Business Analytics

Session 4b. Pre-processing, Feature Engineering, and Variable/Model Selection

Renyu (Philip) Zhang

New York University Shanghai

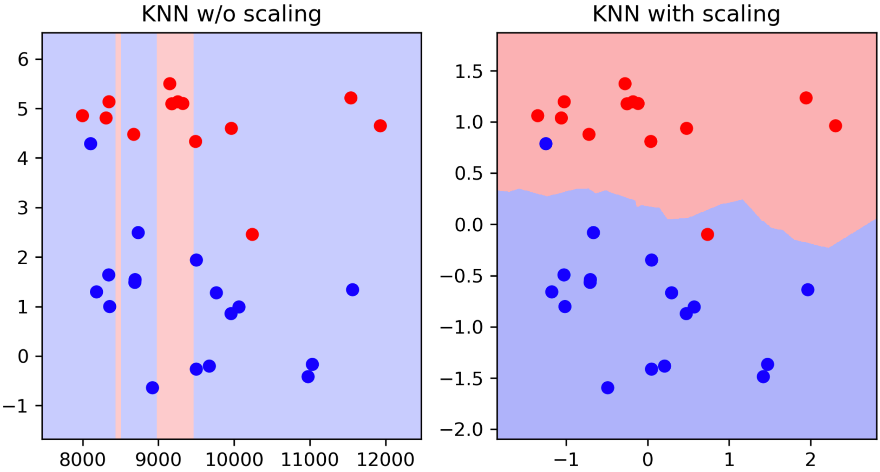
Spring 2019

How Do We Make Better Predictions?

- Understand the business objective.
  - The outcome Y is really of interest to the business.
  - The covariates X have predictive powers on Y.
- Have access to rich and high-quality data.
  - May remove some outliers in the training set.
- "Applied machine learning" is basically feature engineering. — Andrew Ng
- Develop strong predictive models/algorithms.

Scaling

k−Nearest Neighbors with and without Scaling



Scaling

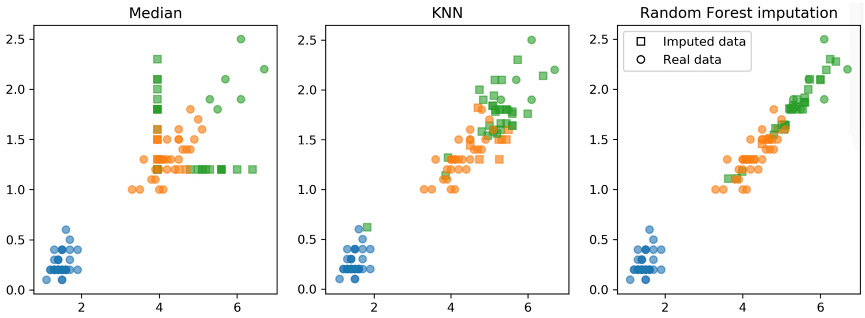
- StandardScaler: Substract mean and devide by standard deviation.
- MinMaxScaler: Substract minimum and devide by the range between maximum and minimum.
- RobustScaler: Substract median and devide by the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile).
  - Not influenced by outliers.
- MaxAbsScaler: Devide by the maximum absolute value.
  - Useful when the data has many zero entries.

Imputation: Dealing with Missing (Covariate) Data

Imputation Methods

- Baseline: Dropping Column
- Mean/Median.
  - Fill in all missing values using the average/median of the covariate in the entire data set.
- k-Nearest Neighbors.
  - Find k nearest neighbors with non-missing values.
  - Fill in all missing values using average of the neighbors.
- Regression models.
  - Train regression model for missing values.
  - Retrain after filling in each missing data.
- Matrix factorization.

Comparison of Imputation Methods



- Mean and median imputation: sklearn.impute.SimpleImputer()
- k-NN and regression imputations have not been included in scikit-learn yet.

Interactions and Polynomial Transformations

Introducing Polynomial Features Systematically

- Systematically generating polynomial and interaction features of degree k: sklearn.preprocessing.PolynomialFeatures(degree=k)

- Can be readily combined with an estimator/model using make_pipeline()

Dealing with Imbalanced Data

Imbalanced Data

- All data are imbalanced.
- If a data set is highly imbalanced, overall accuracy is a misleading measure.
  - In the population, $Y_i = 0$ for 99% of the cases and $Y_i = 1$ for 1% of the cases. The overall accuracy will be as high as 99% if we ignore covariates and blindly predict $Y_i = 0$.
- Addressing data imbalancednes: Resampling
  - Imbalanced-Learn package in Python
  - pip install -U imbalanced-learn

Random Undersampling and Oversampling

- Undersampling: Randomly undersample the training data whose outcome is over-represented.

- Oversampling: Randomly oversample the training data whose outcome is under-represented.

Random Undersampling and Oversampling

- Undersampling: Randomly undersample the training data whose outcome is over-represented.
- Oversampling: Randomly oversample the training data whose outcome is under-represented.

Synthetic Minority Oversampling Technique (SMOTE)

- Adds synthetic interpolated data to minority classes.
  - Kind of oversampling.
- For each sample in minority classes:
  - Pick a random neighbor from k nearest neighbors.
  - Pick a point on the line connecting the two (uniformly random).
  - Repeat until the training set is sufficiently balanced.
- Works very well in practice.

Illustration of SMOTE

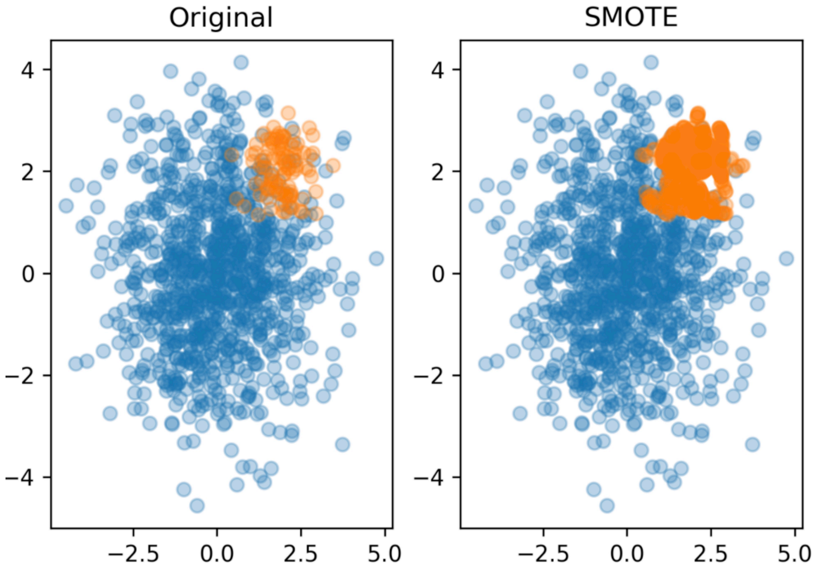
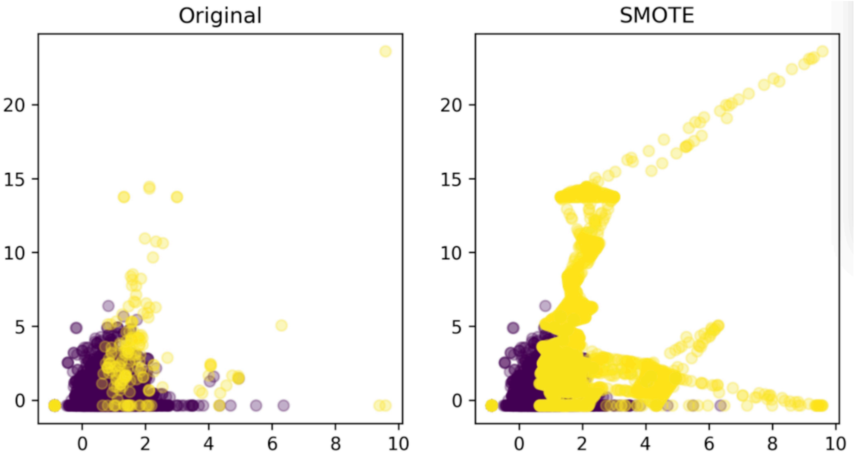


Illustration of SMOTE



Homework

- Finish Homework 4 (NO need to submit it).

- Read "The Analytics Edge", Chapters 8.