Classification – Prediction

So far: mainly ’classification’

- Naive Bayes
- k-Nearest Neighbours
- . . .

Based on predictor variables $X_1, X_2, \ldots, X_p$
Trying to determine class Y (= discrete).

- Training data: to build a model
- Validation data: to test the accuracy of the model

→ confusion matrix

Now: also **’prediction’**

- k-Nearest Neighbours
- Multiple Linear Regression
- . . .

Based on predictor variables $X_1, X_2, \ldots, X_p$ trying to predict the **value of the continuous variable** Y.

- Training data: to build a model

- Validation data: to test the accuracy of the model!

  →**numeric measures**

For each observation i, prediction error (residue):

$$e_i = y_i - \hat{y}_i$$

with

- y_i: the ’real’ value
- $\hat{y}_i$: the value predicted (by the model)

Numeric measures for the accuracy of the prediction model

- MAE/MAED (Mean Absolute Error/Deviation)

…

- Average Error

…

- MAPE (Mean Absolute Percentage Error)

…

- RMSE (Root Mean Squared Error)

…

- TSSE Total Sum of Squared Errors

…

Simple Linear Regression

Based on 1 predictor variable X, trying to predict the value of 1 continuous output variable Y.

Theoretical model (population):

$$Y = \beta_0 + \beta_1 \cdot X + \varepsilon$$

with ε ”noise”, dispersion in Y.

Assumptions:

- dispersion in Y the same for every value of X

  ($\varepsilon \sim N(0, \sigma^2)$)

  = ’homoscedasticity’

- prediction errors (residues) independent

Training data (e.g. 60% entire dataset)

- estimate coefficients β_0 en β_1

  → $Y = b_0 + b_1 \cdot X$
- parameter ε, estimate dispersion in Y

  → ’Std. Dev. estimate’ in output

How good is the found model?

→ depends on the purpose!

• Purpose analyse: description (typical statistics)

→ ’goodness of fit’

→ calculations on training data!

→ calculating R^2 (or R)

→ the closer to 1 (or -1), the better

→ R-squared in output

• Purpose analyse: predicting (typical data mining)

→ numeric measures to predict accuracy (see previous slide)

→ calculations on validation data!

→ Validation Data scoring in output

➡ finding balance between both

→ for data mining: mainly predictive power important

What in case of a different partition?

- for example ratio 60%-40%, but different seed

- for example different ratio, but same seed

➡ different estimates b0 and b1 for β_0 and $\beta_\&$

→ to what extend can this be different from one partition to another?

→ estimates for the dispersion of the parameters from one partition to another?

→ ’Std. Error’ in output with ’Coefficient’

Multiple Linear Regression

Based on multiple predictor variables $X_1, X_2, \ldots, X_p$ trying to predict the value of 1 continuous output variable Y.

Theoretical model (population):

$$Y = \beta_0 + \beta_1 \cdot X_1 + \ldots + \beta_p \cdot X_p + \varepsilon$$

with ε ”noise”, dispersion in Y.

Assumptions:

- dispersion in Y same for each value of X
  ($\varepsilon \sim N(0, \sigma^2)$)
  = ’homoscedasticity’

- prediction errors (residues) independent

Training data (e.g. 60% entire dataset)

- estimating parameters $\beta_0, \beta_1, \ldots, \beta_p$
  $\rightarrow Y = b_0 + b_1 \cdot X_1 + \ldots + b_p \cdot X_p$
- parameter ε, estimating dispersion in Y
  → ’Std. Dev. estimate’ in output

What predictor variables contribute ‘effectively’ to the prediction?

➡ What coefficients b_0, b_1, . . . b_p are significantly different from 0?

Coefficients in the model being different from zero can be coincidence!
(for example by observations (by chance) in the training data)

➡ calculating p-value

• p-value: probability to find this value by chance for a coefficient with the model in the training data if that coefficient in the population equals zero.

➡ Low p-value: coefficient significantly different from zero, contributes ‘effectively’ to the prediction.

• ’p-value’ in output with ’Coefficient’

Optimal number + choice of predictor variables?

• Too many variables: possibility of overfitting!
➡ perhaps low predictive power

• Preferably don’t take in variables that don’t contribute to the prediction.
➡ leads to larger dispersion in the predictions

• Preference for not removing variables that ‘effectively’ contribute to the prediction.
➡ leads to a higher average error in the predictions

• Beware of predictor variables that are strongly correlated!
→ can falsely represent coefficients
→ track down correlations (’matrix plot’ or ’correlation matrix’)

• Be aware of outliers!

• Rule of thumb: number of observations *n* in training data equals at least 5·(p+2)

Methods to choose the best subset of predictor variables

- first: reduce number of predictor variables by means of domain knowledge

- then: use algorithms

  – ’Exhaustive search’: try all predictor variables subsets

  – ’Forward selection’: start with 1 predictor variable, add each time the most significant one

  – ’Backward selection’: start with all predictor variables, remove each time the least significant one

  – . . .