University of Illinois – Deloitte Foundation Center for Business Analytics

Instructor's Guide for Data Analytics Foundations for Accountancy II

Course Description

Data Analytics Foundations for Accountancy II will build on concepts introduced in the first course to enable students to obtain, explore, and analyze richer and more complex data sets. Students will first learn how explore and analyze multi-dimensional data sets, before learning how to obtain text data embedded within websites and how to analyze text data by using standard Python techniques and regular expressions. Next, students will learn how to work with relational databases by using SQL, standard Python, and via the Pandas DataFrame. Similarly, students will learn how to work with times and dates within a Python program, and by extension how to work with time series data both in general and to perform predictive analytics. Finally, density estimation is introduced and demonstrated on both one- and two-dimensional data sets.

Course Goals and Objectives

Upon successful completion of this course, you will be able to:

- Analyze text data and time series data,
- Work with relational databases,
- Effectively work with probabilities,
- Clean, explore, visualize, and construct simple models for two- (or higher) dimensional data, and
- Visualize and analyze large data sets by estimating the density of data rather than focusing on individual data themselves.

Course Organization

This course is organized into eight modules. Each module consists of multiple lessons, which each contain an associated video explaining the lesson content, one or more external readings and included course Jupyter notebooks. Each module also includes a quiz (or assessment) that tests basic mastery of the lesson contents, and a programming assignment that tests synthesis of the lesson contents, especially into the broader module and course content.

Typically, a module is covered in one week, although this pace can be either faster or slower based on student needs. In addition, content can be selectively used from lessons and modules to provide supplemental learning content for an existing course. The module quiz is traditionally given through the course LMS in a timed fashion (e.g., thirty minutes). The assignments are completed on the course JupyterHub server, and are collected at the assignment deadline and processed by an autograder.

Original Content

The original content for this course consists of Jupyter notebooks, which are rendered to standard HTML webpages by github automatically.

HTML Webpage Content

The Jupyter notebook system can convert the original notebooks into a variety of different formats, including HTML, which is the standard format used to create web pages.

Special Notes

Software Versions

This course employs multiple Python modules to demonstrate important data analytics concepts. These open source software modules are undergoing continual development, and occasionally new versions of the software will cause the code in the course notebooks to throw warnings and potentially cause error conditions that might prevent a notebook from working. The notebooks in this course release have been tested with the following versions of Python and associated Python library (note, there is a special notebook called test-version.ipynb in the modified directory that will display your installed software versions for comparison).

```
# Display Python version
import sys
print(sys.version)

3.7.6 (default, Jan  8 2020, 13:42:34)
```

```
[Clang 4.0.1 (tags/RELEASE_401/final)]

# Display basic Python library versions
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib as mpl

print(f'Numpy version        = {np.__version__}')
print(f'Pandas version       = {pd.__version__}')
print(f'Matplotlib version   = {sns.__version__}')
print(f'Seaborn version      = {mpl.__version__}')

Numpy version        = 1.18.1
Pandas version       = 1.0.0
Matplotlib version   = 0.10.0
Seaborn version      = 3.1.2

!sqlite3 -version

3.30.1 2019-10-10 20:19:45
18db032d058f1436ce3dea84081f4ee5a0f2259ad97301d43c426bc7f3df1b0b
```

Data Storage

The machine learning notebooks in this course are designed to use data files stored locally; however, if the data file is not found, the file will be pulled automatically from an internet archive to the local file system. The notebooks are hardcoded to look for the data files in the data directory, which is expected to exist in the home directory of the current user. For example, on a Mac OSX computer with a username Joe, the directory /Users/Joe/data. This directory will need to be created for the notebooks to be able to successfully download and store the data files.

In addition, some of the more advanced database concepts are demonstrated on a sample database, i2ds available as a separate download; note this file should be placed in a database directory, which should be located in the parent directory where the notebooks are located. This database is constructed from the airline on-time data set hosted by the American Statistical Society.

Course Contents

Module 1: Applied Data Analytics

In this module, you will learn to perform basic data analytics tasks by using Python. First, you will learn about how advanced data analytics are often used in business settings. You will also learn about statistical issues that can impact understanding multi-dimensional data, which will allow you to avoid them in the future. Second, you will learn how to work with two-dimensional data by using the NumPy module, including a discussion on analytically quantifying correlations in data. Finally, you will learn about ordinary linear regression and how this technique can be used to model the relationship between two variables.

Learning Objectives

By the end of this module, you should be able to:

- Explain how advanced analytics can empower business,
- Work with multi-dimensional arrays in NumPy,
- Quantify correlations between data by using Python,
- Understand statistical concerns when working with multi-dimensional data, and
- Compute and interpret linear models.

Module 2: Introduction to Text Analytics

In this module, you will learn to perform basic text analytics tasks by using Python. First, you will learn how to perform text processing by using Python. This will include using built-in string functions to parse text data. Next, you will learn about regular expressions, which can perform fast search-and-replace operations on text data and is an important data analytic skill. Finally, you will learn about webpages, and how to programmatically extract data from a webpage or website.

Learning Objectives

By the end of this module, you should be able to:

- Use Python string functions to find and modify text data,
- Effectively use regular expressions in a Python program, and
- Scrape data from a website.

Module 3: Introduction to Relational Databases

In this module, you will learn about relational database management systems (RDBMS), still the most popular data storage technology. You will be introduced to SQL, the language by which database users interact with a relational database. To test database and SQL concepts, we will use the open source database system known as SQLite, specifically version 3 of the SQLite database known as sqlite3. Note, most of the documentation you will read and follow online will refer to running the sqlite program at the command line. You will run the sqlite3 program both from within a Juptyer notebook and at the command line. In this module, you will use SQLite to learn about SQL data definition language (DDL) functionality and SQL data manipulation language (DML) functionality.

Learning Objectives

By the end of this module, you should be able to:

- Understand how databases are commonly used for business applications,
- Understand the basic concepts behind a relational database system,
- Understand how to create and use a SQLite database,
- Understand how to use SQL to interact with a SQLite database, and
- Be able to interact with a database by using a Python program.

Module 4: Introduction to Python and Databases

In this module, you will learn more about databases, including advanced SQL concepts, working with a database from a Python program, and how to perform advanced tasks with a Pandas DataFrame. First, we will explore primary keys, indexes, views, and joins in SQL and also discuss how to modify schemas after they have been created. Second, we look into connecting to a relational database from within a Python program. Finally, we look at connecting a Pandas DataFrame to a database and how to use a DataFrame to clean, explore, visualize, and analyze a large, complex data set.

Learning Objectives

By the end of this module, you should be able to:

- Perform advanced SQL tasks with a relational database,
- Connect a Python program with a relational database,
- Save and load a DataFrame form a database, and
- Process data in a DataFrame to glean new insights.

Module 5: Introduction to Probability

In this module, you will learn the basics of probability, and how it relates to statistical data analysis. First, you will learn about the basic concepts of probability, including random variables, the calculation of simple probabilities, and several theoretical distributions that commonly occur in discussions of probability. Next, you will learn about conditional probability and Bayes theorem. Third, you will learn to calculate probabilities and to apply Bayes theorem directly by using Python. Finally, you will learn to work with both empirical and theoretical distributions in Python, and how to model an empirical data set by using a theoretical distribution.

Learning Objectives

By the end of this module, you should be able to:

- Calculate and interpret basic probability concepts,
- Calculate and interpret conditional probability,
- Understand and apply Bayes theorem,
- Understand the most important empirical distribution measures, and
- Understand and know how to apply the most important theoretical distributions.

Module 6: Introduction to Time Series Data

This module introduces time and date data, which provide unique learning opportunities and challenges. First, we will review how businesses use historical time series data to generate forecasts. Second, we will discuss how to properly handle time and date features within a Python program. Next, we will extend this discussion to handle data indexed (or tagged) by time and date information, which is known as time series data. Finally, we will learn about Fourier analysis, which separates time series data into constituent, periodic components.

Learning Objectives

By the end of this module, you should be able to:

- Discuss different ways businesses use time series data to forecast future events,
- Work effectively with time and date data by using Python,
- Use the Pandas library to manipulate data sets that include time and date features, and
- Apply Fourier analysis to identify periodic components within a time series data set.

Module 7: Introduction to Time Series Analysis

The real world is filled with data generation processes that depend on time. For example, we might record the amount of rain that falls each day, the number of cars that travel across a bridge every hour, or the price of a commodity every second. To analyze these types of data, we must understand how time series data are different than the other forms of data and what these differences mean for subsequent analyses. In this module, you will learn basic concepts in time series analysis that enable time series data to be modeled either to better understand the data generation process or to make reliable forecasts for future observations.

Learning Objectives

By the end of this module, you should be able to:

- Explain the importance of time series analysis for business,
- Apply basic time series analysis and forecasting techniques such as AR, MA, ARMA, and ARIMA by using Python,
- Explain hidden Markov models and apply them by using Python to describe simple systems, and
- Explain Kalman filters and apply them by using Python to model simple systems.

Module 8: Introduction to Density Estimation

Often, as part of exploratory data analysis, a histogram is used to understand how data are distributed, and in fact this technique can be used to compute a probability mass function (or PMF) from a data set as was shown in an earlier module. However, the binning approach has issues, including a dependance on the number and width of the bins used to compute the histogram. One approach to overcome these issues is to fit a function to the binned data, which is known as parametric estimation. Alternatively, we can construct an approximation to the data by employing a non-parametric density estimation. The most commonly used non-parametric technique is kernel density estimation (or KDE). In this module, you will learn about density estimation and specifically how to employ KDE. One often overlooked aspect of density estimation is the model representation that is generated for the data, which can be used to emulate new data. This concept is demonstrated by applying density estimation to images of handwritten digits and sampling from the resulting model.

Learning Objectives

By the end of this module, you should be able to:

- Understand both parametric and non-parametric density estimation,
- Understand the basic concepts behind kernel density estimation,
- Use density estimation to approximate or smooth discrete data,
- Construct a kernel density estimate by using the Python scikit-learn module, and
- Sample from a density estimate model to generate new data.