Chapter 18

Predictive Modeling Project Template

Applied machine learning is an empirical skill. You cannot get better at it by reading books and articles. You have to practice. In this lesson you will discover the simple six-step machine learning project template that you can use to jump-start your project in Python. After completing this lesson you will know:

1. How to structure an end-to-end predictive modeling project.
2. How to map the tasks you learned about in Part II onto a project.
3. How to best use the structured project template to ensure an accurate result for your dataset.

Let's get started.

18.1 Practice Machine Learning With Projects

Working through machine learning problems from end-to-end is critically important. You can read about machine learning. You can also try out small one-off recipes. But applied machine learning will not come alive for you until you work through a dataset from beginning to end.

Working through a project forces you to think about how the model will be used, to challenge your assumptions and to get good at all parts of a project, not just your favorite parts. The best way to practice predictive modeling machine learning projects is to use standardized datasets from the UCI Machine Learning Repository. Once you have a practice dataset and a bunch of Python recipes, how do you put it all together and work through the problem end-to-end?

18.1.1 Use A Structured Step-By-Step Process

Any predictive modeling machine learning project can be broken down into six common tasks:

1. Define Problem.
2. Summarize Data.
3. Prepare Data.

4. Evaluate Algorithms.
5. Improve Results.
6. Present Results.

Tasks can be combined or broken down further, but this is the general structure. To work through predictive modeling machine learning problems in Python, you need to map Python onto this process. The tasks may need to be adapted or renamed slightly to suit the Python way of doing things (e.g. Pandas for data loading and scikit-learn for modeling). The next section provides exactly this mapping and elaborates each task and the types of sub-tasks and libraries that you can use.

18.2 Machine Learning Project Template in Python

This section presents a project template that you can use to work through machine learning problems in Python end-to-end.

18.2.1 Template Summary

Below is the project template that you can use in your machine learning projects in Python.

```
# Python Project Template

# 1. Prepare Problem
# a) Load libraries
# b) Load dataset

# 2. Summarize Data
# a) Descriptive statistics
# b) Data visualizations

# 3. Prepare Data
# a) Data Cleaning
# b) Feature Selection
# c) Data Transforms

# 4. Evaluate Algorithms
# a) Split-out validation dataset
# b) Test options and evaluation metric
# c) Spot Check Algorithms
# d) Compare Algorithms

# 5. Improve Accuracy
# a) Algorithm Tuning
# b) Ensembles

# 6. Finalize Model
# a) Predictions on validation dataset
# b) Create standalone model on entire training dataset
# c) Save model for later use
```

Listing 18.1: Predictive modeling machine learning project template.

18.2.2 How To Use The Project Template

1. Create a new file for your project (e.g. `project_name.py`).
2. Copy the project template.
3. Paste it into your empty project file.
4. Start to fill it in, using recipes from this book and others.

18.3 Machine Learning Project Template Steps

This section gives you additional details on each of the steps of the template.

18.3.1 Prepare Problem

This step is about loading everything you need to start working on your problem. This includes:

- Python modules, classes and functions that you intend to use.
- Loading your dataset from CSV.

This is also the home of any global configuration you might need to do. It is also the place where you might need to make a reduced sample of your dataset if it is too large to work with. Ideally, your dataset should be small enough to build a model or create a visualization within a minute, ideally 30 seconds. You can always scale up well performing models later.

18.3.2 Summarize Data

This step is about better understanding the data that you have available. This includes understanding your data using:

- Descriptive statistics such as summaries.
- Data visualizations such as plots with Matplotlib, ideally using convenience functions from Pandas.

Take your time and use the results to prompt a lot of questions, assumptions and hypotheses that you can investigate later with specialized models.

18.3.3 Prepare Data

This step is about preparing the data in such a way that it best exposes the structure of the problem and the relationships between your input attributes with the output variable. This includes tasks such as:

- Cleaning data by removing duplicates, marking missing values and even imputing missing values.

- Feature selection where redundant features may be removed and new features developed.
- Data transforms where attributes are scaled or redistributed in order to best expose the structure of the problem later to learning algorithms.

Start simple. Revisit this step often and cycle with the next step until you converge on a subset of algorithms and a presentation of the data that results in accurate or accurate-enough models to proceed.

18.3.4 Evaluate Algorithms

This step is about finding a subset of machine learning algorithms that are good at exploiting the structure of your data (e.g. have better than average skill). This involves steps such as:

- Separating out a validation dataset to use for later confirmation of the skill of your developed model.
- Defining test options using scikit-learn such as cross validation and the evaluation metric to use.
- Spot-checking a suite of linear and nonlinear machine learning algorithms.
- Comparing the estimated accuracy of algorithms.

On a given problem you will likely spend most of your time on this and the previous step until you converge on a set of 3-to-5 well performing machine learning algorithms.

18.3.5 Improve Accuracy

Once you have a shortlist of machine learning algorithms, you need to get the most out of them. There are two different ways to improve the accuracy of your models:

- Search for a combination of parameters for each algorithm using scikit-learn that yields the best results.
- Combine the prediction of multiple models into an ensemble prediction using ensemble techniques.

The line between this and the previous step can blur when a project becomes concrete. There may be a little algorithm tuning in the previous step. And in the case of ensembles, you may bring more than a shortlist of algorithms forward to combine their predictions.

18.3.6 Finalize Model

Once you have found a model that you believe can make accurate predictions on unseen data, you are ready to finalize it. Finalizing a model may involve sub-tasks such as:

- Using an optimal model tuned by scikit-learn to make predictions on unseen data.
- Creating a standalone model using the parameters tuned by scikit-learn.

- Saving an optimal model to file for later use.

Once you make it this far you are ready to present results to stakeholders and/or deploy your model to start making predictions on unseen data.

18.4 Tips For Using The Template Well

This section lists tips that you can use to make the most of the machine learning project template in Python.

- **Fast First Pass**. Make a first-pass through the project steps as fast as possible. This will give you confidence that you have all the parts that you need and a baseline from which to improve.
- **Cycles**. The process in not linear but cyclic. You will loop between steps, and probably spend most of your time in tight loops between steps 3-4 or 3-4-5 until you achieve a level of accuracy that is sufficient or you run out of time.
- **Attempt Every Step**. It is easy to skip steps, especially if you are not confident or familiar with the tasks of that step. Try and do something at each step in the process, even if it does not improve accuracy. You can always build upon it later. Don't skip steps, just reduce their contribution.
- **Ratchet Accuracy**. The goal of the project is model accuracy. Every step contributes towards this goal. Treat changes that you make as experiments that increase accuracy as the golden path in the process and reorganize other steps around them. Accuracy is a ratchet that can only move in one direction (better, not worse).
- **Adapt As Needed**. Modify the steps as you need on a project, especially as you become more experienced with the template. Blur the edges of tasks, such as steps 4-5 to best serve model accuracy.

18.5 Summary

In this lesson you discovered a machine learning project template in Python. It laid out the steps of a predictive modeling machine learning project with the goal of maximizing model accuracy. You can copy-and-paste the template and use it to jump-start your current or next machine learning project in Python.

18.5.1 Next Step

Now that you know how to structure a predictive modeling machine learning project in Python, you need to put this knowledge to use. In the next lesson you will work through a simple case study problem end-to-end. This is a famous case study and the *hello world* of machine learning projects.