

**College of San Mateo
Official Course Outline**

1. **COURSE ID:** CIS 503 **TITLE:** Data Visualization and Text Analysis in Python
Units: 3.0 units **Hours/Semester:** 48.0-54.0 Lecture hours; and 96.0-108.0 Homework hours
Method of Grading: Grade Option (Letter Grade or Pass/No Pass)
Prerequisite: CIS 117

2. **COURSE DESIGNATION:**

Degree Credit

Transfer credit: CSU; UC

3. **COURSE DESCRIPTIONS:**

Catalog Description:

Introduction to information visualization, reporting and charting using the Matplotlib library. Text mining and text manipulation basics using the NLTK framework will also be covered.

4. **STUDENT LEARNING OUTCOME(S) (SLO'S):**

Upon successful completion of this course, a student will meet the following outcomes:

1. Create a visualization using Matplotlib.
2. Identify the functions that are best for particular problems.
3. Describe what makes a good or bad visualization.
4. Understand best practices for creating basic charts.
5. Understand how text is handled in Python.
6. Describe the NLTK framework for manipulating text

5. **SPECIFIC INSTRUCTIONAL OBJECTIVES:**

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

1. Create a visualization using Matplotlib.
2. Identify the functions that are best for particular problems.
3. Describe what makes a good or bad visualization.
4. Understand best practices for creating basic charts.
5. Apply basic natural language processing methods.
6. Understand how text is handled in Python.
7. Describe the NLTK framework for manipulating text
8. Write code that groups documents by topic.

6. **COURSE CONTENT:**

Lecture Content:

Principles of Information Visualization

1. Tools for Thinking about Design
2. Data-ink ratio
3. Lie Factor and Spark Lines

Basic Charting

1. Matplotlib Architecture
2. Basic Plotting with Matplotlib
3. Scatterplots
4. Line Plots
5. Bar Charts

Charting Fundamentals

1. Subplots
2. Histograms
3. Box Plots

4. Heatmaps
5. Animation
6. Interactivity
7. Plotting with Pandas
8. Seaborn

Applied Visualizations

1. Plotting with Pandas
2. Seaborn

Working with Text in Python

1. Handling Text in Python
2. Regular Expressions
3. Regex with Pandas and Named Groups
4. Internationalization and Issues with Non-ASCII Characters

Basic Natural Language Processing

1. Basic NLP tasks with NLTK
2. Advanced NLP tasks with NLTK

Text Classification

1. Identifying Features from Text
2. Naive Bayes Classifiers
3. Naive Bayes Variations
4. Support Vector Machines
5. Learning Text Classifiers in Python

Topic Modeling

1. Semantic Text Similarity
2. Topic Modeling
3. Generative Models and LDA
4. Information Extraction

7. REPRESENTATIVE METHODS OF INSTRUCTION:

Typical methods of instruction may include:

- A. Lecture
- B. Directed Study
- C. Activity
- D. Discussion
- E. Individualized Instruction
- F. Observation and Demonstration
- G. Other (Specify): The course includes the following instructional methods as appropriate: Lecture, to introduce new topics; "Models" for problem-solving techniques; In class group problem solving, each person contributing a potential "next step"; Student participation in short in-class projects (in teacher-organized small groups) ; Q/A sessions in which the students provide both the question AND the answers; Small groups of students working together to solve significant programming assignments.

8. REPRESENTATIVE ASSIGNMENTS

Representative assignments in this course may include, but are not limited to the following:

Writing Assignments:

Students will be assigned weekly homework problems from the required textbook.

Reading Assignments:

Students will read all chapters of the required textbook, reading parallel current assignments, and lecture content.

Other Outside Assignments:

Weekly homework problems
Internet research

9. REPRESENTATIVE METHODS OF EVALUATION

Representative methods of evaluation may include:

- A. Class Participation
- B. Exams/Tests
- C. Group Projects
- D. Lab Activities
- E. Projects
- F. Quizzes
- G. Written examination
- H. Weekly textbook readings, textbook excursions, and bi-weekly programming projects comprises the majority of out-of-class assignments. One or more of the programming projects will be a small-group project to provide experience in a realistic program development environment, in order to improve technical communication skills and simulate a more realistic programming environment.

10. REPRESENTATIVE TEXT(S):

Possible textbooks include:

- A. B.J.Korites. *Python Graphics: A Reference for Creating 2D and 3D Images*, 1 ed. Apress, 2019
- B. Kyran Dale. *Data Visualization with Python and JavaScript*, ed. O'Reilly Media Inc., 2016
- C. Benjamin Bengfort, Tony Ojeda, Rebecca Bilbro. *Applied Text Analysis with Python*, 1 ed. O'Reilly Media Inc., 2018

Origination Date: October 2019

Curriculum Committee Approval Date: December 2019

Effective Term: Fall 2020

Course Originator: Kamran Eftekhari