scaler="gaussian")
{
    source("replace_ratings.R")
    source("convert_categoricals.R")
    source("impute_data.R")
    source("encode_time_variables.R")
    source("impute_outliers.R")
    source("scale_data.R")
    source("delete_nearzero_variables.R")
    X_ratings = replace_ratings(X_com)
    X_imputed = naive_imputation(X_ratings)
    X_no_outlier = data.frame(lapply(X_imputed, iqr_outlier))
    X_time_encoded = include_quarter_dummies(X_no_outlier)
    X_scaled = scale_data(X_time_encoded, scale_method = scaler)
    X_encoded = data.frame(lapply(X_scaled, cat_to_dummy))
    X_com = delect_nz_variable(X_encoded)
    idx_train = c(1:length(y))
    train = cbind(X_com[idx_train, ]
    test = X_com[-idx_train, ]
    return(list(train = train, X_com = X_com, test = test))
}
```

dataProcessing

# Outline

# Optimizing Hyper-parameters

**Algorithm 1:** t-time k-fold crossvalidation and gridSearch

```
foreach i in 1:t do
    Randomly split the data into k folds of the same size
    foreach j in 1:k do
        Use jth fold as test set and the union of remaining folds as training set
        foreach p in 1:grid do
            Fit model on training set using parameter set p
            Predict on test set and calculate RMSE
        end
    end
    foreach p in 1:grid do
        Calculate average RMSE over the t × k-runs
    end
    choose p with the lowest RMSE
end
```

xgbTuning rfTuning svmTuning

# Taking on the curse of Dimensionality

Problem:

- many variables (99 after pre-processing)
- small training set ($n = 1460$)
- variables are correlated with each other

Our approaches:

- Variable selection through variable importance ranking
- Extract a smaller set of variable using PCA

# Outline

1. Introduction ✓
2. Pre-processing Steps ✓
3. Model Selection ✓
4. Variable Importance and Dimensionality Reduction ✓
5. Results and Conclusion

# Results

- Gaussian SVR with all variable is the single best model
- PCA did not work well
- Models perform best with the full set of variables as Figure **??** suggested

| Inputs | Gaussian SVR | Random Forest | GBM |
|---|---|---|---|
| All Variables | **0.1308** | 0.1484 | 0.1333 |
| Top 30 | 0.1323 | 0.1515 | 0.1436 |
| PCA | 0.1607 | 0.1657 | 0.1657 |

Table 1: RMSE of submitted predictions

Github: finalModels

# Outline

1. Introduction ✓
2. Pre-processing Steps ✓
3. Model Selection ✓
4. Variable Importance and Dimensionality Reduction ✓
5. Results and Conclusion ✓

# References


Breiman, Leo
*"Random Forest." Machine learning, 45(1), 5-32, (1999)*
available on http://machinelearning202.pbworks.com

Chen, Tianqi, and Carlos Guestrin
*"XGBoost: Reliable Large-scale Tree Boosting System", Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining Pages 785-794 (2015)*
available on http://learningsys.org

De Cock, Dean
*"Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project" Journal of Statistics Education 19.3 (2011)*
available on https://ww2.amstat.org

# References


Friedman, Jerome H.
*"Greedy function approximation: a gradient boosting machine." Annals of statistics 1189-1232 (2001).*
available on https://www.jstor.org/journal/annalsstatistics

Kuhn, Max, and Kjell Johnson
*"Applied predictive modeling". New York: Springer (2013)*

Vapnik, Vladimir, Steven E. Golowich, and Alex Smola
*"Support vector method for function approximation, regression estimation, and signal processing." Advances in neural information processing systems 281-287 (1997)*
available on https://semanticscholar.org