A case study in combining project-based learning and autograding in Machine Learning education

Dragan Vidaković[1[0000-0003-3983-7249]], Jelena Slivka[1[0000-0003-0351-1183]], Nikola Luburić[1[0000-0002-2436-7881]], Goran Savić[1[0000-0002-3917-5487]] and Aleksandar Kovačević[1[0000-0002-8342-9333]]

[1] University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia

Abstract. As many industries are in high demand for Machine Learning (ML) practitioners to solve business problems, it is essential to ensure that students know how to select adequate ML tools for given contexts and apply them adequately. To this aim, we designed a project-based undergraduate university ML course. The course utilizes a blended approach, in which students collaboratively work on real-world projects using an autograding platform for code-based assignments specially developed for the needs of the course. The course includes traditional lectures, discussions, reporting, and oral presentations. The course was evaluated using class assessment outcomes, faculty surveys, and observations. The results indicated that the blended learning approach was well-received and helped students better understand how to apply ML tools. They also suggest that project-based learning, in combination with an autograding platform and a blended approach, can be an effective way to teach undergraduate ML.

Keywords: Machine Learning, Education, Project-Based Learning, Autograding Platform, E-Learning.

1 Introduction

Machine Learning (ML) is a rapidly evolving field that has revolutionized industries and transformed various aspects of our lives. According to a report by McKinsey Global Institute [1], ML is projected to contribute up to $13 trillion to the global economy and create numerous new job opportunities by 2030. To adapt to these changes and meet industry demands, it is crucial for universities to incorporate ML education into their curricula.

Integrating ML into university courses equips students with essential skills in data analysis, informed decision-making, and an understanding of the ethical and social implications associated with these technologies [2]. However, ML is often perceived as more challenging than other areas of Computer Science (CS) due to its reliance on various mathematical and statistical concepts and the need to comprehend algorithmic complexities [3]. Consequently, innovative teaching methods are necessary to motivate and engage students in ML courses and ensure they acquire the skills required to effectively utilize this technology.

This study aims to develop a university-level ML course that offers students a comprehensive understanding of the field and prepares them for careers in Data Science and related domains. To achieve this objective, we have adopted a blended learning approach that incorporates collaborative project-based learning. Furthermore, we have leveraged an autograding platform specifically designed to meet the unique requirements of the course. Our focus is to empower students with the ability to apply learned concepts in real-world scenarios and effectively communicate and present their results.

As academic integrity is of utmost importance, we have implemented measures to monitor and prohibit plagiarism, safeguarding the educational process and maintaining the credibility of both students and the university. While previous studies have explored the development of ML courses, evaluations have often been limited. In this study, we aim to evaluate the effectiveness of our approach through class assessment outcomes, faculty surveys, and observations.

The rest of the paper is structured as follows. Section 2 provides the background of the research, contextualizing the need for an autograding platform in ML education. In Section 3, we present the methodology employed in the development and implementation of the course and the malepy platform. Section 4 discusses the course structure, including theoretical lectures, practical assignments, and the course project. Additionally, a detailed overview of the malepy platform is provided, encompassing its design, features, and functionalities. The results and findings of the evaluation are presented in Section 5, showcasing the impact of the platform on student performance and learning outcomes. Finally, in Section 6, we provide a concise summary of the key findings and implications discussed throughout the paper.

2 Related work

Previous studies have emphasized the importance of project-based learning in ML education, as it fosters problem-solving skills, deepens understanding, and promotes effective collaboration. Raschka [4] incorporated project-based learning components into a Deep Learning (DL) course, which included face-to-face lectures, weekly quizzes, coding-based homework, a midterm exam, and a class project. However, the study lacked automated evaluation of coding-based assignments and relied solely on anonymous class survey for evaluation.

Goel and Joyner [5] incorporated principles from cognitive learning sciences in the development of a foundational course on Knowledge-Based AI. While students enjoyed the course compared to traditional approaches, it did not include class projects to provide hands-on experience.

The Model AI Assignments session [6] aimed to create adaptable and engaging AI assignments that could serve as the core learning experience. However, the evaluation of these assignments is yet to be conducted. Canziani [7] compiled best practices from years of teaching ML and DL courses, but the work lacks evaluation.

Project Deep.Teaching [3] provided practical exercises in various ML topics to increase student motivation and understanding. However, the exercises developed with

Jupyter notebooks [8] did not include automated grading, and the overall work requires evaluation.

Blended learning is an educational approach that combines traditional face-to-face instruction with technology-enhanced learning, which allows students to work at their own pace. This approach has been shown to enhance student engagement and learning outcomes in CS education [9]. Previous studies [10][11] on blended learning in ML education used Kaggle [12] platform. These studies reported positive learning outcomes, but they also had limitations. Firstly, the blended approach was only used in a limited number of assignments. Secondly, the Kaggle platform is primarily oriented towards researchers and professionals and does not provide instructors with the ability to design custom challenges and offer flexible automatic evaluations. Additionally, Kaggle is a closed-form platform, which limits its use in education.

Open-source Kaggle alternatives focus on data and experiments exchange [13], bind users to use additional platforms for submissions and don't provide customization [14], orientate towards research and professionals [15][16] and thus are too complex and yet to be evaluated in the education domain. Like Kaggle, these platforms don't offer features like source code analysis or plagiarism detection.

Existing studies emphasize the need for a comprehensive approach that combines project-based learning, automated grading, and a customizable platform for ML education. However, a significant gap in the literature is the lack of comprehensive evaluations of such approaches. Our study aims to fill this gap. Through class assessment outcomes, surveys, and observations, we gather evidence of the benefits of and limitations of combining project-based learning, automated grading, and a customizable platform. This evaluation will provide insights into student engagement, learning outcomes, and critical skill development.

3 Methodology

We identified several essential requirements for equipping students with the necessary ML skills. Firstly, students should have a strong background in mathematical concepts critical for understanding ML algorithms [17]. Secondly, students should possess programming skills, including experience with Python [18] and its ML libraries [19]. Thirdly, students should learn data preprocessing techniques, such as handling missing values, normalizing data, and visualizing data distributions [20]. Fourthly, students should acquire the ability to select and evaluate models, comparing performance metrics and avoiding overfitting and underfitting [21]. Finally, students must be aware of ethical and social considerations in ML, including data privacy, algorithmic bias, and accountability [2].

Given the applied nature of ML, project-based learning is the cornerstone of our course. As student learning is enhanced through engagement with scientific practices [22], projects use datasets and challenges from ML research. Collaborative project-based learning is emphasized to address the complexity of these challenges, with a preference for groups of three students [23]. From-scratch implementation is encouraged to enhance idea implementation and experimentation efficiency [24]. Students are required to submit reports for each task, documenting their approach, experiments, and results, thereby demonstrating their understanding and creativity in solution development.

To facilitate the learning process, a custom platform was developed to host code-upload ML challenges, support autograding, provide a leaderboard, and enable source code analysis for plagiarism detection. The existence of the leaderboard offers students insights into the relative performance of their solution, motivating and stimulating further improvements [11]. This platform was developed as an alternative to existing options like Kaggle, which were found to be inadequate for our educational objectives.

By following this methodology, we aim to equip students with the necessary ML skills through a comprehensive curriculum, practical project-based learning, and the support of a customized learning platform. The evaluation of student outcomes, including engagement, learning effectiveness, and critical skill development, will provide valuable insights into the efficacy of this approach.

4 Implementation

In this section, we explore the implementation of the ML course and the malepy autograding platform. We discuss the course structure, including lectures, assignments, and the project. Additionally, we provide an overview of the platform, highlighting its design and key features. The section offers insights into the practical execution of the course and the integration of the autograding platform, emphasizing the factors that contribute to its effectiveness and student engagement.

4.1 Course structure

Our course is taught in the fourth year of undergraduate studies and is structured into theoretical lectures and practical assignments. The theoretical part consists of 15 face-to-face lectures, covering fundamental concepts of statistical ML. The lectures encompass various topics, including regression, regularization, linear classifiers, ensemble models, clustering, dimensionality reduction, and self-supervised learning. Table 1 contains a comprehensive list of lecture topics.

The practical component of the course consists of six code-based homework assignments and a culminating course project. These assignments offer students valuable hands-on experience in applying the concepts and techniques learned during the course. We emphasize a collaborative project-based learning approach, where students form teams of three at the beginning of the course, as depicted in Figure 1. This team formation allows for study groups and encourages collaborative learning. The course project serves as the capstone experience, enabling students to tackle a real-world ML problem of their choice and apply their skills in a comprehensive manner.

Table 1. Lecture topics

Number	Topic
1	Introductory Lecture
2	Simple Linear Regression
3	Multiple Linear Regression
4	Regularization
5	Regularization (continued) and Nonparametric Approach
6	Maximum Likelihood Method
7	Linear Models for Classification (Logistic Regression and Perceptron)
8	Linear Models for Classification (Naïve Bayes and Practical Applications)
9	Support Vector Machines
10	Ensemble Models
11	Clustering
12	Dimensionality Reduction
13	Practical Tips for ML Application and 3 Learning Principles
14	Semi-Supervised Learning
15	Theoretical Foundations of Supervised Learning

Homework Assignments. The homework assignments are designed to simulate real-world problems and are introduced during the auditorial exercises. Each assignment includes a training dataset and a hidden test dataset, along with an evaluation metric. To successfully complete an assignment, students must achieve a predefined acceptance criterion, which represents the minimum evaluation metric value their model needs to surpass when evaluated on the hidden test dataset. The assignments encompass various areas of ML, such as regression, classification, clustering, and dimensionality reduction, and involve the application of diverse data preprocessing techniques. The following is a summary of homework assignments:

1. Simple Linear Regression problem on the dataset that contains outliers
2. Multiple Regression problem on the dataset that contains outliers and categorical data
3. Binary text classification problem on the dataset that contains outliers and that must be solved using Support Vector Machine model
4. Multiclass classification problem on the imbalance dataset that contains outliers, categorical data, and missing values, and must be solved using ensemble methods
5. Clustering problem that must be solved using Gaussian Mixture model on the dataset that contains outliers, categorical data, and missing values
6. Multiclass classification problem on the imbalance dataset that contains outliers, categorical data, and missing values, and must be solved using Principal Component Analysis for dimensionality reduction.

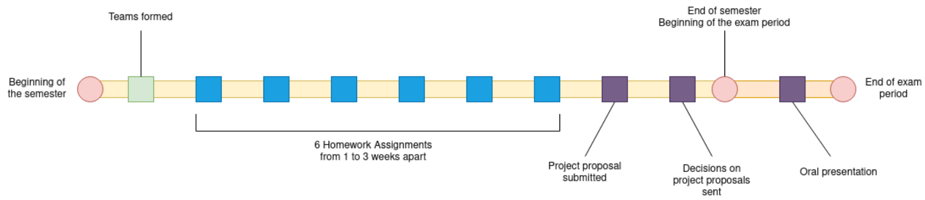

Fig. 1. Summary and timeline of the course through the semester

For each assignment they complete, students are required to write a comprehensive report that includes detailed descriptions of their approach, feature and algorithm exploration, experimental procedures, and observations of the results. The purpose of the report is to demonstrate the students' understanding of the problem, methodologies used, and the outcomes, while also highlighting their creativity in finding solutions.

During the following auditorial exercises, the best-performing team from each group presents their solution orally and responds to questions from the instructor and other students. The instructor then provides a recap of the assignment, highlighting notable aspects and providing additional insights into the solution. Furthermore, the instructor selects an additional team to present their solution and answer questions. Teams that exhibit outstanding performance, produce high-quality reports, and deliver excellent oral presentations for homework assignments are rewarded by being exempted from working on the course project.

Course Project. The course project is optional and provides an opportunity for teams that were not satisfied with their performance on the homework assignments to win some additional points, or for those who wish to apply their newly acquired skills to a real-world problem of their choice. The course project involves defining the problem statement, including the process of finding or collecting the dataset, conducting experiments, and writing a report following a predefined template.

When writing a project proposal, students are required to address the components outlined in Table 2. Although some teams found the conceptualization of the project to be more challenging than others, we have not encountered any cases where students were unable to find a project they were interested in working on. The projects undertaken by the students were highly diverse, with a significant number of them involving new data collection.

The project report follows a defined template and includes sections such as Motivation, Research Questions, Related Work, Methodology, and Discussion. To ensure realistic writing and review efforts, students are required to limit their reports to 4 pages, including references. During the exam period, students present their projects, with presentation lengths capped at 15 minutes.

Table 2. Components of the course project proposal

Component	Description
Project Topic	The topic or theme of the project
Problem Statement	A brief description of the problem being addressed by the project
Data	Reference to the existing dataset or briefly described the data collection procedure
Methodology	A brief outline of the methodology or approach to solving the problem
Evaluation	A brief description of the evaluation process, including the metrics that will be used to assess the solution

4.2 Platform

The malepy[1] platform (an acronym derived from **ma**chine **le**arning in **py**thon) was developed to support the practical component of the ML course. It serves as an autograding platform, designed using the Python programming language and the Django web framework [25]. The platform provides an interactive and user-friendly environment for instructors and students to engage in hands-on assignments. Figure 2 illustrates the conceptual diagram of the platform, showcasing the instructors' and students' interaction with the platform.

The development of the malepy platform was motivated by the unique needs and requirements of the ML course. Its purpose is to provide a seamless and efficient experience for students as they complete their homework assignments and gain practical experience in ML. Throughout the development process, careful attention was given to creating an intuitive user interface that allows instructors and students to navigate the platform effortlessly.

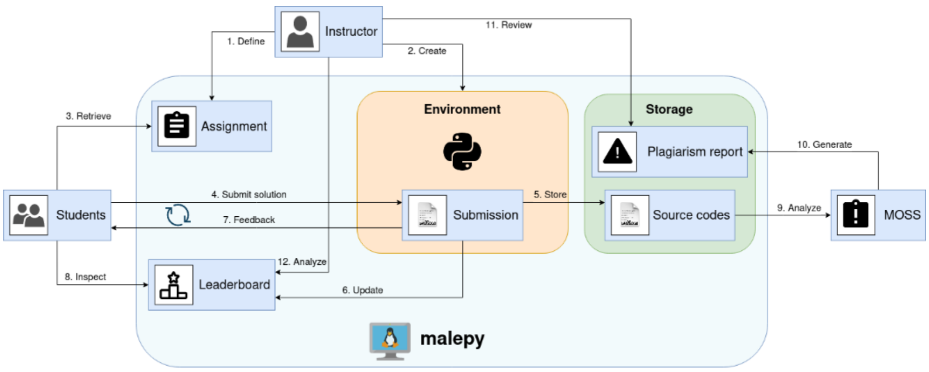

Fig. 2. Conceptual diagram of instructors' and students' interaction with malepy

[1] https://github.com/vdragan1993/malepy-platform/

Instructors play a crucial role in the platform's functionality. Besides managing courses and teams, they create assignments by providing a textual definition, evaluation metric, acceptance criteria, starting and ending dates, and datasets. Students are given access to the training dataset and a preview of the test dataset for each assignment. The solution development takes place using the provided training dataset, while the test set preview allows students to familiarize themselves with the test dataset's structure.

Once students have developed their solutions, they submit their source code, which is then executed within the instructor-defined environment. The platform utilizes Python's virtualenv [26] to create a virtual environment for solution execution. Executions within this environment are managed using Python's core subprocess module. Instructors can specify the maximum number of submissions per team and set the allowed time for solution execution. If the solution is successfully executed, students can view their results and see their position on the updated leaderboard.

To ensure academic integrity, the platform incorporates measures for plagiarism detection. Instructors have the ability to inspect all submissions and run source code analysis using MOSS [27]. In cases where irregularities are found, instructors can disapprove submissions. At the end of the assignment period, students gain access to the complete test dataset and can review the source code of other teams.

The malepy platform not only supports the practical component of the ML course but also enhances the learning experience by integrating with the course materials and curriculum. It provides a hands-on approach that reinforces the concepts covered in the theoretical lectures. Running on the Gunicorn server [28] allows it to handle a large number of student submissions and execute their solutions efficiently within defined time constraints. Security and privacy measures are implemented to ensure students' code and data confidentiality and integrity.

5 Discussion

We evaluated the effectiveness of our approach through a comprehensive assessment that included class assessment outcomes, faculty surveys, and observations. The evaluation was conducted over six iterations of the course, with the first iteration held without the malepy platform and the subsequent five iterations incorporating the platform. Throughout all iterations, the grading scale remained consistent, with the practical part accounting for a maximum of 60 points and the oral exam accounting for 40 points. Grades were in the range from 5 (falling grade) to 10 (excellent – outstanding).

Analysis of the assessment data revealed notable differences between students who utilized the malepy platform and those who did not. As shown in Table 3, students who actively engaged with the autograding platform demonstrated higher average scores on homework assignments and the theoretical exam, resulting in higher overall average grades. These findings indicate that the platform positively influenced student performance and learning outcomes.

Table 3. Class assessment outcomes evaluation

Platform	Students	Homework points	Oral exam points	Average grade
No	47	45	24.39	8.02
Yes	**73.6**	**47.53**	**33.43**	**8.95**

In addition to quantitative assessments, we sought qualitative feedback through anonymous class surveys administered by the faculty. The responses collected, as shown in Table 4, were overwhelmingly positive. In textual comments, students praised the course concept, engaging lectures, and the organization of the practical part, particularly highlighting the effectiveness of the homework assignments.

Table 4. Results of the anonymous class survey conducted by the faculty

Platform	Students	Rating
No	5	9.20
Yes	**23**	**9.75**

While quantitative data and surveys provided valuable insights, personal communication with students and the instructors' observations further supported the effectiveness of our approach. Collaborative project-based learning fostered a supportive and interactive learning environment, enabling students to delve deeper into the subject matter, engage in meaningful interactions, and showcase their creativity in solving ML problems. Notably, students who demonstrated creativity in their feature engineering efforts appeared to achieve higher rankings on the platform's leaderboard. This correlation reinforces the significance of hands-on learning and the positive impact of a supportive learning environment on student achievement.

Despite the positive outcomes, it is essential to acknowledge certain limitations of our evaluation. For instance, the study's sample size and context were limited to a specific course and student population, which may affect the generalizability of the findings. Furthermore, the evaluation focused primarily on the short-term impact of the malepy platform on student performance, and long-term effects warrant further investigation.

6 Conclusion

The malepy platform has significantly enhanced our ML course, as evidenced by our evaluation involving assessments, surveys, and observations. Students who utilized the platform demonstrated improved performance on homework assignments and the theoretical exam, indicating its positive impact on learning outcomes.

Collaborative project-based learning, facilitated by the platform, fostered a supportive and interactive environment where students could deepen their understanding of ML concepts and showcase their creativity. The positive feedback from students underscores the effectiveness of hands-on learning experiences, particularly the homework assignments.

While acknowledging the limitations of our evaluation, such as the specific context and sample size, we are confident in the platform's ability to empower students and nurture their passion for ML. Going forward, we will leverage the insights gained to refine the course structure and platform features, such as code quality analysis and potential integration with other tools to further enhance the learning experience.

Acknowledgements

This research was supported by the Science Fund of the Republic of Serbia, Grant No 6521051, AI – Clean CaDET and the Ministry of Science, Technological Development and Innovation through project no. 451-03-47/2023-01/200156 "Innovative scientific and artistic research from the FTS (activity) domain".

References

1. Manyika, J., Chui, M., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H.: Harnessing automation for a future that works. In: McKinsey Global Institute, New York, USA (2017).
2. Bostrom, N., Yudowsky, E.: Ethics and artificial intelligence. AI Magazine 35(4), 105-117 (2014).
3. Hertha, C., Voigt, B., Baumann, P., Strohmenger, K., Jansen, C., Fischer, O., Zhang, G., Hufnagel, P.: Deep Teaching: Materials for Teaching Machine and Deep Learning. In: HEAd' 19 5th International Conference on Higher Education Advances, pp. 1153-1131. Editorial Universitat Politecnica de Valencia, Spain (2019).
4. Raschka, S.: Deeper Learning By Doing: Integrating Hands-On Research Projects Into a Machine Learning Course. In: Proceedings of the Second Teaching Machine Learning and Artificial Intelligence Workshop, pp. 46-50. PMLR, (2022).
5. Goel, A.K., Joyner, D.A.: Using AI to Teach AI: Lessons from an Online AI Class. AI Magazine 38(2), 48-59 (2017).
6. Neller, T.W., Sooriamurthi, R., Guerzhoy, M., Zhang, L., Talaga, P., Archibald, C., Summerville, A., Osborn, J., Resnick, C., Oliver, A., Bhupatiraju, S., Agrawal, K. K., Derbinsky, N., Strange, E., Neumann, M., Chen, J., Christensen, Z., Wollowski, M., Youngquist, O..: Model AI Assignments. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 9751-9753. (2019)
7. Canziani, A.: Teaching Deep Learning, a boisterous ever-evolving field. In: Proceedings of the Second Teaching Machine Learning and Artificial Intelligence Workshop, pp. 30-34. PMLR, (2022).
8. Kluyver, T., Ragan-Kelley, B., Perez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C.: Jupyter Notebooks – a publishing format for reproducible computational workflows. In: Loizides, F., Schmidt, B. (eds.) Positioning and Power in Academic Publishing: Players, Agents and Agendas, pp. 87-90. (2016).
9. Garrison, D.R., Kanuka, H.: Blended learning: Uncovering its transformative potential in higher education. The Internet and Higher Education 7(2), 95-105 (2004).
10. Verspoor, K., Nicholson, J.: E-learning with Kaggle in class: Adapting the ALTA shared taks 2013 to a class project. In: Proceedings of the Australasian Language Technology Association Workshop 2013, pp. 142-145. ALTA, (2013).

11. Chow, W.A.: Pedagogy that Uses a Kaggle Comptetion for Teaching Machine Learning: an Experience Sharing. In: 2019 IEEE International Conference on Engineering, Technology and Education, pp. 1-5. TALE, (2019).
12. Kaggle, https://www.kaggle.com/, last accessed 2023/05/27.
13. Vanschoren, J., van Rijn, J. N., Bischi, B., Torgo, L.: OpenML: networked science in machine learning. ACM SIGKDD Explorations Newsletter 15(2), 49-60 (2014).
14. AIcrowd, https://www.aicrowd.com/, last accessed 2023/05/27.
15. Pavao, A., Guyon, I., Letournel, A.C., Baro, X., Escalante, H., Escalera, S., Thomas, T., Xu, Z.: CodaLab Comptetitions: An open source platform to organize scientific challenges. PhD Thesis. Universite Paris-Saclay, France (2022).
16. Yadav, D., Jain, R., Agrawal, H., Chattopadhyay, P., Singh, T., Jain, A., Singh, S. B., Lee, S., Batra, D.: EvalAI: Towards Better Evaluation Systems for AI Agents. arXiv preprint arXiv:1902.03570, (2019).
17. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, (2016).
18. Van Rossum, G., Drake Jr, F.L.: Python reference manual. Centrum voor Wiskunde en Informatica Amsterdam, (1995).
19. Raschka, S.: Python machine learning. Packt publishing ltd, (2015).
20. Witten, I., Frank, E., Hall, M. A., Pal, C. J.: Data mining: Practical machine learning tools and techniques. Morgan Kaufman Publishers, (2016).
21. Alpaydin, E.: Introduction to machine learning. MIT Press, (2010).
22. Edelson, D.: Realising Authentic Science Learning Through the Adaptation of Science Practice. International Handbook of Science Education, 317-331 (1998).
23. Apedoe, X. S., Ellefson, M.R., Schunn, C. D.: Learning together while designing: Does group size make a difference? Journal of Science Education and Technology 21(1), 83-94 (2012).
24. Scliendorfer, A., Gajek, C., Reif, W.: Turning software engineers into machine learning engineers. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, pp. 36-41. PMLR, (2021).
25. Django: The web framework for perfectionists with deadlines, https://www.djangoproject.com/, last accessed 2023/05/27.
26. virtualenv, https://virtualenv.pypa.io/en/latest/, last accessed 2023/05/27.
27. Schleimer, S., Wilkerson, D.S., Aiken, A.: Winnowing: local algorithms for document fingerprinting. In: Proceedings of the 2003 ACM SIGMOID international conference on Management of data, pp. 76-85. (2003).
28. Gunicorn – Python WSGI HTTP Server for UNIX, https://gunicorn.org/, last accessed 2023/05/27.