About the Book

If data is the new oil, then machine learning is the drill. As companies gain access to ever increasing quantities of raw data, the ability to deliver state-of-the-art predictive models that support business decision making becomes more and more valuable.

In this book, you’ll work on an end-to-end project based around a real data set of credit defaults, split up into bite-sized practical exercises. This case study approach simulates the working conditions you’ll experience in real-world data science projects.

You’ll learn how to use key Python packages, including pandas, Matplotlib, scikit-learn, XGBoost, and SHAP, to explore and process data, before moving on to fitting, evaluating, and tuning algorithms such as regularized logistic regression, random forest, and gradient boosting.

Now in its second edition, this book will take you through the complete process of exploring data and delivering machine learning models. Updated to Python 3.8, this new edition for 2021 includes brand new content on XGBoost and SHAP values.

By the end of this book, you’ll have the skills, understanding, and confidence to build your own machine learning models to help define business strategies based on real data.

About the Author

Stephen Klosterman is a Machine Learning Data Scientist with a background in math, environmental science, and ecology. His education includes a PhD in Biology from Harvard University, where he was assistant teacher of the Data Science course. His professional experience includes work in the environmental, health care, and financial sectors. At work, he likes to research and develop machine learning solutions that create value, and that stakeholders understand. In his spare time, he enjoys running, biking, paddle boarding, and music.

Objectives

* Load, explore, and process data using the pandas Python package
* Use Matplotlib to create effective data visualizations
* Implement predictive machine learning models with scikit-learn and XGBoost
* Use lasso and ridge regression to reduce model overfitting
* Build ensemble models of decision trees, using random forest and gradient boosting
* Evaluate model performance and interpret model predictions
* Deliver valuable insights by making clear business recommendations

Audience

Data Science Projects with Python – Second Edition is for anyone who wants to get started with data science and machine learning. If you’re keen to advance your career by using data analysis and predictive modeling to generate business insights, then this book is the perfect place to begin. To quickly grasp the concepts covered, it is recommended that you have basic experience with programming in Python or another similar language (R, Matlab, C, etc). Additionally, knowledge of statistics that would be covered in a basic course, including topics such as probability and linear regression, or a willingness to learn about these on your own while reading this book would be useful.

Approach

Data Science Projects with Python takes a practical case study approach to learning, teaching concepts in the context of a real-world dataset. Clear explanations will deepen your knowledge, while engaging exercises and challenging activities will reinforce it with hands-on practice.

About the Chapters

Chapter 1, Data Exploration and Cleaning, gets you started with Python and Jupyter notebooks. The chapter then explores the case study dataset and delves into exploratory data analysis, quality assurance, and data cleaning using pandas.

Chapter 2, Introduction to Scikit-Learn and Model Evaluation, introduces you to the evaluation metrics for binary classification models. You'll learn how to build and evaluate binary classification models using scikit-learn.

Chapter 3, Details of Logistic Regression and Feature Exploration, dives deep into logistic regression and feature exploration. You'll learn how to generate correlation plots of many features and a response variable and interpret logistic regression as a linear model.

Chapter 4, The Bias-Variance Trade-Off, explores of the foundational machine learning concepts of overfitting, underfitting, and the bias-variance trade-off by examining how the logistic regression model can be extended to address the overfitting problem.

Chapter 5, Decision Trees and Random Forests, introduces you to tree-based machine learning models. You'll learn how to train decision trees for machine learning purposes, visualize trained decision trees, and train random forests and visualize the results.

Chapter 6, Gradient Boosting, XGBoost, and SHAP (SHapley Additive exPlanations) Values, introduces you to two key concepts: gradient boosting and shapley additive explanations (SHAP). You'll learn to train XGBoost models and understand how SHAP values can be used to provide individualized explanations for model predictions from any dataset.

Chapter 7, Test Set Analysis, Financial Insights, and Delivery to the Client, presents several techniques for analyzing a model test set for deriving insights into likely model performance in the future. The chapter also describes key elements to consider when delivering and deploying a model, such as the format of delivery and ways to monitor the model as it is being used.

Hardware Requirements

For the optimal student experience, we recommend the following hardware configuration:

* Processor: Intel Core i5 or equivalent
* Memory: 4 GB RAM
* Storage: 35 GB available space

Software Requirements

You'll also need the following software installed in advance:

* OS: Windows 7 SP1 64-bit, Windows 8.1 64-bit or Windows 10 64-bit, Ubuntu Linux, or the latest version of OS X
* Browser: Google Chrome/Mozilla Firefox Latest Version
* Notepad++/Sublime Text as IDE (this is optional, as you can practice everything using the Jupyter Notebook on your browser)
* Python 3.8+. This book uses Python 3.9.5 for the majority of the chapters. However, certain sections in Chapter 6, Gradient Boosting, XGBoost, and SHAP (SHapley Additive exPlanations) Values, use an environment that runs on Python 3.8.2. You can find further details on setting up environments in the next section.
* Python libraries as needed (Jupyter, NumPy, Pandas, Matplotlib, and so on, installed via Anaconda as recommended below)

Installation and Setup

Before you start this book, it is recommended to install the Anaconda package manager and use it to coordinate installation of Python and its packages.

**Code Bundle**

Please find the code bundle for this book, hosted on GitHub at <https://github.com/PacktPublishing/Data-Science-Projects-with-Python-Second-Ed>.

Anaconda and Setting up Your Environment

Install Anaconda by following the instructions at this link: <https://www.anaconda.com/products/individual>.

The Python code in chapters 1-5 and 7 work with version 3.9.5. However, certain sections in Chapter 6, Gradient Boosting, XGBoost, and SHAP (SHapley Additive exPlanations) Values,specifically those on using SHAP, work only with version 3.8+ at the time of writing this book.Hence, we recommend that you set up two different Python environments. The guidance for setting up these environment is as follows:

Environment for Python 3.9.5: Once you have Anaconda installed, open a Terminal, if you're using macOS or Linux, or a Command Prompt window in Windows, and do the following:

1. Create an environment with most required packages. You can call it whatever you want; here it’s called dspwp2\_39. Copy and paste, or type the entire statement here on one line in the terminal:

conda create -n dspwp2\_39 python=3.9.5 jupyter=1.0.0 pandas=1.2.1 scikit-learn=0.23.2 numpy=1.19.2 matplotlib=3.3.2 seaborn=0.11.1 python-graphviz=0.15 xlrd=2.0.1

1. Type 'y' and [Enter] when prompted.
2. Activate the environment:

conda activate dspwp2

1. Type 'y' and [Enter] when prompted.
2. You are ready to use the environment. To deactivate it when finished:

conda deactivate

Note

While setting up your environment, if you get a PackagesNotFoundError error for specific packages, then we recommend that you install such packages using pip install package\_name == version\_number after you have created and activated your environment.

Environment for Python 3.8.2: Once you have Anaconda installed, open a Terminal, if you're using macOS or Linux, or a Command Prompt window in Windows, and do the following:

1. Create an environment with most required packages. You can call it whatever you want; here it’s called dspwp2\_38. Copy and paste, or type the entire statement here on one line in the terminal:

conda create -n dspwp2\_38 python=3.8.2 jupyter=1.0.0 pandas=1.2.1 scikit-learn=0.23.2 numpy=1.19.2 matplotlib=3.3.2 seaborn=0.11.1 python-graphviz=0.15 xlrd=2.0.1

1. Type 'y' and [Enter] when prompted.
2. Activate the environment:

conda activate dspwp2

1. Type 'y' and [Enter] when prompted.
2. You are ready to use the environment. To deactivate it when finished:

conda deactivate

We also have other code bundles from our rich catalog of books and videos available at <https://github.com/PacktPublishing/>. Check them out!

Conventions

Code words in the text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "By typing conda list at the command line, you can see all the packages installed in your environment."

A block of code is set as follows:

import numpy as np #numerical computation

import pandas as pd #data wrangling

import matplotlib.pyplot as plt #plotting package

#Next line helps with rendering plots

%matplotlib inline

import matplotlib as mpl #add'l plotting functionality

mpl.rcParams['figure.dpi'] = 400 #high res figures

import graphviz #to visualize decision trees

New terms and important words are shown in bold. Words that you see on the screen, for example, in menus or dialog boxes, appear in the text like this: "Create a new Python 3 notebook from the New menu as shown."

Code Presentation

Lines of code that span multiple lines are split using a backslash ( \ ). When the code is executed, Python will ignore the backslash, and treat the code on the next line as a direct continuation of the current line.

For example:

my\_new\_lr = LogisticRegression(penalty='l2', dual=False,\

                               tol=0.0001, C=1.0,\

                               fit\_intercept=True,\

                               intercept\_scaling=1,\

                               class\_weight=None,\

                               random\_state=None,\

                               solver='lbfgs',\

                               max\_iter=100,\

                               multi\_class='auto',\

                               verbose=0, warm\_start=False,\

                               n\_jobs=None, l1\_ratio=None)

Comments are added into code to help explain specific bits of logic. Single-line comments are denoted using the # symbol, as follows:

import pandas as pd

import matplotlib.pyplot as plt #import plotting package

#render plotting automatically

%matplotlib inline

Get in Touch

Feedback from our readers is always welcome.

General feedback: If you have any questions about this book, please mention the book title in the subject of your message and email us at customercare@packtpub.com.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you could report this to us. Please visit www.packtpub.com/support/errata and complete the form.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you could provide us with the location address or website name. Please contact us at copyright@packt.com with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit authors.packtpub.com.

Please Leave a Review

Let us know what you think by leaving a detailed, impartial review on Amazon. We appreciate all feedback – it helps us continue to make great products and help aspiring developers build their skills. Please spare a few minutes to give your thoughts – it makes a big difference to us.