Machine Learning with Python

Session 2: End-to-end Machine Learning Project

Arghya Ray

Main steps you need to go through:

1. Look at the big picture (what is the objective, frame the problem, what type of algorithm to use, performance measures)
2. Get the data (get a quick view of data using head(), info(), value_counts(), describe(),etc.)
3. Discover and visualize the data to gain insights (generalization error→ data snooping bias; Finding correlations)
4. Prepare the data for Machine Learning Algorithms (data cleansing, handling text and categorical attributes, custom transformers, feature scaling→Min-max scaling, standardization)
5. Select a model and train it (Split into training and testing sets, training and evaluating, Better evaluation using cross-validation)
6. Fine tune your model (Grid-search, Randomized search, Ensemble methods)
7. Present your solution (analyze the best models and their errors)
8. Launch, Monitor and Maintain your system

Data Collection and Pre-processing:

- Improving the quality of data in databases for use in data-mining is a challenging task. The presence of incorrect and inconsistent data can significantly impact the result of data mining analysis and therefore potential benefits of using data-mining may not be achieved.
- Usually data required for data mining tasks needs to be extracted from a number of databases, integrated and perhaps cleansed and transformed. This process is called ***ETL (Extraction, Transformation and Loading)***.
- **Data Cleansing** is a process used to determine inaccurate, incomplete or unreasonable data items of a dataset and then improving the data quality through corrections of the detected errors and omissions.

- <u>**Sources of errors in the data:**</u>
  - **Instance Identity Errors**: Same individual may be represented slightly differently in different source systems.
  - **Data Errors**: Deals with missing attribute values, duplicate records, wrong aggregations, non-unique identifiers, inconsistent use of nulls spaces and empty spaces, coding mismatch across databases, inappropriate use of address lines, etc.
  - **Record Linkage Problem**: The problem of linking information from different databases that relates to the same customer or client.
  - **Semantic Integration Problem**: Deals with errors that arise during integration of information found in different sources.
  - **Data Integrity Problem**: Data integrity deals with issues like referential integrity, null values, domain of values, etc.
  - **Data Entry Errors**: Due to unmotivated data entry staff.
  - **Measurement Errors:** Errors creep in because of instrument malfunctioning, poor calibration, or poor design of s/w used in instrument.
  - **Filtering Errors**: Each step of filtering, smoothing, and summarization of data is prone to produce errors.

Detecting Outliers:

- An **outlier** is an observation that is “extreme”, being distant from the rest of the data (definition of “distant” is deliberately vague)
- Different data mining software appear to include different criteria for identifying outliers.
- Outliers can have disproportionate influence on models. Detecting outliers is an important step in data pre-processing.
- Once detected, domain knowledge is required to determine if it is an error, or truly extreme.
- Even though it is often thought that outliers should be quickly eliminated, but outliers can contain useful information. Some cases:
  - In a dataset about number of visas or passports issued by different offices or branches in a country, an outlier may show that too many visas or passports were issued by one agency or branch.
  - In a dataset of expenditure incurred by each branch of a company, many overseas trips funded by one overseas branch of a MNC.
  - In a computer system that has software that monitors behaviour of its users, a user’s behaviour may be found to be different than what is normally expected. This user may be flagged. Such an approach is used in what is called ***anomaly detection***.
  - Finding outliers is the purpose of the DM exercise (airport security screening). This is called “anomaly detection”.
- Outliers may be of different types: **Univariate**, **Multivariate**, or **Time-series**.
- Some classify outliers are:
  - **Global Outliers**: When an outlier is significantly different from the rest of the data-points.
  - **Contextual Outliers**: When an outlier is significantly different from the rest of the data-points in the same context.
  - **Collective Outliers**: When a number of outliers are significantly different from the rest of the dataset.

Mining Outliers:

- **Mining Univariate Outliers:** A single dimension variable. Robust statistics to detect outliers: $(\mu - 3\sigma, \mu + 3\sigma)$
- **Mining Multivariate Outliers**: A multivariate dataset is a set of vectors, each data point being a vector. It is sometimes necessary to consider a number of attributes together like, population and population growth. Mean value and s.d. of the pair (x,y)
- **Distance based outliers**: In the discussion of outliers above, we have assumed that variables are normally distributed. In case the normality assumption is not true, a non-parametric model free approach is adopted that involves the pair wise distances.
- **Mining Time-series Outliers**: Time series data are mainly used for identifying seasonality, trend, etc. One technique is to use Mean absolute deviation (MAD).
- **Other Techniques**:
  - Some methods are based on classification methods- ***Supervised classification*** *and* ***Unsupervised Classification.***
  - Some outlier detection methods use **statistical tests** (Grubb's test) while others may use **distance-based approach** (Euclidian distance).
  - Outliers in some cases may be identified by examination of **unique rules** (Each value of the given attribute must be different from all other values of the attribute), **consecutive rules** (There can be no missing values between the lowest and highest values for the attribute and that all values must also be unique. E.g., as in check numbers), and **null rules** (Specifies the use of blanks, question marks, special characters or other strings that may indicate the null condition).
  - A common outlier detection method is the use of good data visualization software (histogram, box-plot, etc.).

Further Reading: https://towardsdatascience.com/assessing-the-quality-of-data-e5e996a1681b

Handling Missing Data:

- There can be a number of reasons for missing values including:
    - The particular data has no value associated with it.
    - The field was not applicable, the event did not happen, or the data was not available.
    - The person who entered the data did not know the right value or did not care if the value is filled in.
    - The value is to be provided by a later step of the process.
- Most algorithms will not process records with missing values. Default is to drop those records.
- **Solution 1: Omission**
    - If a small number of records have missing values, can omit them
    - If many records are missing values on a small set of variables, can drop those variables (or use proxies)
    - If many records have missing values, omission is not practical
- **Solution 2: Imputation**
    - Replace missing values with reasonable substitutes
    - Lets you keep the record and use the rest of its (non-missing) information

Normalizing (Standardizing) Data:

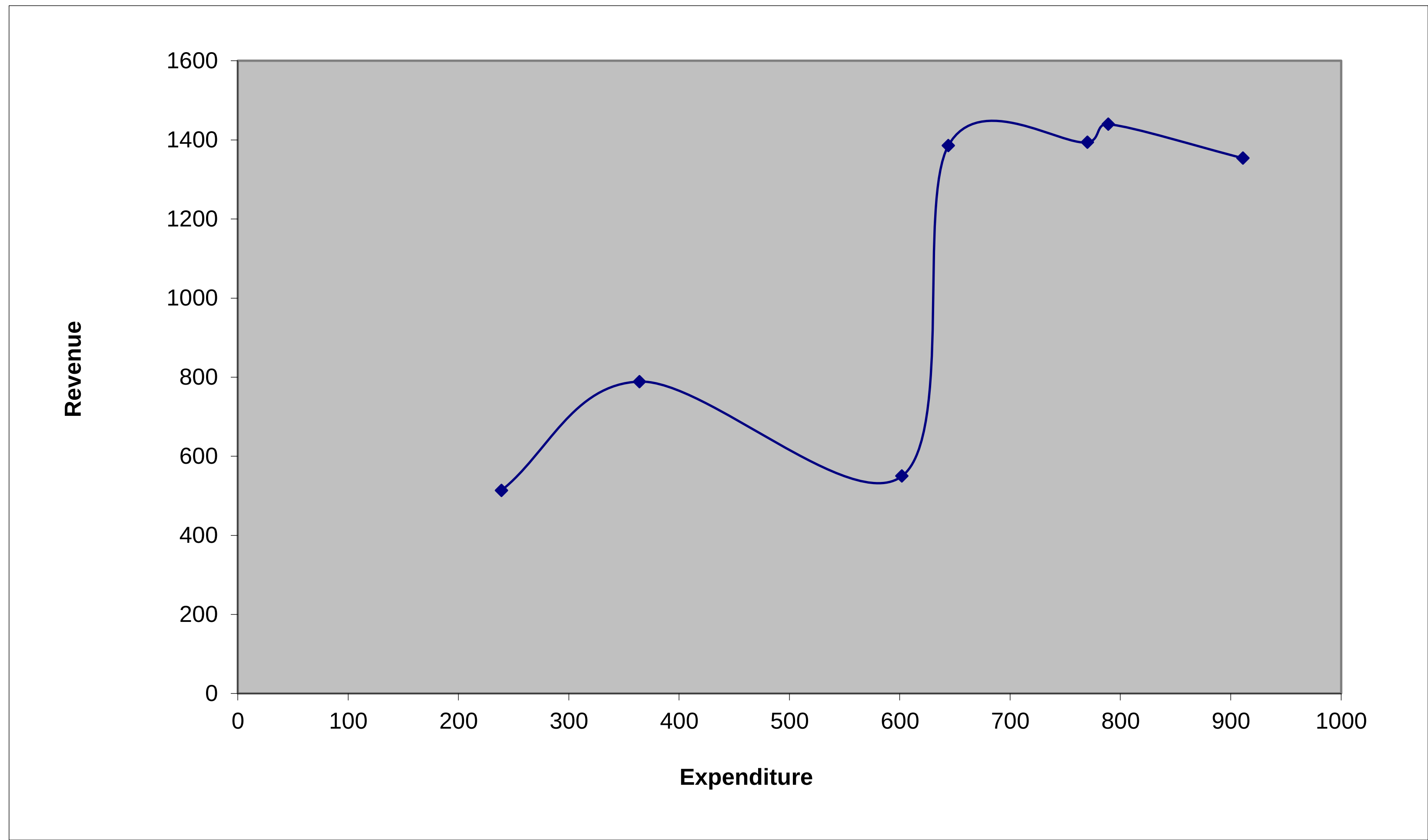
- Used in some techniques when variables with the largest scales would dominate and skew results
- Puts all variables on same scale
- Normalizing function: Subtract mean and divide by standard deviation (used in XLMiner)
- Alternative function: scale to 0-1 by subtracting minimum and dividing by the range

Rare event oversampling

- Often the event of interest is rare. Examples: response to mailing, fraud in taxes, etc.
- Sampling may yield too few "interesting" cases to effectively train a model
- Popular solution: oversample the rare cases to obtain a more balanced training set. Later, need to adjust results for oversampling.

The Problem of Over-fitting

- Statistical models can produce highly complex explanations of relationships between variables.
- The "fit" may be excellent. But when used with new data, models of great complexity do not do so well.
- Causes:
  - Too many predictors
  - A model with too many parameters
  - Trying many different models
- Consequence: Deployed model will not work as expected with completely new data.
- To handle the problem of over-fitting, we need to go for validation and testing.



Partitioning the Data:

Problem: How well will our model perform with new data?

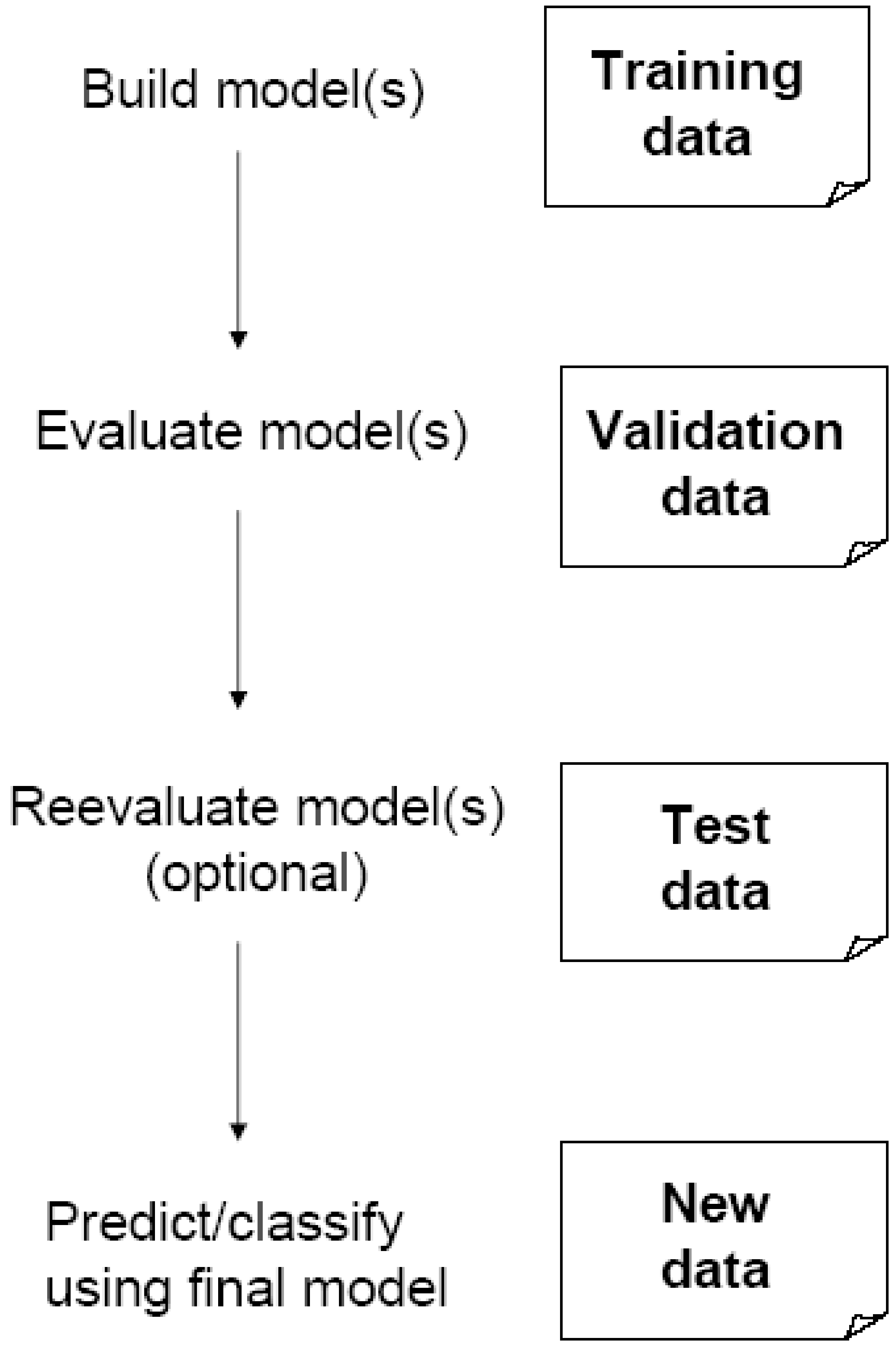
Solution: Separate data into two parts.

- Training partition to develop the model
- Validation partition to implement the model and evaluate its performance on "new" data.

Test Partition

- When a model is developed on training data, it can overfit the training data (hence need to assess on validation)
- Assessing multiple models on same validation data can overfit validation data.
- Some methods use the validation data to choose a parameter. This too can lead to overfitting the validation data .

Solution: final selected model is applied to a test partition to give unbiased estimate of its performance on new data



The content of the slides are prepared from different textbooks.

References:

- Links:
  - https://www.sas.com/en_in/insights/big-data/what-is-big-data.html
  - https://www.oracle.com/big-data/what-is-big-data/
  - https://www.w3schools.com/python/python_variables_multiple.asp

- Predictive Analytics for Dummies, By Anasse Bari, Mohamed Chaouchi, & Tommy Jung, Copyright 2016, John Wiley & Sons, Inc.
- Introduction to Data Mining with Case Studies, By G.K. Gupta. Copyright 2014 by PHI Learning Private Limited.


Thank you..