INFORMATION SYSTEMS EDUCATION JOURNAL

Volume 19, No. 3
June 2021
ISSN: 1545-679X

In this issue:

iscap
Information Systems & Computing Academic Professionals

The **Information Systems Education Journal** (ISEDJ) is a double-blind peer-reviewed academic journal published by **ISCAP** (Information Systems and Computing Academic Professionals). Publishing frequency is six times per year. The first year of publication was 2003.

ISEDJ is published online (https://isedj.org). Our sister publication, the Proceedings of EDSIGCON (https://proc.iscap.info) features all papers, panels, workshops, and presentations from the conference.

The journal acceptance review process involves a minimum of three double-blind peer reviews, where both the reviewer is not aware of the identities of the authors and the authors are not aware of the identities of the reviewers. The initial reviews happen before the EDSIGCON conference. At that point papers are divided into award papers (top 15%), other journal papers (top 25%), unsettled papers, and non-journal papers. The unsettled papers are subjected to a second round of blind peer review to establish whether they will be accepted to the journal or not. Those papers that are deemed of sufficient quality are accepted for publication in the ISEDJ journal. Currently the target acceptance rate for the journal is under 40%.

Information Systems Education Journal is pleased to be listed in the Cabell's Directory of Publishing Opportunities in Educational Technology and Library Science, in both the electronic and printed editions. Questions should be addressed to the editor at editor@isedj.org or the publisher at publisher@isedj.org. Special thanks to members of ISCAP/EDSIG who perform the editorial and review processes for ISEDJ.

INFORMATION SYSTEMS EDUCATION JOURNAL

From Engagement to Empowerment: Project-Based Learning in Python Coding Courses

Mark Frydenberg
Computer Information Systems Department
Bentley University
Waltham, Massachusetts
mfrydenberg@bentley.edu

Kevin Mentzer
Bryant University
Information Systems and Analytics Department
Smithfield, Rhode Island
kmentzer@bryant.edu

Abstract

Project-based learning (PBL) engages students deeply with course concepts and empowers them to drive their own learning through the development of solutions to real-world challenges. By taking ownership of and completing a project that they designed, students develop and demonstrate creativity, critical thinking, and collaboration skills. This paper describes two different software development projects, designed with a PBL approach, in Python coding courses at two business universities in the United States, in which students queried real-world data to answer their own questions and interpret the results. The authors contend that projects based on a PBL approach motivate students for self-exploration and allow for the measure of student learning. The authors present their respective projects, share examples of student work, and offer suggestions and lessons learned from implementing PBL assignments in their classrooms. Finally, the authors reflect, through sharing student comments, on how key aspects of PBL are manifest in this project and discuss challenges in offering and managing PBL assignments. With Python's popularity on the rise, these two class examples serve as a model for how instructors can incorporate autonomy in PBL assignments, offering a valuable learning opportunity for students to create software applications that meaningfully demonstrate their coding skills.

Keywords: project-based learning, Python, data analytics, data science, data visualization, coding

1. INTRODUCTION

Project-based learning (PBL) describes a learning scenario where students are engaged developing solutions to real-world problems often of their own design. The process of identifying a problem and developing a solution contributes to learning. Instructors need to specify required tasks, encourage students to think creatively, keep them motivated.

With its foundations in constructivism, which encourages students to learn through designing their own learning experiences, PBL requires a motivating problem or question for students to investigate. This culminates in the students creating original artefacts that illustrate their findings and demonstrate their understanding of a problem (Blumenfeld, Soloway, Marx, Krajcik, Guzdial, & Palincsar, 1991) process of completing such a project moves students from a place of engagement to a place of empowerment as they take control over their own learning, assess their own knowledge and skills, and demonstrate their competencies in a relevant project of their own design.

This paper describes how a PBL approach informed two software development projects given in Python coding courses at two business universities in the United States. The authors present their respective projects and requirements, share examples of student work, provide student reflections, and offer suggestions and lessons learned from implementing PBL assignments in their classrooms.

A contribution of this work is that it illustrates how carefully crafted coding projects such as these can influence student learning. While the literature has addressed PBL approaches in coding courses, this paper has the unique focus of using data analytics tools in a Python coding course to engage students in interacting in a project of their own choosing, and empower them to discern meaning from information by identifying their own requirements for analyzing real-world data.

These research questions guided this study:

- How can instructors design a course assignment that exemplify key aspects of PBL?
- Can a PBL approach motivate students and serve as an authentic measure of student learning?

2. PROJECT BASED LEARNING IN CODING COURSES

Many introductory programming courses include coding assignments of varying complexity, where the instructor specifies requirements or outcomes for students to complete. Assignments often are associated with textbook chapters or learning modules: when the week's lesson covers loops and if statements, the instructor's carefully constructed assignment ensures their use in the solution. All students work on essentially the same assignment (though some instructors may modify an assignment's requirements from semester to semester or within multiple sections of a course, to offer variety and promote academic integrity). In a PBL approach, students create their own questions, focusing on process over product, as "engaging students in the process of inquiry involves guiding them to ask meaningful questions to investigate compelling real-world problems. Through this process, students build crucial problem-solving skills and learn how to generate creative solutions to complex problems" (McKay, Frank, 2017).

Project Based Learning emphasizes student involvement through direct experience in directing their own learning. Ownership of the project is emphasized throughout the project by having the student in control of the project definition. Students utilize creativity through both the unique definition of the project as well as the election of techniques used to execute the project. Collaboration happens when student interact and provide feedback between peers. Finally, critical thinking enables problem solving throughout the project. Figure 1 summarizes these key aspects of PBL.

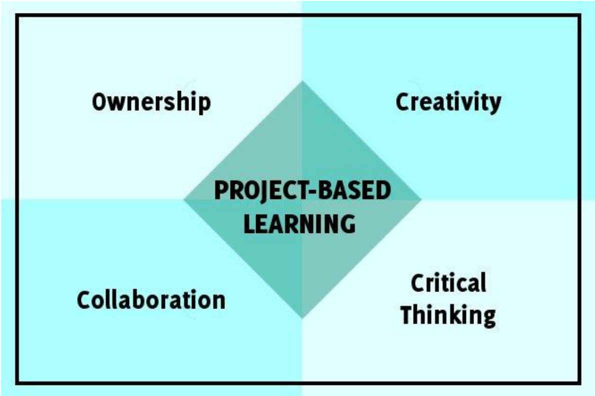

Figure 1. Key aspects of project-based learning [Adapted from (Stefanou, Stolk, Prince, Chen, & Lord, 2013)]

In a well-designed PBL experience, the student has ownership of the project. Student learning outcomes are improved if the project demands both creativity and critical thinking (Rice & Shannon, 2016; Sharkey & Weimer, 2003)(Rice & Shannon, 2016; Sharkey & Weimer, 2003). Finally, in many learner-centered environments, different forms of collaboration, such as learning from and with peers, often improve the quality of course projects (Aditomo, Goodyear, Bliuc, & Ellis, 2013; Jackson & Bruegmann, 2009; Stefanou, Stolk, Prince, Chen, & Lord, 2013).

VanDeGrift describes a learning scenario where students take ownership by creating their own programming problems in an introductory CS 1 course. "Every assignment includes open ended elements to encourage students to decide how to define part of the specification and provide latitude for students to be creative in their design and implementation" (VanDeGrift, 2015, p. 54). Students build their own interpretations of the material based on their own experiences, resulting in projects that foster creativity, maintain interest, and encourage students to take ownership of their projects.

When implementing a PBL scenario in a coding course, assignments are usually of a larger scale, and require students to select the programming constructs, modules, and data analysis most appropriate to implementing or discovering a solution. "Project-based learning, unlike the traditional textbook/lecture approach, motivates the student to do additional work, illustrates to the student the value of the material covered, and most importantly, provides practical experiences that enrich the student's academic growth" (Baugh, 2011, p. 15).

Courses offering PBL differ from those offering individual or group active learning problem-solving exercises. While students often work on specific well-defined problems during class in flipped classroom environments, (Bergmann & Sams, 2014; Whittington, 2004), in a PBL environment, students identify a problem, often open-ended, to investigate, and then implement their solution in a software application. "Project work … requires the student to develop an entire system - a complicated and new task for most students"(Scherz & Polak, 1999, p. 88).

PBL increases student engagement by having students apply their knowledge as they complete learning activities to challenge their understanding and involve them in the learning process, rather than passively watching, listening, or reading about the topic. Projects are adaptable to a student's interests, abilities, and needs. PBL enriches the classroom experience as students work on different problems in assignments of varying durations, requiring them to integrate their knowledge of several topics. The instructor's role shifts from providing solutions to helping students overcome immediate challenges and roadblocks so they can move on independently with their work. Students often work with or share their work with each other.

As students long for finding relevance and autonomy in the classroom, instructors are evolving the way they offer students assignments to demonstrate their knowledge. In a PBL environment, course projects shift from instructors developing homework problems or exams for students to complete, to students identifying their own problems to solve that meet specified learning objectives. Assignments range from defining their own problems to creating their own final exam questions (Brink, Capps, & Sutko, 2004; Brown, 1991; Jones, Jennifer, 2016). This expands the student's role from learner to assessor, as the process of making up one's own project or exam requires determining relevant topics, examining one's own learning and capabilities, and developing a mechanism to demonstrate competency and knowledge. The process requires use of higher order thinking skills (Bloom, 1956) to generate problems that required more than mere memorization or recall of facts.

3. PYTHON COURSE DESCRIPTIONS

This paper describes two different PBL learning assignments implemented in undergraduate Python coding courses at two universities. Students in both classes completed a project in which they had to use real world data to answer their own questions to demonstrate their mastery of several learning outcomes. Section 5 summarizes comments and responses to open-ended survey questions from students as they reflected on their learning and the value of a PBL methodology in completing their projects.

Both courses met in person at their respective universities during the spring 2020 semester until spring break, and then moved to online delivery in March 2020 because of the COVID-19 pandemic. The mid-semester shift online informed the creation of PBL assignments in these classes as both instructors considered alternative means for students to demonstrate their learning outcomes from the course in a way that genuinely reflected their newfound skills. Administering online exams brought many practical concerns; giving students the opportunity to design, build, present, and explain their solutions offered a practical way to evaluate a student's ability to master and apply course concepts.

The next sections describe the two courses in which the authors implemented PBL final projects in lieu of a more standard final exam, such as multiple choice or pencil-and-paper problems.

CS 299: Problem Solving with Coding in Python

CS 299, Problem Solving with Coding in Python, is an experimental elective open to all students at Bentley University, a northeastern U.S. business university. This course introduces problem solving using programming and teaches the fundamental concepts of algorithm development along with the underlying abstractions that are the basis of software systems. Students develop and integrate critical thinking skills by creating solutions to problems in a systematic, algorithmic manner using the Python programming language. In addition to teaching fundamental Python coding concepts, four class sessions included computational thinking topics and methods:

filtering data based on what is relevant (abstraction), developing algorithms, breaking problems into smaller problems (decomposition), and recognizing patterns (Astrachan, Hambrusch, Peckham, & Settle, 2009; Bell & Lodi, 2019; Rich & Hodges, 2017; Sengupta, Dickes, & Farris, 2018). These learning experiences are paramount in developing computational thinking, an ability to solve complex problems from authentic contexts and everyday life situations by decomposing them into smaller steps that are systematic and suitable for automation.

Students completed many small-group coding exercises and commented on each other's solutions during class so their peers could see alternative solutions to the same problems.

Throughout the course, understanding of coding concepts reinforced throughout the course by the development of several standalone applications, in which the instructor emphasizes the importance of writing efficient, clear, and well-structured code. No prior knowledge of Python or other programming languages is required.

This course met for two 80-minute sessions each week in a 14-week semester. The course had 27 students enrolled, 61% of whom had no prior coding experience. Students were primarily a mix of sophomores and juniors, most of whom were Computer Information Systems (CIS) or Finance majors, or CIS or Data Technologies minors. Each class session included instructor-led presentations and demonstrations, and several in-class exercises, completed in small groups, that reinforced the topics presented.

This course presents basic programming concepts and techniques using version 3 of the Python programming language, such as loops and selection statements; data structures (e.g., lists and dictionaries); classes, and objects. Instructors omitted advanced topics such as higher order functions (e.g., map, reduce, filter, lambda), and other topics frequently taught in Java programming courses (e.g., graphics and user interface design), teaching instead, basic capabilities of several popular Python libraries for data analysis: NumPy, Matplotlib, and Pandas. The course also introduced Streamlit (Treuille, Teixeira, & Kelly, 2020), an open-source app framework to code interactive web pages, to display their results. Incorporating Streamlit moves Python applications out of the console window and into a browser, using a simple platform to create web applications and share their work more widely

Several assessments contribute to evaluating a student's performance: five programming assignments (40%), class participation including completing in-class exercises (5%), short practice programs started during and often completed after each class (10%), a hands-on midterm exam (20%), and a design-your-own final project (25%) in lieu of a standard final exam.

Table 1 in Appendix 1 presents the topics covered in the five programming assignments.

ISA 330: Programing for Data Science

ISA 330, Programming for Data Science, is the second course in Python for students majoring in Data Science at Bryant University in the northeastern United States. This course, which has an introductory Python course as a prerequisite, is an advanced Python programming course focusing on common programming tools used for Data Science application development with an emphasis on libraries commonly used by data scientists (such as NumPy, Pandas, Matplotlib). Data analysts often implement their solutions using programming languages such as R and Python. Because of this, the data analyst/scientist must be comfortable in such development environments and be able to understand when a solution needs to be programmatically developed. The course covers hands-on programming techniques for analytics, including web scraping and other data extraction techniques, data transformation, data staging, data analysis, and finally data presentation and visualization. The course gives the students the skills to highlight their capability of producing notebooks appropriate for a data analytics/data science application.

This course runs each semester with one section offered. The students are primarily a mix of sophomores and juniors. Roughly, 75% of the students are data science majors and the rest is a mix of other business or mathematics majors. Due to the heavy hands-on programming aspect of the course, the class has a maximum of 25 students. The course typically meets three times a week for 50 minutes each session.

Even prior to the moving online after spring break, the course had a flipped component where students watched pre-recorded videos of lectures on their own schedule outside of class. This allowed the class time clear up anything that the students were still unsure about and work on in-class exercises meant to reinforce the concepts learned in the recorded lectures.

In addition to the recorded lectures, students worked with provided Jupyter notebooks that demonstrated the topics for the week. As part of their homework, the student had to modify these notebooks to expand, or modify, the notebook's functionality.

Multiple methods of assessment contributed to evaluating student performance including seven programming assignments (30%), three in class hands-on exams (30%), class participation including attendance and quizzes (10%), and the final project (30%).

Table 2 in Appendix 1 presents the topics covered in the seven programming assignments.

4. PROJECT DESCRIPTIONS

Introducing project-based learning assignments in these courses allows students to demonstrate their skills in applying course concepts to solve real-world problems. The variation among student projects and solutions encourages creativity and engagement as students identify the project components that they will implement to meet the project requirements.

CS 299 Project

CS 299 presented several ideas related to computational thinking and good practices for visualizing data in addition to introducing fundamental coding concepts and principles. The final project for the course required students to demonstrate mastery of these concepts. Given a data file containing approximately 3,400 actual Boston-area AirBnB listings available from http://insideairbnb.com/get-the-data.html, the project had students describe two questions for which they would find like to find answers from the data, and design two visualizations (charts, graphs, or maps) to display the results. Appendix 2, Figures 1 and 2 show screenshots of two sample student projects. Students completed the project in these phases:

Phase 1. Design. Describe two questions and two visualizations that you can create to analyze this data. Examples include: What are the most expensive rentals in each neighborhood? Is there a correlation between reviews and nightly prices? How many rentals are available in each neighborhood? Describe how your queries will be interactive using Streamlit user interface elements. (Time allowed: 4 days.)

Phase 2. Build. Build the solution in a well-documented and structured Python program. (Time allowed: 1 week)

Phase 3. Present and Review. Create a five-minute video (if attending class synchronously is impossible) or present in class. Students watched each other's presentations, and evaluated them using an online form, based on perceived complexity (compared to their own projects), the student's ability to explain their code, what they liked the best about the project, and suggestions for improvement. (Due with project submission.) The instructor needed to approve all proposals before the implementation phase, to ensure they were of adequate complexity.)

Incorporating Streamlit widgets enabled students to create a user interface enabling interactive queries. For example, a user might interact with a slider to specify a maximum rental price and a dropdown list to select a neighborhood. The display shows on a map all homes in that neighborhood whose rental price is below the specified price. As the slider updates, the results update automatically, as shown in Appendix 1, Figures 1 and 2.

If presented in class, students reviewed the presentations of their peers. Involving students in the direct assessment of their classmates' project required students to compare the quality and complexity of their solutions with those of their peers. Asking students to provide praise and constructive recommendations placed them in the role of being active listeners, and the quality of the feedback they provided in written comments to their classmates contributed to their overall project grades.

Students compared their solutions with those of their peers and noting innovations such as, "The Map showing the available listings connected to the user's input was very nice." A student commented that they liked how a classmate included photos of homes in different neighborhoods, "which gave the app a more visual appeal." "It makes it seem like a real website." Students also offered constructive suggestions for improvement, suggesting, "Maybe you could connect the histogram with the data that you filter at the beginning. That way [we] could see the range of prices for each neighborhood" and "The maximum price slider only goes up to 499 but there are listings that are left out because they are more expensive than that."

ISA 330 Project

The "Twitter Project" is an individual project that teaches students how to interface with the Twitter

APIs and explore a dataset of their own choosing. Appendix 3, Figures 1 and 2 show screenshots of two sample student projects. Students completed the project in these phases:

Phase 1: Prepare. Students set up their own Twitter developer accounts. The class explored the Twitter APIs and discussed pulling historical Tweets versus setting up a Twitter listener. A sample notebook was shared that allowed students to "listen" for Tweets immediately after receiving their credentials from Twitter. By choosing a common hashtag or Twitter handle, students were able to see their listener program working before they exited the classroom. We used the handle @RealDonaldTrump due to the high volume of Tweets posted at that time referencing this handle.

Phase 2: Explore. The goal of the project was to compare Tweets on two different topics on Twitter. For example, they could choose "Nike" versus "Adidas", "Microsoft" versus "Apple", "Red Sox" versus "Yankees". They had two weeks to collect real time data using the Twitter streamer API. Once the student had an idea of a topic, they then used the Twitter website to explore the data. This helped the student confirm that the data they retrieved matched what they were interested in studying. In some cases, the students found their topic was too broad, while in other cases it was too narrow. Students refined their selection of handles and/or hashtags to get a dataset that represented the topic they wished to explore.

Phase 3. Acquire Data. Once the student had their targeted list of handles and/or hashtags, they set up a Twitter Listener by modifying the sample Jupyter Notebook. Students collected data and stored it locally on their laptops.

Phase 4: Analyze Data. During several weeks, students explored and shared with each other through informal class presentations, different ways to analyze the dataset. Students began their analysis performing a sentiment analysis on the Tweets making use of code shared with them to assist with this task. Students were encouraged to find examples of other techniques and to walk the whole class through the implementation of that technique. For example, one student shared the implementation of a word cloud using the text of the Tweets.

Phase 5: Reflect and Summarize. The final deliverable included an executive summary of their analysis along with their Jupyter Notebook.

5. STUDENT REFLECTIONS

While the PBL assignments in their respective courses were different, student reflections from both courses suggest that students shared common experiences while completing them. The four key aspects of PBL (ownership, collaboration, creativity, and critical thinking) (Stefanou et al., 2013) provide a reference for evaluating student reflections on their experiences completing these projects.

Ownership

Students in both courses commented on the value that a PBL assignment offered them to demonstrate the competencies and skills they learned in the course, which often exceeded their own expectations.

From CS 299 students:

"I liked that this final provided us with a concrete example of our own code that we can add to our portfolio."

"This was a difficult project but I appreciated the work."

"Our final exam/assignment was an excellent idea. In general, especially with classes like this, doing exams in this fashion is much better for us students. Firstly, it is a more accurate representation of the student's capabilities. This is because in an exam I may have studied something but forget it and lose marks, however, in real life, if I forget something then I can just look it up and apply it. Secondly, the whole experience of doing a final assignment such as the one we just did enables us to apply what we learned in class better. Whereas, in a final, students tend to just memorize things without understanding it sometimes. Ultimately, I am very grateful that we are allowed to have a take home exam because it gave us students the opportunity to demonstrate what we have learned in class. Also, it is much less stressful for us because we have more time to prepare and do the assignment."

From ISA 330 students:

"Since the students were allowed to pick to topic their project focused on, it allowed us to have ownership and creativity on the project."

Collaboration

Students commented on how completing this assignment offered an opportunity for collaboration during the development phase, and they recognized the value in sharing completed projects with their classmates:

From CS 299 students:

Students commented on sharing their work with classmates and commenting on their work:

One student offered to his classmate, "Maybe you could connect the histogram with the data that you filter at the beginning. That way we could see the range of prices for each neighborhood."

From ISA 330 students:

ISA 330 students commented on the sharing of ideas with their peers:

"We were welcomed to branch off of the given code and discover new findings and discuss them in class."

"I enjoyed exchanging findings with classmates and trading ideas about unique ways to use Python to develop results differently."

Critical Thinking
Students commented that completing the project developed their critical thinking skills as they dealt with real-world problems creating tangible work product that demonstrated their understanding of course concepts, one that can help them as they begin their professional careers:

From CS 299 students:

"I learned that I knew more than I thought I did and was able to apply for the most part by myself without running into too many issues."

"I learned that the error I got had nothing to do with my code but was really an error with the data."

From ISA 330 students:

"The location data is incredibly messy because it is inputted by the user and it required me to really figure out how to work with the data."

Creativity
Students commented that this project allowed them to express their own creativity in choosing how to design and present their results:

From CS 299 students:

"I learned that the possibilities with code are endless and my project barely breaks the surface for what I can do."

"I learned to apply what I've learned in the class by myself and realized what I've learned … can be used in a lot of ways to portray data."

From ISA 330 students:

"I liked that I could pick the companies I wanted to study. When I started looking at the data, I realized that I was way too general and needed to narrow my focus more."

6. SUMMARY AND DISCUSSION

The PBL aspects of ownership, collaboration critical thinking, and creativity, contribute to students meeting their learning outcomes and move students from a place of engagement to a position of empowerment, motivating them to create their own original work products. While CS 299 is an introductory course without any prerequisites and ISA 330 is an advanced programming course with one prerequisite, students in both classes benefited. This implies that a PBL approach is effective for students in both beginning and more advanced courses. Students performed within reasonable expectations given their prior experience (or lack thereof) because the assignments in each course were set with reasonable expectations given the student's backgrounds. One factor influencing the effectiveness of a PBL approach is creating open-ended assignments at the appropriate level that will both challenge students and enable them to meet with success.

Students chose how they wanted to analyze and present their data, resulting in a highly personal project. When finding sample Python code from other sources, students had to understand that code so they could adapt it to their project and explain it to others. In both projects, students had to own their work even when incorporating or adapting a framework or code examples found elsewhere.

They were able to either build a solution entirely from scratch, find examples of work done by others, or review code online and adapt that code to work for their project. They shared their results and offered feedback and critique of their classmates' projects. They saw how their solutions could bring about knowledge discovery.

Students had the flexibility to pursue any avenue of their choice to analyze their data. Each student's project was different, as they had to be creative in finding the most appropriate ways using charts, graphs, word clouds, maps, or other formats to convey their findings visually. For example, in ISA 330, some students created word clouds to show hashtag frequency, while others created bar charts to present the same information.

Both projects had students interact with real datasets. They had to debug their programs in ways that required them to think critically about the context of their data. For example, some students in CS 299 reported receiving runtime errors when they chose certain combinations of data to display. While they wrote their code to filter the data correctly, they neglected to check for empty results. For example, when analyzing AirBnB listings data, some students experienced runtime errors when they tried to plot hotel rooms in the Allston neighborhood of Boston. Because the Allston neighborhood has no hotel rooms and their programs did not check for this case, their programs crashed.

PBL assignments bring additional challenges and complexities for instructors introducing them in their classrooms. The project's problem needs to be open-ended enough to provide for a variety of interpretations and solutions, but not "so open-ended" that it becomes impossible for students to grasp. The project needs to be real and manageable, without feeling contrived. Instructors must keep track of what each student is doing, and what each student is capable of doing, and determine an accurate method to assess student projects. In CS 299, students completed a short online survey after each project presentation, asking them to compare the complexity of each student's project to that of their own. This provided a baseline for assessment to the instructor, using crowd sourcing to help identify the simpler and the more complex projects. Some ISA 330 students struggled because they were trying to learn new coding skills while at the same time trying to apply mathematical analysis. They needed more examples to understand better the data analytics techniques that were applicable to their individual learning scenarios.

Students knew that they were creating a work product that they would not only present to their classmates, but also one that could become part of their professional portfolios to demonstrate their Python coding skills at interviews with potential future employers. Designing a product whose potential audience extends beyond the classroom added to their level of engagement with the project and empowered many students to explore additional ways to query and visually share their data, beyond those required for the project. Presenting their projects in class prepared students to speak confidently about their work in a future interview situation. One limitation of this project is that the authors acknowledge that a control group with a course taught using traditional methods is not part of this study. Evidence of learning is based solely on outcomes of student work and student perception of the value that they received by completing their projects.

PBL assignments offer a valuable learning opportunity for students to create software applications that demonstrate their coding skills in a meaningful way. These projects provide students the opportunity to apply their coding skills and share their work directly with others outside of the classroom. Students develop and demonstrate their skills as they work through a project, interact with real world data, evaluate their own coding abilities, and review the work of their classmates. The assignments described in this paper show two different examples of how students can have a personalized, software development experience resulting in an original data-driven Python application that they designed, developed, and implemented entirely on their own.

7. REFERENCES

Aditomo, A., Goodyear, P., Bliuc, A.-M., & Ellis, R. A. (2013). Inquiry-based learning in higher education: Principal forms, educational objectives, and disciplinary variations. *Studies in Higher Education*, *38*(9), 1239–1258. Routledge.

Astrachan, O., Hambrusch, S., Peckham, J., & Settle, A. (2009). The Present and Future of Computational Thinking. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, SIGCSE '09 (pp. 549–550). New York, NY, USA: ACM. Retrieved September 14, 2018, from http://doi.acm.org/10.1145/1508865.1509053

Baugh, J. (2011). Make it relevant and they just may learn it. *Information Systems Education Journal*, *9*(7), 14.

Bell, T., & Lodi, M. (2019). Constructing Computational Thinking Without Using

Computers. *Constructivist foundations*, *14*(3), 342–351.

Bergmann, J., & Sams, A. (2014). Flipping for Mastery. *Educational Leadership*, *71*(4), 24–29.

Bloom, B. S. (1956). Taxonomy of educational objectives. Vol. 1: Cognitive domain. *New York: McKay*, *20*, 24.

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating Project-Based Learning: Sustaining the Doing, Supporting the Learning. *Educational Psychologist*, *26*(3–4), 369–398.

Brink, J., Capps, E., & Sutko, A. (2004). Student Exam Creation as a Learning Tool. *College Student Journal*, *38*(2).

Brown, I. W. (1991). To learn is to teach is to create the final exam. *College Teaching*, *39*(4), 150–153.

Jackson, C. (Kirabo), & Bruegmann, E. (2009). Teaching Students and Teaching Each Other: The Importance of Peer Learning for Teachers. *Working Papers*. Retrieved from https://digitalcommons.ilr.cornell.edu/workingpapers/77

Jones, Jennifer. (2016). The Student-Developed Quiz (or Exam): Scaffolding Higher-Order Thinking. *NACTA Journal*, *60*(2), 262.

McKay, Frank. (2017, October 25). 5 PBL Pitfalls to Avoid. *Edutopia*. Retrieved July 27, 2020, from https://www.edutopia.org/article/5-pbl-pitfalls-avoid

Rice, M., & Shannon, L.-J. Y. (2016). Developing project based learning, integrated courses from two different colleges at an institution of higher education: An overview of the processes, challenges, and lessons learned. *Information Systems Education Journal*, *14*(3), 55.

Rich, P. J., & Hodges, C. B. (Eds.). (2017). *Emerging Research, Practice, and Policy on Computational Thinking*. Cham: Springer International Publishing. Retrieved January 21, 2018, from http://link.springer.com/10.1007/978-3-319-52691-1

Scherz, Z., & Polak, S. (1999). An organizer for project-based learning and instruction in computer science. *Proceedings of the 4th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education*, ITiCSE '99 (pp. 88–90). Cracow, Poland: Association for Computing Machinery. Retrieved July 9, 2020, from http://doi.org/10.1145/305786.305874

Sengupta, P., Dickes, A., & Farris, A. (2018). Toward a Phenomenology of Computational Thinking in STEM Education. *ArXiv:1801.09258 [physics]*. Retrieved January 5, 2019, from http://arxiv.org/abs/1801.09258

Sharkey, S., & Weimer, M. (2003). Learner-Centered Teaching: Five Key Changes to Practice. *Teaching Sociology*, *31*(2), 251.

Stefanou, C., Stolk, J. D., Prince, M., Chen, J. C., & Lord, S. M. (2013). Self-regulation and autonomy in problem- and project-based learning environments. *Active Learning in Higher Education*, *14*(2), 109–122. US: Sage Publications.

Treuille, A., Teixeira, T., & Kelly. (2020). Streamlit—The fastest way to create data apps. Retrieved July 22, 2020, from https://www.streamlit.io/

VanDeGrift, T. (2015). Supporting creativity and user interaction in CS 1 homework assignments. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education* (pp. 54–59).

Whittington, K. J. (2004). Infusing active learning into introductory programming courses. *Journal of Computing Sciences in Colleges*, *19*(5), 249–259.

Appendix 1. Assignments

Table 1. Assignments in CS 299

Assignment	Description	Concepts Introduced
Assignment #1	Calendar Calculations	Data Types and Calculations, Variables, Input, Output
Assignment #2	Laptop Configuration	Conditional Programming and Formatting Data, String Processing
Assignment #3	Mastermind Game	Loops, Lists, Strings, and Functions
Assignment #4	Numerology (calculating numerical values for words based on values of each letter)	Dictionaries, File Processing
Assignment #5	AirBnB Visualizations	Pandas DataFrames, MatPlotLib charts, StreamLit

Table 2. Assignments in ISA 330

Assignment	Description	Concepts Introduced
Assignment #1	Tools for Data Analysts	Markdown, Magic Commands, and Control Flow
Assignment #2	Representing Data	Arrays, Indexing, Slicing, Lists, and Dictionaries
Assignment #3	Programming with NumPy	Vectors, Matrix Algebra, Linear Regression
Assignment #4	Pandas Part 1	Columns/Row Manipulation, DataFrames, Loading Data, Indexing
Assignment #5	Pandas Part 2	DataFrame Operation, Sorting, Statistics, Plotting, Data Wrangling
Assignment #6	Scraping the Web	Twitter API
Assignment #7	Regression	Intro to Regression and K-Nearest Neighbor Classification

Appendix 2. Description of CS 299 Project and Examples of Student Work

CS 299 PROJECT DESCRIPTION

The second half of *CS299-1 Problem Solving with Coding* course covered these major topics:

- Lists and list comprehensions
- Dictionaries, keys, values, items, iterating
- Functions: passing parameters, returning values
- Text Files and CSV Files: reading, writing
- MatPlot Lib and various types of charts
- StreamLit.io for making interactive applications
- Pandas

Throughout the course we also talked about computational thinking ideas, and good practices for developing visualizations of data. Your final exam project is to write a Python program that shows your mastery of many of these coding concepts (and others, such as loops, strings, if statements, formatting, as needed) as you interact with data found a CSV file containing Airbnb listings from Boston. Download the Boston Airbnb listings CSV file. The data originates from http://insideairbnb.com/get-the-data.html (look for the listings.csv file for Boston).

Phase 1. Design. Due by Thursday April 30, before 12:00 pm EST
Develop two questions and two visualizations that you can create, based on this data. Examples include: What are the most expensive rentals? Is there a correlation between reviews and nightly prices? How many rentals are available in each neighborhood? You can see more sample visualizations and computations at http://insideairbnb.com/boston/. Be sure to describe how your queries will be interactive – what Streamlit user interface elements will you use? For example, you might use a slider to specify a rental price, and then a listing of homes with rental prices lower than that value. Submit your document. I will respond within 24 hours by email approving your proposed questions or making suggestions if they appear to be too complicated or too easy.

Phase 2. Build. Due by Thursday, May 5 at 7:59 AM EST (before our final exam begins)
Write a Python program to compute the answers to your questions and create the two different visualizations. Display the results using an interactive webpage coded with Streamlit.io. Place all UI controls in the left sidebar, and your visualizations in the main content area.
Your code should demonstrate mastery of these capabilities:

- At least one function that has two parameters and returns a value
- At least one function that does not return a value
- Creating and Accessing keys and values from a dictionary
- The statistics module functions (average, median, mode, etc.)
- Charts and Graphs (at least two different charts and graphs of different types, with custom legends, axis labels, tick marks, colors, other features), or map showing latitude and longitude
- User Interface and dashboard with StreamLit.io

Usual rules about writing "good" code apply:

- Make your code as modular and easy to follow as possible
- Include a docstring, comments, and meaningful variable names. If you did something "cool" in your code that you are incredibly proud of, please write a comment to point this out.
- If you referred to any online articles or other information beyond class examples, please be sure to list them as references in your code.
- Make sure the program runs and the output is correct.

Getting Help:

- This is a final exam, so please do not discuss your program with anyone other than me.
- You can ask tutors for assistance on related topics, but you cannot ask them to help debug the program you write for the final exam. For example, you can ask tutors to help review examples of how to create bar charts in Python (in general), but you cannot ask them to help you debug a bar chart you might create for this exam using the Airbnb data.

Phase 3. Present and Review.
In Class on Tuesday, May 5 at from 8:00 am to 10:00 am (During Scheduled Final Exam)
You will present your project for approximately 5 minutes during the final exam period, and your classmates will provide feedback to you in an online form. For students in other time zones, or who cannot attend the class session live, please create a short (fewer than 5 minutes) video in which you describe your code and show us how it runs. Upload the video (unlisted to YouTube) or to your cloud storage, and send me a link.

Grading:

- Design Proposal and Document – 10 %
- Code – 80%
- Presentation – 10 %

EXAMPLES OF STUDENT WORK

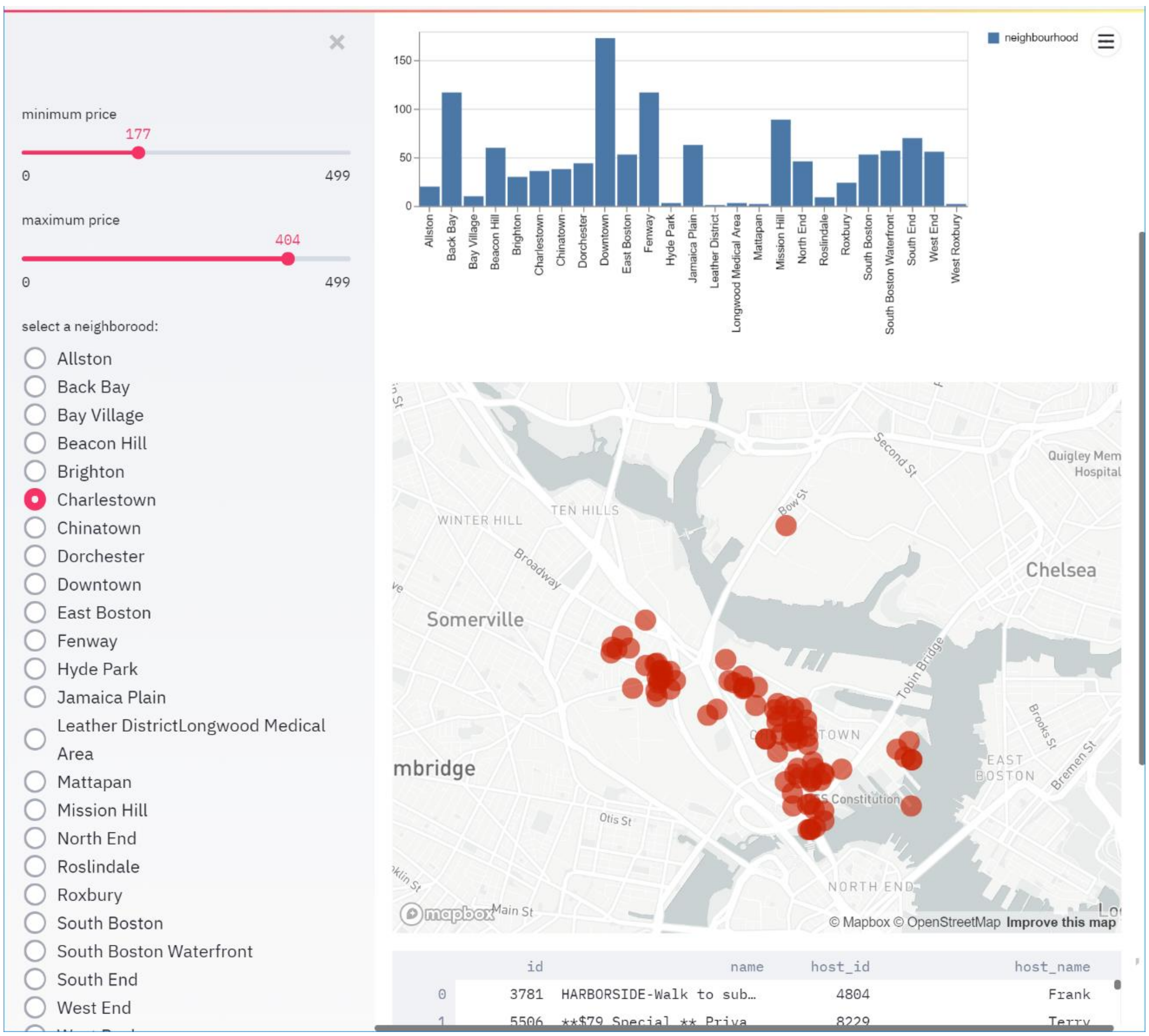

Figure 1. Chart and Map

In this visualization of Boston AirBnB listings data, a student chose to create a page displaying properties within a specified price range in a chosen neighborhood. Streamlit controls make the query interactive by allowing the user to select these values and then the chart and map update to reflect the new results from the query.

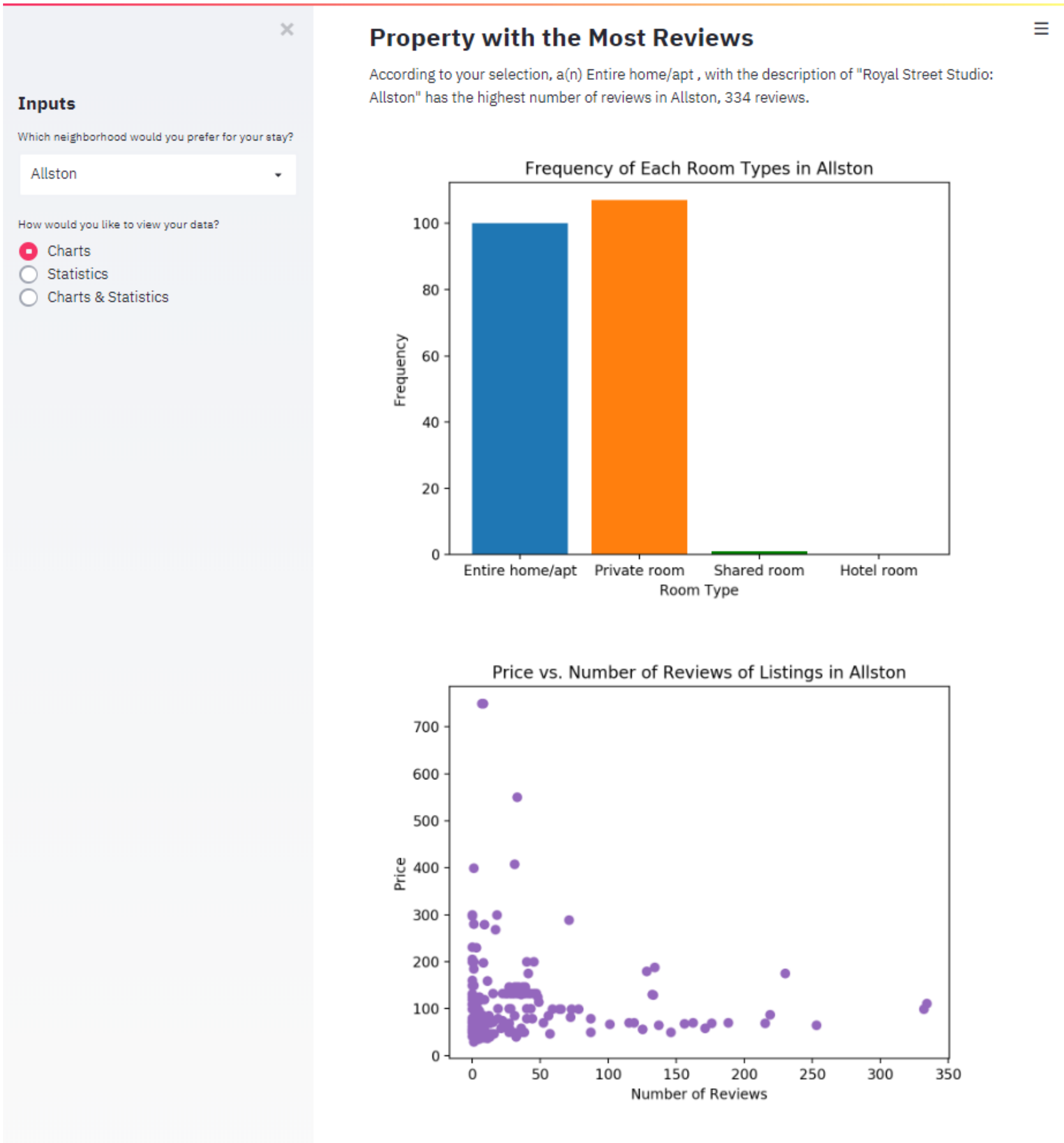

Figure 2. Bar chart and Scatter Plot

In this visualization of Boston AirBnB listings data, a student chose to create a page displaying information about properties with the largest number of reviews. Streamlit controls make the query interactive by allowing the user to select the neighborhood from a dropdown list and the desired output, and then the charts update to reflect the new results from the query.

Appendix 3. Description of IS 330 Project and Examples of Student Work

ISA 330 PROJECT DESCRIPTION

In this individual project, you will learn how to collect and compare social media data on two companies/products/topics of your choosing.

Phase 1: Set Up Account (In Class). You will each set up a developer accounts for Twitter (developer.twitter.com). Having a developer account allows you to gather a vast amount of Twitter data. Each of you will have your own account that you can use in future classes or even with your future employer. You will learn the different between the various APIs available to you and the difference between "pulling" historical data and "listening" for real-time data. By the end of this phase, you will have an active Twitter account gathering data.

Phase 2: Topic Identification (Both In-Class and Outside of Class). Once you have your Twitter account set up you will decide on a topic to examine. These topics could be two different companies (Microsoft versus Apple, Adidas versus Nike, etc.), two different products (Doritos versus Fritos, Corvette versus Mustang, etc.), or even two different topics (Black Lives Matter versus Blue Lives Matter, Pro-Choice versus Pro-Life, etc.). You will start (in class) by using the Twitter front end to explore your topic and make sure your hashtags are appropriate for data gathering. You will want a set of hashtags that are both appropriate for your topic (and without non-applicable data) as well as popular enough so you can gather several thousand Tweets over a several week period (i.e. #Trump and #Clinton would probably give you too much data while #ILoveGreenMarbles and #IHateGreenMarbles will not give you enough data). Make sure you confirm with the professor your topic before you start to acquire the data.

Phase 3. Acquire Data (Outside of class). Once you have your appropriate hashtags identified then you will run the Twitter streamer on your own laptops. It is important that you keep your laptop on and running during your period of study.

Phase 4: Analyze Data. We will start with a basic sample Jupyter Notebook and a dataset collected by the professor. You will segment the data into two different DataFrames and perform some basic text analysis on the Tweets. We will discuss how to find other sample code. You will explore different avenues of data exploration and share that knowledge with the class through mini informal presentations. You may share ideas and code with each other, but you are responsible to fully understand and customize any code you obtain from other sources (as well as cite that source in the notebook).

Phase 5: Reflect and Summarize. The final deliverable includes:

1. Executive summary. A 2-3 page summary of your findings. Please present this in such a way to be consumable by management (or other non-technical people).
2. Jupyter Notebook. Submit your fully documented Jupyter Notebook. It should contain enough comments throughout to walk another developer through your process. Make sure you also include lessons learned and any analysis that you did even if you did not include it in your executive analysis. As an analytics professional, you often may do exploratory work that has disappointing results, and it is important to document this so others learn from these trials.
3. Data. Submit your .JSON data file that you collected during the study. You can use this data file to rerun your entire Jupyter Notebook.
4. Presentation. Give a brief presentation to your fellow classmates. Imagine you are in a job interview and your interviewer asks you to discuss the project – this is your presentation. In about 5 minutes, walk through the notebook, and discuss your project.

Grading:

- Informal class presentations and participation: 20 %
- Fully Documented Code – 50%
- Executive Summary - 20%
- Presentation – 10 %

EXAMPLES OF STUDENT WORK

Using Jupyter Notebooks allowed students to create a work product that not only demonstrated their coding skills but also presented the findings in a single notebook. In this example, the student uses various techniques to explore the differences in text content of Twitter messages mentioning "Spotify" versus "Pandora", two popular music-streaming services.

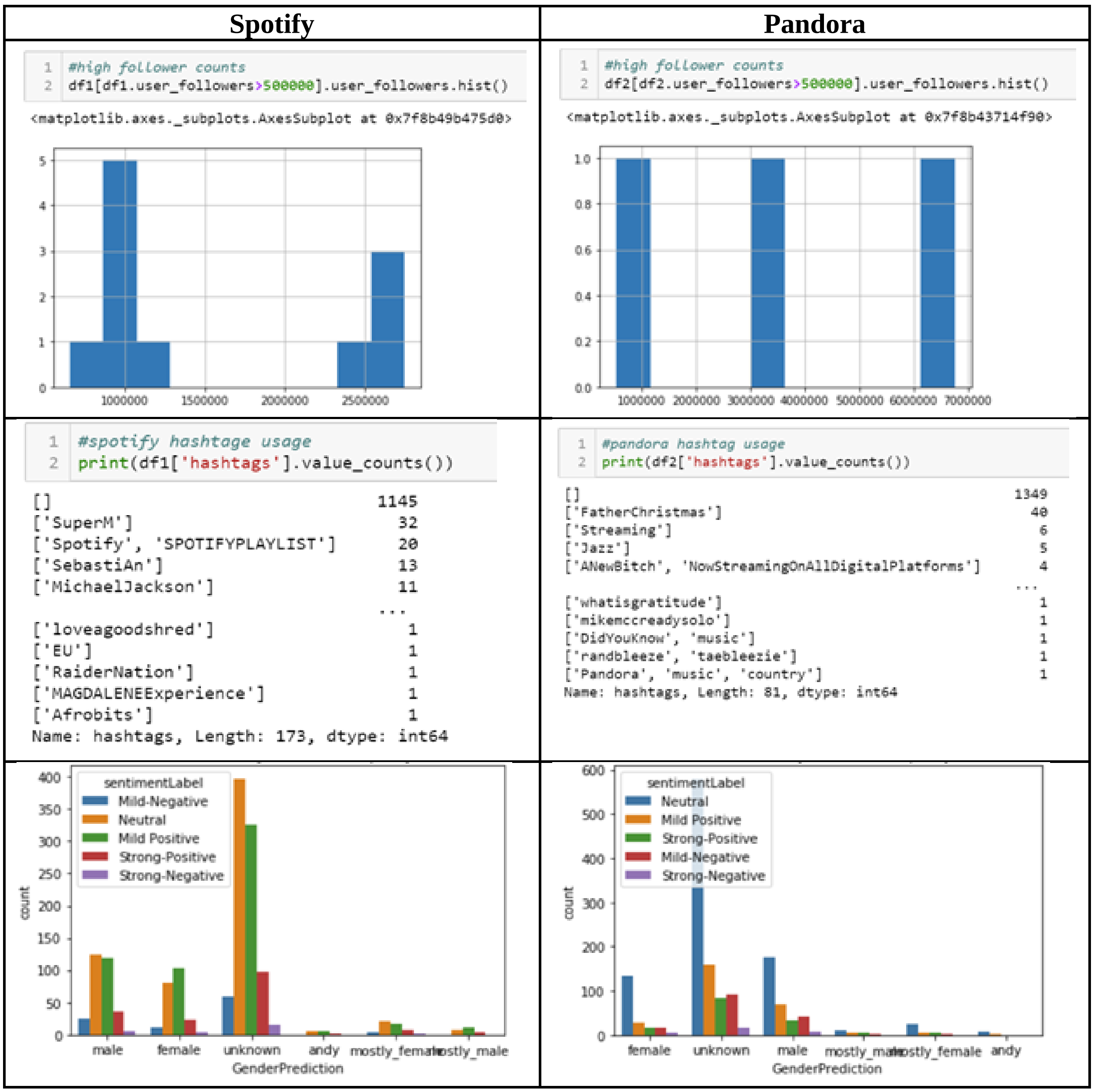

Figure 1. Charts comparing Tweets mentioning Spotify versus Pandora

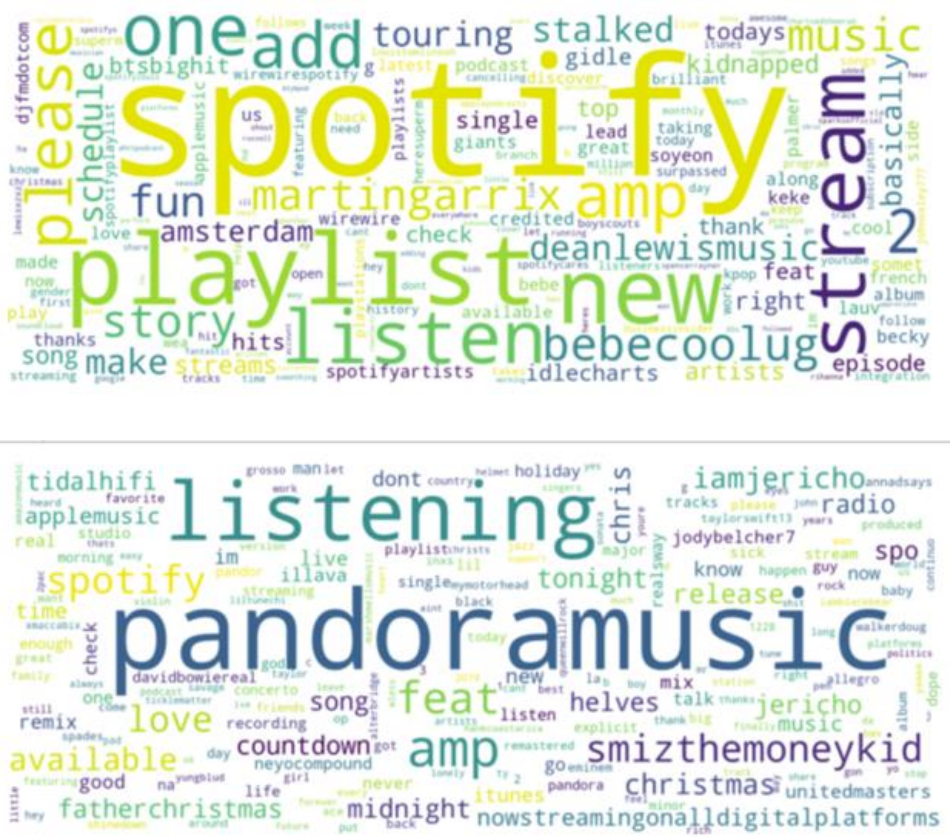

Figure 2. Word Clouds showing text data of Spotify versus Pandora.

Students enjoyed the quick visualization obtained using word clouds. Here the student was able to hone in quickly that playlists related to Spotify had more discussion on Twitter than playlists talking about Pandora, perhaps because Pandora playlists are a premium service. Word Clouds are popular among students and provide an easy technique for summarizing unstructured text data.