JBI030: Data Mining

End-to-End Data Mining Project Checklist

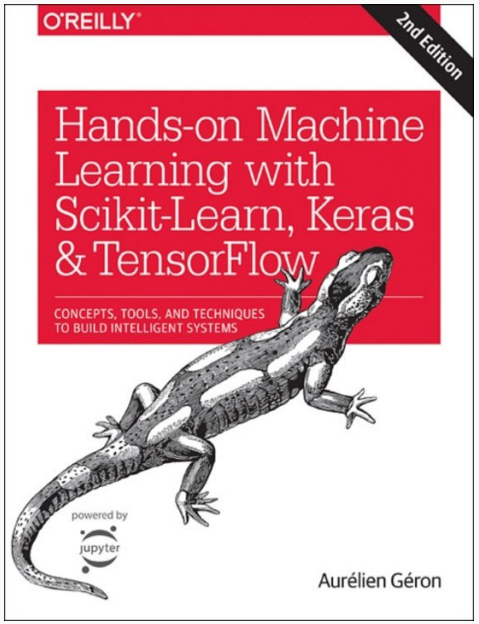
Davide Vidotto

Data Science (TuE/UvT)

In This Lecture

1. Frame the Problem
2. Get the Data
3. Data Exploration
4. Preprocessing
5. Shortlist of Promising Models
6. Fine-Tune the Algorithms
7. Communicate and Deploy Your Solution

This lectures presents an adapted version of Appendix B ("Machine Learnign Project Checklist") of the book "Hands-On Machine Learning with Scikit-Learn and TensorFlow" by Aurélien Géron.



Frame the Problem

Frame the Problem

- Define the objective in business term
- How the solution will be used?
- What type of data do you need? How much?
- What's the problem setting? (Supervised/Unsupervised Learning)
- Performance:
  - What type of metric do I need to improve to achieve my goal?
  - What the minimum performance to reach the objective?
- Can I count on the aid of domain experts?

Get the Data

Get the Data

- Collect the data (e.g., extract them from a **DataBase**)
- Convert them in such a way that you can manipulate them
- Ensure sensitive information is deleted or protected
- Check data size and type
- **Sample a test set, and never look at it** (no data snooping)

Data Exploration

Data Exploration (1/2)

Note: at this point, insights from domain experts can better address your steps.

- Create a copy of your data (sample it down to a manageable size if necessary)
- Keep track of the steps you undertake for exploration
- Study each attribute and its characteristics:
  - name
  - type (continuous, categorical,...)
  - % missingness
  - distribution
  - usefulness
  - ...

Data Exploration (2/2)

- If Supervised Learning: identify the target(s)
- Check data correlations
- Plot and visualize the data
- Identify (with domain experts) possible data transformations

Preprocessing

Preprocessing (1/2)

- Use a copy of the data
- Detect useful preprocessing steps for your data:
  - Data Cleaning (outliers-optional, missing values)
  - Feature Selection (drop features that do not provide useful information for the task)
  - Feature Engineering (discretization, encoding, transformations such as polynomial, logarithmic, aggregate or combine features, ...) - this step is better done with domain experts
  - Scaling

Preprocessing (2/2)

- Write and save the funtions you use for your preprocessing step:
  - You can **reuse** them also for other data
  - You can apply these transformations to the test set
  - To make it easy to treat the preprocessing methods as hyperparameters for model tuning (e.g., in cross-validation)

Shortlist of Promising Models

Shortlist of Promising Models (1/2)

Note: If the data is huge, you may want to sample smaller training sets so you can train many different models in a reasonable time (be aware that this penalizes complex models such as large neural nets or Random Forests).

1. Try many quick-and-dirty models from different categories (linear models, Naive Bayes, KNN, Neural Nets, Trees, etc.)
2. Measure and compare their performance (using for example a cross-validation metric)
3. for each model, analyze the most significant features
4. Analyze the errors each algorithm makes
5. Optional: Refine the features giving the results of previous steps (feature engineering, feature selection, ...)

Shortlist of Promising Models (2/2)

- Re-perform steps (1)-(5) above a couple of times (also by varying some hyperparameters, if you desire; you don't need to tune the model now, but just to check how they perform when hyperparameters take on different values or orders of magnitude)
- List the top three to five most promising models: these will be candidates for the next steps

Fine-Tune the Algorithms

Fine-Tune the Algorithms (1/2)

Note: In this step, you would want to use as much of the original dataset (not its copy!) as possible.

- Use **cross-validation** (or holdout if you have enough data):
  - treat the data preprocessing methods you have chosen earlier as hyperparameters to avoid data leakage (For example: should I impute the data with the mean or the median? Or can I just drop the rows with missing values?)
  - Unless you have very few hyperparameters to tune, prefer Random Search over Grid Search

Fine-Tune the Algorithms (2/2)

Note: In this step, you would want to use as much of the original dataset (not its copy!) as possible.

- Try **ensembles** (combine trees, or the best models you have found)
- Once you are confident about your best model, measure its performance on the **test set** to estimate the generalization performance

Warning

Don't tune the model after measuring generalization performance, or you would end up overfitting the test set.

Communicate and Deploy Your Solution

Communicate Your Results

- Document your steps
- Create a nice presentation
- Explain how the solution achieves the business objective
- Present also assumptions and limitations of the solutions
- Highlight key points ("feature X_j is the number one predictor of Y")

Deploy Your Model

- Get your solution ready for production (plug into production data inputs, etc.)
- Write monitoring code to assess the model's live performance at regular intervals
- Monitor input's quality
- Retrain your model on a regular basis on fresh data
- Alternatively, use your model's results to go deeper into the business objective, the data you collected, and so on (restart the cycle), or use it to come up with new business goals