EDA Recap

Dr. Mohamed Elshenawy
mmelshenawy@gmail.com

1

In Previous Lectures

1. Linear Regression
2. Polynomial Regression
3. Training and test error
4. Bias and Variance Tradeoff
5. Assessment of Classification and Regression Models
6. KNN
7. Logistic Regression
8. SVM
9. Decision trees
10. Ensemble Learning
    1. Bagging
    2. Boosting

2

In this session

- Other ensemble techniques that you need to know.
- Practice on Ensemble Learning

3

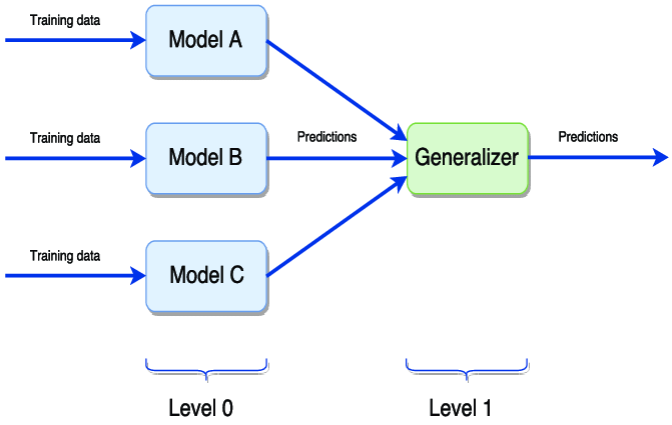
Algorithms you need to know

- Bagging
  - Random Forest
- Boosting techniques
  - AdaBoost (Weak models are added sequentially, trained using the weighted training data) – not covered in this course
  - Gradient Boosting (create ensemble in a way that optimizes a differentiable loss function (in a way similar to gradient descent) – not covered in this course
- Stacking
  - We will discuss the main idea
- All the above techniques can be use as classifiers or regressors

4

Stacking

- The architecture of a stacking model involves two or more base models, often referred to as level-0 models, and a meta-model that combines the predictions of the base models, referred to as a level-1 model.



Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting

5

EDA Recap

6

When starting my EDA

- Steps to consider:
  - Quantify missing data
  - Identify numerical and categorical variables
  - Determine unique values (cardinality) in categorical variables
  - Check rare/ dominant categories in categorical variables
  - Highlight outliers
- Other useful steps
  - Identify linear relationships
  - Identify a normal distribution
  - Check histograms

7

Imputing missing data

- Before you impute
  - Calculate percentage of missing values for each variable and sort them.
  - Consider dropping columns with high percentage of missing values.
- Decided to impute??
  - You need to understand the data first and decide on the best strategy. Imputation may change the distribution of your data.
  - Consider mean or median imputation for numerical features.
  - Consider mode or frequent category for categorical features.
  - Replace with an arbitrary value (-1 for positive features)
  - Replace with 'Missing' for categorical features

8

Other imputation strategies

- IterativeImputer
  - Multivariate imputer that estimates each feature from all the others. It models each feature with missing values as a function of other features in a round-robin fashion.
- MissingIndicator
  - Adds a binary indicator that specifies whether a value was missing (1) or not (0). You may replace the value in the original feature using mean, median, or mode while flagging those missing observations with a missing indicator.
- KNNImputer
  - Completes missing values using k-Nearest Neighbors.

9

Encoding Categorical Variables

- Different techniques
  - Creating binary variables through one-hot encoding (OneHotEncoder)
  - Performing one-hot encoding of frequent categories
  - Replace categories with ordinal numbers (OrdinalEncoder(input), LabelEncoder(traget))
  - Encoding with integers in an ordered manner.
- Consider
  - Grouping categories that have similar meaning
  - Grouping rare or infrequent categories.

10

Performing Feature Scaling

- Standardizing the features (StandardScaler)

$$z = \frac{x - mean(x)}{std(x)}$$

- Scaling to the maximum and minimum values (MinMaxScaler)

$$z = \frac{x - min(x)}{max(x) - \min(x)}$$

- Scaling with the median and quantiles (RobustScaler) – not influenced by a few number of large marginal outliers

$$z = \frac{x - median(x)}{75thQuantile(x) - 25\text{th}Quantile(x)}$$

11

Performing Feature Scaling (cont.)

- Scaling to vector unit length (Normalizer)
- Each sample (i.e. each row of the data matrix) with at least one non zero component is rescaled independently of other samples so that its norm (l1, l2 or inf) equals one.
- Unlike the other scalers which work on the individual column values, the Normalizer works on the rows.
- Scaling inputs to unit norms is a common operation for text classification or clustering.

12

Other transformations

- Polynomial transformation
- Log Transformation: used to convert a skewed distribution to a normal distribution/less-skewed distribution.
  - Remember
    - log(10) = 1
    - log(100) = 2
    - log(10000) = 4.
- Note: if our data has negative values or values ranging from 0 to 1, we cannot apply log transform directly. such cases, we can add a number to these values to make them all greater than 1. Then, we can apply the log transform.

13

You need to work on the following

- **Credit card fraud**
  - https://www.kaggle.com/mlg-ulb/creditcardfraud
- **Bike Sharing Demand**
  - https://www.kaggle.com/c/bike-sharing-demand

- Quick and dirty submission: finish the two models before 3:30
- Final submission: You have till Friday

14