Intelligent Decision Support System

Practical Work 3

"Educational Success Predictor"

Gustavo Rayo

Umberto Salviati

Giacomo Sanguin

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Barcelona, January 9, 2024

Contents

1 Abstract

This project presents the development of an Intelligent Decision Support System designed to predict students' academic outcomes based on a comprehensive dataset. Implemented as a web application using, the IDSS combines traditional student features with an expert opinion component, allowing users to input responses to ten interview-like questions, scored by an expert. The system offers model customization, enabling users to choose between basic and advanced configurations. This approach aims to enhance the accuracy of predictions and provide educational institutions with a valuable tool for student support and academic planning.

2 Domain of application

The domain of application for this Intelligent Decision Support System is situated within the critical sphere of education, specifically within the context of predicting and optimizing student academic outcomes. In today's dynamic educational landscape, institutions face the challenge of understanding and addressing the diverse factors that influence students' success or potential dropout. Our IDSS addresses this challenge by leveraging a comprehensive dataset containing a large number of features, including demographic information, academic history, and socio-economic factors. This rich data repository serves as the foundation for predictive modeling, enabling the system to generate insights into students' academic trajectories.

One distinctive aspect of our prototype, is the incoporation of an expert opinion component, simulating a college interview scenario. Users are prompted to respond to ten questions, each evaluated and scored by an expert (typically a professor or experienced academic figure). The fusion of traditional student features with expert insights enhances the IDSS's predictive accuracy and provides a more accurate understanding of the factors influencing student success.

The domain of application extends to various stakeholders within educational institutions, including academic advisors, administrators, and educators. This IDSS serves as a valuable tool for these professionals, offering a proactive approach to identifying at-risk students early in their academic journey. By adjusting support plans according to the predictions, institutions can step in with specific help, offering personalized assistance to students dealing with possible difficulties. Furthermore, it contributes to optimizing resource allocation, making sure to create a positive and good learning environment.

3 Main identified decisions

The system is a valuable tool for higher educational institutions. It could help in multiple decisions during admission and throughout the student's academic journey.

- The system could help determine the students who are more likely to complete the university. On the basis of the results, the university could make a more informed decision on accepting or denying an application. It could also serve to identify those students accepted but are at risk of dropping out. With this information, the university could provide special guidance to those students.
  Furthermore, it enables institutions to proactively identify students at risk of academic challenges early in their academic journey, facilitating timely intervention and support.

- As the system supports multiple models that include academic and financial features, it could help in the decision-making process of awarding scholarships to those students that are likely to drop out due to financial constraints.

- During the academic years, the system would allow professors to make updated predictions based on the results obtained in the first and second semesters. The result could serve to encourage the university to remain proactive and optimize the learning experience.

4 Functional architecture of the IDSS prototype

The architecture was designed to be modular and extensible. It provides easy updates of the components and facilitates concurrent development. It consists of three main components as shown in Figure 1.

- The first component is the user interface. It is implemented in React, a Javascript library for developing user interfaces. The development of an application from scratch is time-consuming and unnecessary, so we started the development using a template called *material-kit-react* that includes some components based on Material Design, a design language developed by Google. The interface contains three main elements. A form where the professor can select a model to predict the results. Depending on the model selected, the corresponding fields are displayed. The second element is a survey where the professor can assign a value for each of the questions. The last element is the result view where the model prediction is displayed, along with the professor evaluation and the joined prediction.

- The second component is the API. It consists of a simple REST API developed in Python. The primary library for the development was FastAPI, which provides high performance and fast development. We have a single endpoint where the application sends the information captured in the form and the survey. Depending on the selected model, the server decides what model to use. Additionally, it processes the survey information and does the calculation of the final result based on the prediction of the model and the result of the survey.

- The third component is the predictor. There are 7 ensemble predictors composed of a Logistic Regression, a Random Forest and a Support vector machine each. They handle different sets of features and the final result is based on a majority voting method.

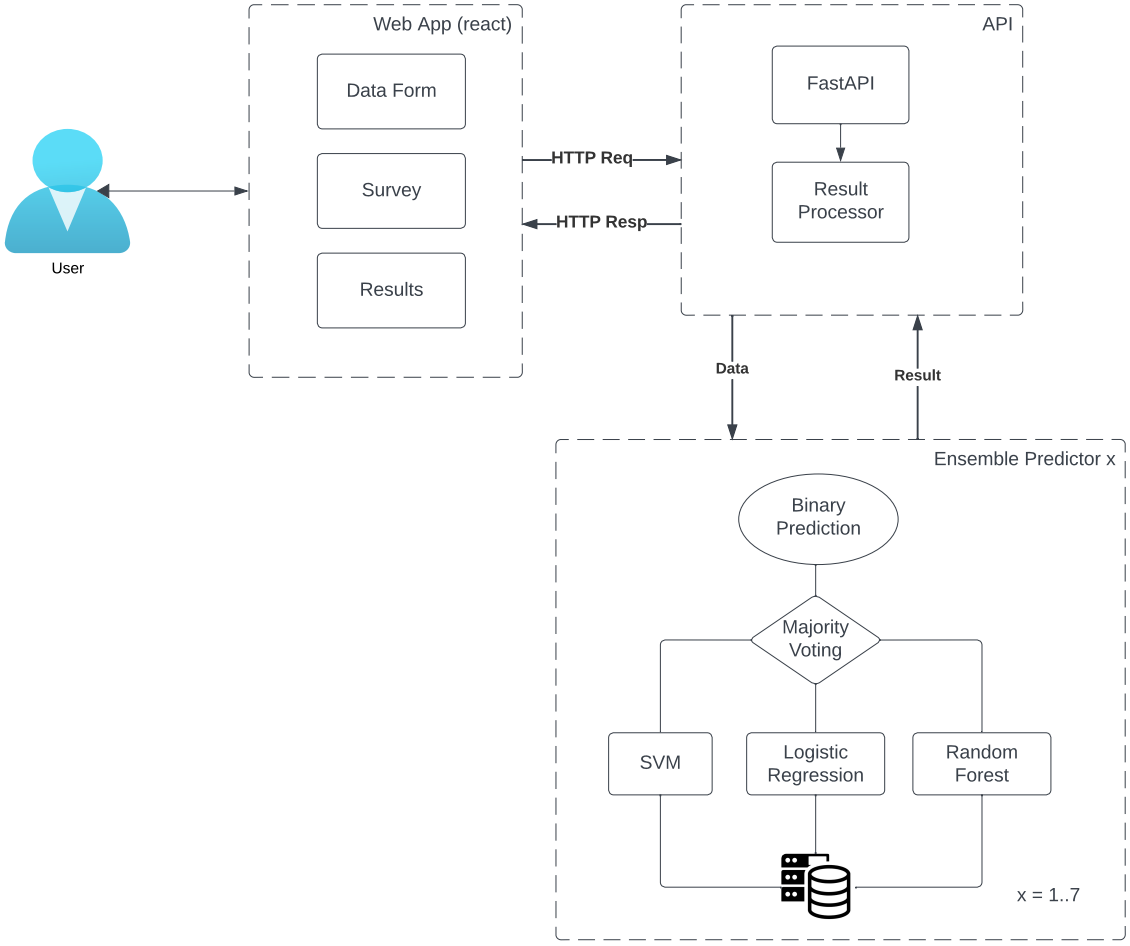

Figure 1: Functional Architecture

All the components were deployed in a virtual machine in Microsoft Azure. The web application was deployed in Apache server while the backend was deployed using Uvicorn, an ASGI (Asynchronous Server Gateway Interface) web server implementation for Python.

5 Data pre-processing summary

We tackled data pre-processing by first conducting a thorough analysis. Thanks to Python and the plotting (`matplotlib`) and statistical visualization (`seaborn`) libraries, we were able to effectively visualize the data. However, the data presented some imperfections for our intended purpose. Indeed, within the data collection, there were details about students that were irrelevant to our analysis, as we could not ascertain whether they would graduate or drop out.

This approach diverges from analyses found on Kaggle, where some consider students as non-dropouts from the start. Our decision to exclude such data was motivated by the need to focus on more certain information. The main phase of our pre-processing, therefore, involved filtering the data, retaining only those with the `"graduate"` or `"dropout"` status in the target variable.

Another distinctive step in our analyses was balancing the data. Observing the data distribution, we noticed an imbalance (Fig. 2), with a significantly larger number of dropouts compared to graduates. We explored various balancing techniques, including undersampling and oversampling. Among these, oversampling with the SMOTE technique proved to be the best, generating models that, albeit marginally, outperformed those published on Kaggle.

As a result, we fed our models with a dataset balanced at 50% between graduates and dropouts, providing a fairer basis for our analysis.

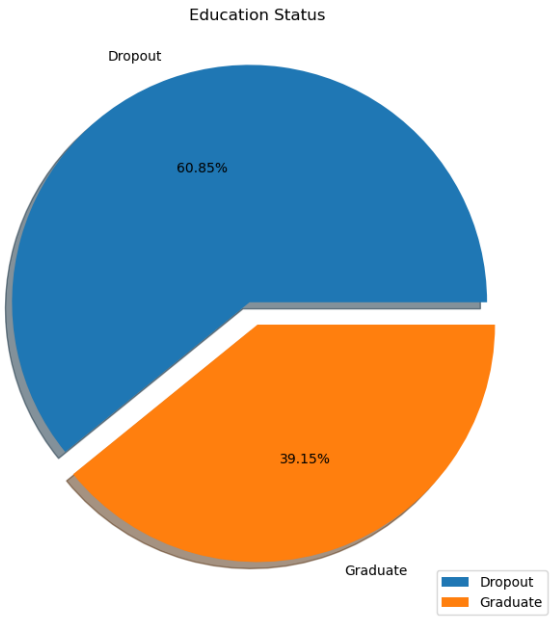

Figure 2: Pie Chart of the imbalanced data

6 Flowchart of the data-driven IDSS model/s gathering

To train my data-driven models, I initially experimented with various algorithms, including the three selected ones: Random Forest, Logistic Regression, and SVM (with multiple kernel variations). Additionally, I explored KNN and Gaussian Naive Bayes. On a subset of features, where the distributions seemed to follow a normal pattern (refer to the Python notebook where I plotted feature distributions), I attempted to apply Linear Discriminant Analysis.

The pivotal decision was to select three models from the trained set. I chose the top three models, ensuring they had distinct functional domains. It made little sense to include another linear regression akin to logistic regression or an SVM with a linear kernel, as they would share a similar domain and potentially override the Random Forest model – which performed exceptionally well among the trained models, along with Logistic Regression.

Therefore, the selection of these three models was deliberate, aiming for diversity in functional domains. This careful curation gives us confidence in our results, as these models outperformed some never-before-trained models on this dataset. Further details on their performance will be explored in the validation section.

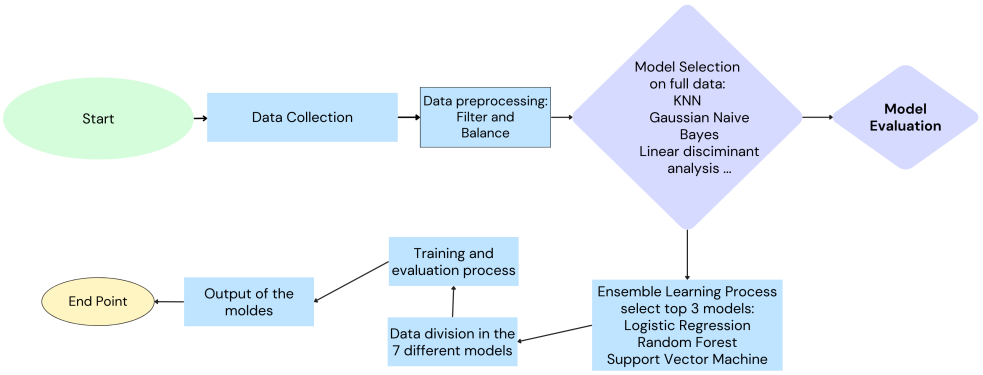

Figure 3: Chart of the Training of the Models

As we can se in the image (fig. 3), our approach involved implementing ensemble learning, where we connected and leveraged three models: Logistic Regression, Random Forest, and SVM with an RBF kernel.

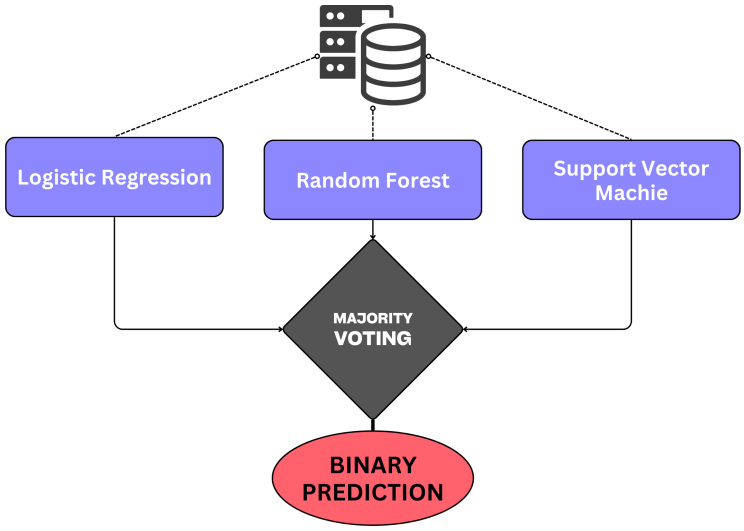

Figure 4: Structure of the prediction majority voting

7 Data post-processing and validation

The post-processing utilized in our case does not involve any data filtering or smoothing. In fact, the step designed to identify and rectify potential errors in the model outputs involves the integration with the expert-based model, namely the survey completed by the professor. It is important to note that the data from our data-driven model is not independently treated; rather, it is weighted differently based on the accuracy of the individual models.

It is crucial to emphasize that our data-driven model outputs are not considered independently accurate. Instead, they are assigned varying weights according to the accuracy of the respective models. The integration with the expert-based model serves as a corrective measure to enhance the overall reliability and precision of the final results. This collaborative approach, combining data-driven insights with expert knowledge, aims to provide a more robust and accurate analysis of the given scenario. In our case, model validation is crucial for the proper functioning of our Intelligent Decision Support System (IDSS). If we can enhance the performance through potential feedback, it not only improves the accuracy of our models but also influences the weight these models contribute to the final prediction in collaboration with the expert-based component.

We now will discuss the accuracy of our models, we can examine the various confusion matrices potentially available in the attached python notebook. For clarity and conciseness, I'll provide the accuracies of the seven models (note: only the accuracy of the ensemble is mentioned, not the individual models):

- Simple features: 0.654
- Basic features: 0.673

- Basic with academic features: 0.67
- Basic with financial features: 0.76
- Basic with full academic: 0.913
- Basic with full academic and financial: 0.924
- Full data model: 0.937

These accuracy scores highlight the varying performance levels of each model configuration. The incorporation of different feature sets and data completeness plays a significant role in improving the accuracy of the predictions. The highest accuracy is achieved with the full data model, showcasing the effectiveness of a comprehensive dataset in enhancing the overall predictive capabilities of our IDSS.

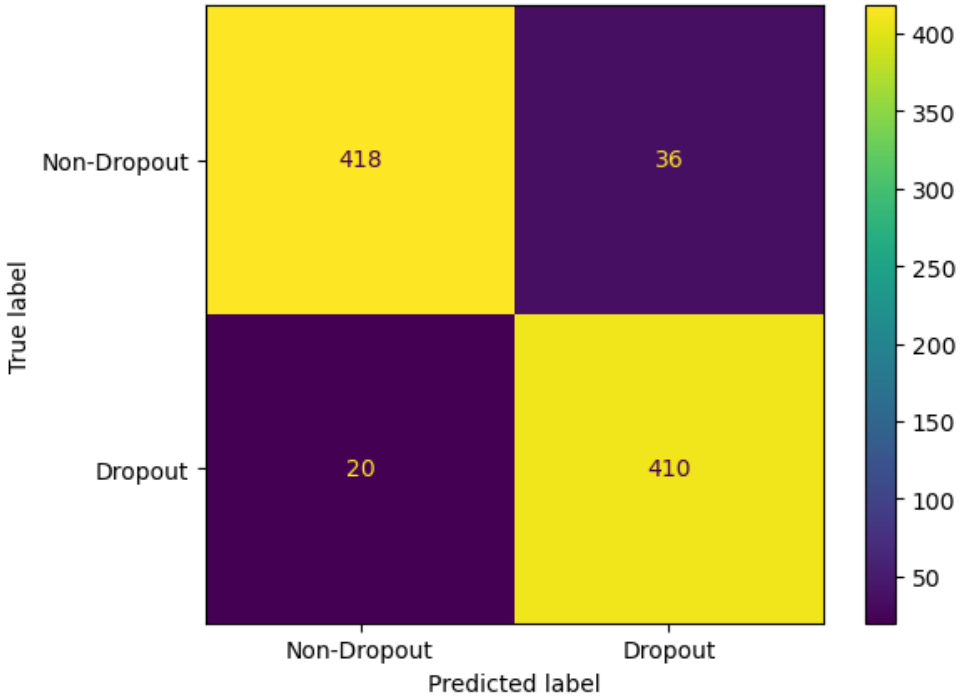

Figure 5: Confusion Matrix for the Full Data Model

As we can observer in the Fig. 5, the two data classes are equally accurate in being extracted; indeed, the false positives and false negatives are well balanced. This information is crucial because, in our case, we aim for both classes (dropout and graduate) to be as balanced as possible in the predictions. Having a balanced prediction is more important than having one of the classes at zero, and this is reflected in the balanced distribution of false positives and false negatives.

8 Model-driven IDSS techniques used

Our Intelligent Decision Support System relies on a model-driven approach to enhance its predictive capabilities. In it, we have incorporated various techniques to create a versatile support tool.

- **expert opinions** A survey, designed to simulate a sort of college interview, collects some qualitative insights from the students, which are then evaluated and scored, from 1 to 10, by an expert (probably a professor or a credible academic figure)
- **Data Mining Models:**
  - Utilizes supervised learning techniques to train models on labeled data for accurate predictions.
  - Incorporates discriminant models to distinguish between different classes, contributing to a nuanced understanding of potential outcomes.
- **Statistical Models / Rule-Based Models:**
  - Explores statistical models and rule-based models for transparency and interpretability.
  - Reveals relationships between different features, enhancing the system's understanding of student outcomes.
  - **Logistic Regression:**
    - Selected as a key model for binary classification problems, capturing the probability of events such as student dropout or graduation.
    - Valued for its simplicity, interpretability, and effectiveness in predicting outcomes based on input features.
  - **Random Forest and Support Vector Machines (SVM):**
    - Selected to expand the scope of predictions and enhance the robustness of forecasting

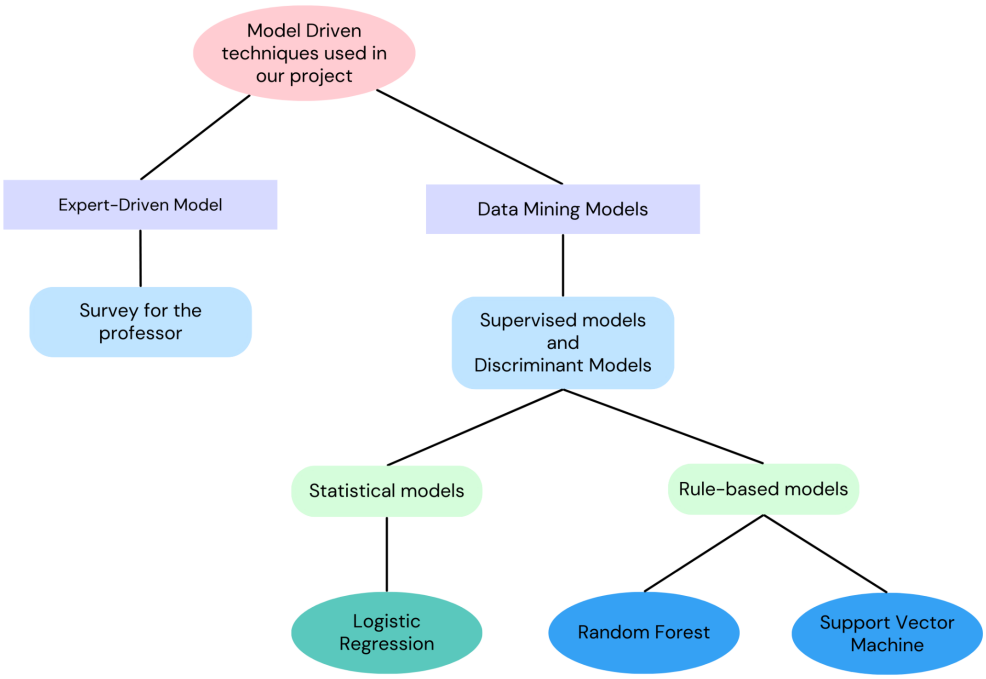

Figure 6: Model-based scheme of our IDSS

9 Evaluation of results and conclusions

In terms of evaluation, we lack feedback on the expert-based component as we simulated its creation, assuming collaboration with an expert. However, we can compare our data-driven part with similar projects on Kaggle (`https://www.kaggle.com/datasets/thedevastator/higher-education-predictors-of-student-retention`). Many contributors on Kaggle, focusing primarily on logistic regression, achieved a maximum result of around 92%. In contrast, by incorporating various models, we managed to slightly outperform these results. Our accuracy in the full model is just below 94%, indicating an improvement of almost 2%.

Our project goes beyond a typical machine learning and data analysis effort. We developed more than 21 models (considering three models for each category) and, crucially, integrated the data-driven part with an expert-based approach. This integration allows us to be satisfied with the work done, demonstrating how decision support systems can benefit from the coupling of machine learning models and expert-based insights.

While we acknowledge the success, it's important to note that the accuracy of this system needs confirmation, potentially by fine-tuning the proportion of predictions from the data-driven models and the expert-based model in the future.

10 Future work and improvements

Currently, the evaluation is done by a single professor. To enhance the robustness of the opinions, future work could support the evaluation of multiple professors for the same student.

The current implementation of the model provides predictions without describing the reason for the result. An improvement could involve incorporating tools for explainability. The inclusion of this feature would give trust by providing information for a particular decision. This could be implemented using techniques such as LIME or SHAP.

Another improvement could be the incorporation of a feedback method from the users. This would provide a method to identify issues and areas of improvement.

11 Gantt diagram and tasks planning

We divided our work in two fundamental phases:

- **Phase 1:** A design phase, with Dataset selection and analysis, definition of general Design for our IDSS
- **Phase 2:** The implementation phase, with data preparation and model development, Application development, integration of expert opinion and different feature models selection, and report documentation

the first phase took the first 20 days, while the second took the last 16, as you can see in the following image representing the Gantt Diagram:

Educational Success Predictor

Group 2
Gustavo Rayo, Umberto Salviati, Giacomo Sanguin

Project Start: 1/12/2023
Display Week: 1

TASK	ASSIGNED	PROGRESS	START	DAYS	END
Phase 1		100%	**1/12/2023**		**21/12/2023**
Dataset Selection		100%	1/12/2023	7	7/12/2023
Defining Scope and Objectives		100%	8/12/2023	6	13/12/2023
Dataset Analysis		100%	8/12/2023	6	13/12/2023
Web App Framework Selection		100%	14/12/2023	8	21/12/2023
Phase 2		100%	**22/12/2023**		**7/1/2024**
Data Preparation		100%	22/12/2023	3	24/12/2023
Model Development		100%	26/12/2023	6	31/12/2023
Application development		100%	28/12/2023	6	2/1/2024
Expert opinion Integration		100%	3/1/2024	1	3/1/2024
Customization Options		100%	4/1/2024	1	4/1/2024
Testing and Deployment		100%	5/1/2024	2	6/1/2024
Report Documentation		100%	26/12/2023	13	7/1/2024

Figure 7: Gantt diagram and task planning

12 Tasks assignment and responsibilities among teamwork members

In order to optimize the our work, we divided all the different tasks between team members. However, it's important to say that this project was a collective effort, in which all the members helped each other out during the implementation phase. In particular, during the first phase, in which we decided the objectives of our project and selected the dataset, the work was done collectively during in-presence or video session.

In general, this was the main task subdivision:

- **Gustavo Rayo:** Web App Framework Selection, Application Development, Customization Option
- **Umberto Salviati:** Data preparation, Model Development, Testing and deployment
- **Giacomo Sanguin:** Dataset Analysis, Expert Opinion Integration, Report Documentation

13 Time sheet

In general, the Data selection and the Definition of the scope and objectives, was done toghether for a total of 12 hours. This amount will be added to each member in the total work load

- **Gustavo Rayo:** Total of 57 hours
  - Web App Framework Selection: 4 hours
  - Application Development: 40 hours
  - Customization Option: 5 hours
- **Umberto Salviati:** Total of 60 hours
  - Data Preparation: 8 hours
  - Model development: 25 hours
  - Testing and Deployment: 15 hours
- **Giacomo Sanguin:** Total of 52 hours
  - Dataset Analysis: 15 hours
  - Expert Opinion Integration: 10 hours
  - Report Documentation: 15 hours

It's important to add that, in any case, the hours amount is a generic evaluation of the workload done, since every team member in the end, contributed to almost all of the tasks.

14 Conclusion

In conclusion, our Intelligent Decision Support System, stands as a robust solution, leveraging predictive modeling and expert insights to improvee our understanding of students' academic journeys. Its approach helps educators and administrators in identifying at-risk students early, offering personalized assistance and optimizing resource allocation. Our "Educational Success Predictor" emerges then as a valuable tool, contributing to informed decision-making and encouraging a positive learning environment within educational institutions.

15 Annexes

Link of the IDSS: http://idss.eastus.cloudapp.azure.com/