ISM 6930: Text Analytics
Sec 002; CRN 93090, BSN 118
M 6:30 am – 9:45 pm, Fall Semester 2023

Professor: Dr. Tim Smith	*Office:* CIS 2079
E-mail: smith515@usf.edu	*Office Hours:* M 3:15-4:15 pm or by appointment

Course Overview:

Over 80% of the data we generate today is unstructured, such as text (e.g., blog posts, social media feeds, tweets, corporate reports), audio, and video. Text data can tell us a lot about business performance, customer preferences, etc., but is often ignored because we do not have good computational tools to handle such data. Text data is also very "noisy" and requires extensive effort to extract meaningful features, topics, sentiments, meaning, patterns, summary, etc., from this data. The types of analysis we can do with text data go well beyond classification or clustering to natural language processing (NLP) applications such as understanding text, answering text questions, summarizing text, text conversations, etc. This course will discuss how to work with noisy text data, clean them, convert them into features for computational modeling, build machine (or deep) learning models to analyze them and extract usable insights from this data. The course requires advanced knowledge of Python.

Learning Objectives:

To learn how to:

- Source data for text analytics, including web scraping and API.
- Clean, preprocess, and structure text data for analysis.
- Extract latent features, topics, and word associations from text data.
- Build computational models to extract the semantic meaning of text data.
- Write Python code and use Python libraries to analyze huge volumes of text data.
- Analyze data using machine learning or deep learning models.
- Conceptualize, plan, conduct, and document a text analytics project from end to end.

Course Prerequisite:

ISM 6251 (Data Science Programming) or advanced proficiency in Python (e.g., writing your own Python functions and classes, working with JSON objects and Pandas data frames, and building machine learning and deep learning models using Scikit-Learn. This is a hard-core coding class; if you are not proficient in Python, you should NOT take this course.

Grading:

Grade components and weights:

Individual Assignments (five)	45-50%
Exams (two)	35%
Team Project (one)	15-20%

Grading scale: A+: >97%, A: >93%, A-: >90%, B+: >87%, B: >83%, B-: >80%, C+: >77%, C: >73%, C-: >70%, D+: >67%, D: >63%, D-: >=60%, F: Below 60%

Books and Materials:

Books: None. Materials for this class have been sourced from many books and online sources and the professor's personal knowledge of the domain and work done by his students.

Readings, slides, data sets, and other materials: Download from Canvas.

Hardware/software: A fast computer with more RAM and GPU is preferred since text processing is resource-intensive. Either Windows or Mac is OK. Install the Miniconda distribution freely available from https://docs.conda.io/en/latest/miniconda.html (we will use Jupyter Lab/Notebook as our IDE). Bring your laptop computer EVERY class for in-class exercises.

Assignments:

Assignments are due every week at the start of class. Assignments must be submitted via Canvas and are date-stamped when submitted. Correct answers will be discussed in class. Hence, late assignments are not accepted. You are welcome to help each other with the assignment for learning purposes. Still, the assignments that you submit must represent your own individual effort, and you must be able to explain the code you submitted for your assignment. Failing to do so will be viewed as "plagiarism," leading to a zero grade on that assignment. Repeat plagiarism will result in an F grade in the class, and a third instance of plagiarism will lead to expulsion from the MS-BAIS program. Exam questions will be based on these assignments; hence copying assignments will hurt your exam performance.

Exams:

There will be two online exams during the semester. In each exam, you will be asked to solve one problem, similar to the assignments, and you will be graded based on not just what you did but also on your thought process and the simplicity, elegance, and understandability of your code. Exams will be similar to the "screening test" used by companies like Google and Microsoft. Exams are open-book, open-notes, and open Internet – but are time-limited; therefore, if you depend too heavily on such support tools, you will likely not have enough time to complete the submission. The best way to prepare for an exam is to pay attention in class, complete assignments by yourself, and continuously look for ways to improve your solutions.

Please do NOT miss any exam without a documented medical or family emergency. A "documented emergency" must be accompanied by appropriate documentation, such as a doctor's note. Make-up exams will be different and more complex than regular ones.

Team Project:

This term project, to be done in teams of 3-4 students, is the heart of this class. Sample projects from prior semesters are posted on Canvas. For your project, you can either (a) pick a topic of interest or (b) select a topic suggested by the professor. Whatever topic you select, you must discuss the suitability of that topic and your data-sourcing strategy with the professor. Note that simple classification or clustering of text is insufficient for this project; the project must focus on extracting meaningful information from text.

In the past, many teams spent so much time searching for the "right project" that they did not leave themselves sufficient time to do a good-quality project. If you are struggling to find a suitable project, you can select one from the following list:

- Fake/paid review classifier for Yelp or Amazon (using meaningful features).
- Multilingual classification (e.g., hate speech vs. not; bot vs. not, etc.)
- Designing intelligent chatbots (Q&A system).

- Creating feature-based (aspect-based) review scores from overall product reviews.
- Live consumer sentiment and service failure dashboard using Twitter feeds (may require some Javascript/full-stack development).
- Corporate risk assessment or detecting corporate fraud from SEC 10-K filings.
- Extracting useful information (e.g., disease co-occurrence) from hospital discharge reports.
- Understanding what makes a Twitter, Facebook, Tiktok, or Instagram post go viral (may require integration of text analytics and statistical data modeling).
- NLP albumentation: Text augmentation for creating labeled text data (antonym replacement, GPT-3, SBERT, NLPaug library).

Your project report (10-12 pages, plus appendix) must include the following sections: (1) executive summary; (2) problem definition and significance; (3) prior literature (how have others tried to address this problem and with what outcomes); (4) data source/preparation; (5) text analytics workflow; (6) exploratory data analysis; (7) choice and rationale for text analytic methods, and results (benchmark against prior studies); (8) actionable recommendations; (9) references, and (10) appendix (Python code and output). The project report must be sufficiently detailed, include appropriate graphics, and be of professional quality. Intermediate project deliverables are due at different points during the semester, and the final project report is due at the end of the semester. See the project report assignment on Canvas for further details. Sample projects are posted on Canvas. NOTE: Original project ideas will be marked more favorably; therefore, you are encouraged to develop your own project ideas.

Note that data sourcing (web scraping, web APIs, etc.) and cleaning are integral and graded components of this project. A Kaggle or UCI dataset of unknown quality is unacceptable and will get you zero points on the project's data sourcing/cleaning sections. Such data is okay for personal learning but not for serious analysis.

Your professor will compare your code with similar projects on the Internet/Github, run a plagiarism check, and ask you to explain your work. Feel free to browse online code and learn from them, but your work must be original, practical, and useful. Downloading an online project, making minor modifications, and submitting it as your team project will be considered "plagiarism" and will get the entire team a zero grade.

The professor is available during office hours to help you with your project idea(s) throughout the semester. Use this free help or lose it. Please use this project to showcase what you learned inside and outside class. A well-done project can help you get internships and jobs. I give extra credit to teams who surpass my expectations, experiment with new methods and tools, and help the class learn something new.

E-mail me the names of your team members and a tentative topic for your project before the second day of class. Choose your team members "wisely", making sure that you have team members proficient in Python, in reading and understanding technical papers, and in writing professional-quality reports. If someone in your team does not show up for team meetings or is perpetually late with their work, you have the option to fire that person by majority vote.

Class Policies:

Since this is a business class, I expect a certain level of **business professionalism** in this class. This means coming to class on time, not making excuses, not plagiarizing content from the Internet, timely completion of assigned work, willingness to learn things on your own, managing your team project like a professional work project with weekly scrums and reviews, etc. Students who don't behave professionally will be penalized appropriately.

For USF policy and procedures regarding academic integrity, academic grievance, disability access, disruption to academic progress, religious observances, sexual misconduct/harassment, and statement of academic continuity, please visit https://www.usf.edu/provost/faculty/core-syllabus-policy-statements.aspx

Class Schedule:

Week	Date	Topics & Class Activities	Deliverables
1	Aug 21	Introduction to Text Mining File I/O and text encoding Python: “I Have a Dream” Speech	
2	Aug 28	Corpus Preparation Unstructured Text Files Structured Text Files JSON Data Collection Python: WebAPIs Python: Web Scraping	P1: Teams & Topics Due
	Sept 4	No class – Labor Day	
3	Sept 11	Text Preprocessing using Regex Text Visualization using Seaborn Simple NLP using Textblob Python: Donald Trump’s tweets	
4	Sept 18	POS Tagging & Named Entity Recognition TF-IDF	A1: US Presidential Speeches
5	Sept 25	Word Embedding Python: Word2vec, Cosine similarity	A2: Glassdoor Employee Review Analysis
6	Oct 2	**MIDTERM EXAM**	P2: Problem Significance & Prior Work
7	Oct 9	Supervised Learning: Text Classification Python: Predicting success of online news articles, Multiclass classification of CFPB complaints	
8	Oct 16	Unsupervised Learning: Text Clustering Topic Modeling: LDA, LSA Python: Sentiment analysis of airline tweets Python: Clustering of CFPB complaints, Dimension reduction of news headlines Python: Topic modeling of airline tweets	A3: Customer Service Query
9	Oct 23	Deep Learning & Autoencoders Python: Classifying hate speech Python TensorFlow and Keras	A4: BBC News P3: Data Preparation, Visualization, and Prelim Analysis
10	Oct 30	Transformer Architectures Semantic text similarity using BERT Text translation, Text Summarization & Text Generation using GPT-2	A5: Clustering P4: Final Project Presentation
11	Nov 6	**FINAL EXAM**	P5: Project Report

Note: The syllabus is tentative and subject to change. Any such changes will be announced in class.