UNDER GRADUATE STUDENT DROPOUT PREDICTION USING MACHINE LEARNING

[1] D Mahitha, [2]A. Ramya Sri, [3]A. Rishitha, [4]A. Jeevani, [5]B. Sanya
[1]Assistant Professor, [2,3,4,5]UG Students, Dept. of CSE (AI & ML), Malla Reddy Engineering College for Women (Autonomous), Hyderabad, India. E-Mail: mahithadilli@gmail.com

ABSTRACT
Higher education plays a vital role in national development, but promoting educational quality and addressing the issue of university dropouts pose significant challenges for governments. University dropouts not only have negative economic consequences but also impact students personally. This project focuses on finding the factors that are affecting the student dropouts from semester to semester and different classification models can be built to predict whether a student is on the verge on dropping out of college and helps in giving early warnings to the university management to help such students. By leveraging principles of data mining and machine learning, we collected student data from the admin internal sources by making sure the data has no missing or null values. The dataset comprises information from 4424 undergraduate students. Utilizing Support Vector Classifier (SVC) and Logistic Regression, we built a basic model that is capable of predicting the on the verge drop out students and the model also is capable of understanding the most affecting features to the dropout rate. Our findings highlight the top five significant features for predicting educational status: Curricular units 2nd semester (approved), Curricular units 1st semester (approved), Curricular units 2nd semester (grade), Curricular units 1st semester (grade), and Tuition fee up to date. These top five features showed us that the most affecting features are the grades in the previous semesters. In order to increase the accuracy of the predictions, we took two datasets- DBS.csv and DBS_2000.csv. Through careful analysis, we identified key features that contribute to improving the accuracy and applicability of our predictive models.

INTRODUCTION
The aim of this project is to address the challenges of promoting educational quality and reducing university dropouts in higher education, which are crucial for national development. The negative economic consequences and personal impact on students resulting from university dropouts necessitate a comprehensive understanding of the factors influencing student's educational status. To achieve this, the researchers utilized principles of data mining and machine learning to collect and analyse data from internal sources.
The dataset used in this project comprises information from 4424 undergraduate students, providing a robust foundation for analysis and prediction. By leveraging Support Vector Classifiers (SVC), we developed classification models to predict whether a student will be a dropout or fall into non-dropout categories.
To ensure the accuracy and applicability of the predictive models, a meticulous analysis of the dataset was conducted. We identified key features that significantly contribute to predicting educational status. These features include Curricular units 2nd sem (approved) , Curricular units 1st sem (approved), Curricular units 2nd sem (grade), Curricular units 1st sem (grade), Tuition fees up to date. By incorporating these features, the classifiers were able to make reliable predictions regarding student's educational outcomes.
The findings of this project shed light on the complex factors that influence university dropouts and provide valuable insights for implementing targeted interventions. With a deeper understanding of these factors, educational institutions and policymakers can develop strategies to enhance educational quality and support student success. By leveraging the power of data mining and machine learning, this research contributes to the

ongoing efforts to improve higher education systems, foster national development, and empower students to achieve their educational goals.

LITERATURE SURVEY

An "ever-enrolled person" who does not finish the last level of education for which he or she enrolled and is not currently enrolled in any educational institution is considered a dropout. The implementation of a socially equal environment presents a significant problem for the Indian community in terms of student dropout. According to a recent poll by the National Statistical Office (NSO), 12.6% of Indian pupils drop out of school, 19.8% of them do so in secondary school, and 17.5% do so in upper primary. In addition to the student's bad self-perception brought on by the sense of failure and despair, university dropouts have an economic and societal consequence.

Even while the government's Right to Education Act and National Policy on Education may have encouraged everyone to receive an education, it is equally necessary to assess the system's viability and effectiveness. Dropout rates are viewed as a huge waste of the educational system's limited resources since many students quit school before gaining even the most basic abilities and because their early departure implies a significant loss of those resources.

Schools can identify students who are at risk of dropping out and concentrate on those who have difficulty performing well thanks to dropout early warning systems. Children and adolescents who drop out of college typically do so for a variety of reasons. Instead, this is a process that is shaped by numerous variables that interact in complicated, dynamic ways. These variables may relate not just to traits or situations specific to a person or family, but also to variables at the college, local, and international levels.

These can include shortcomings in educational environments and procedures, in social welfare and education systems, in general social policies for young people and employment, and in societal norms, especially gender standards that may be detrimental to education. The reason why kids and teenagers leave school is frequently due to "individual and family circumstances that structures and systems are unable to respond to or address appropriately."

EXISTING SYSTEM

Early Warning Systems (EWS) for school children who are at risk of quitting. UNICEF supports nations and education experts in enhancing their approaches to stop school dropout. It promotes EWS as a means of fostering fair access to high-quality education and lowering dropout rates. The main objective is to spot pupils who are at risk of leaving school before they stop attending altogether.

Choosing indicators to pinpoint students at risk of dropping out is the first step in the current system for forecasting student dropouts, which normally entails a step-by-step procedure. The current system handles this step as follows:

Formation of a school staff group: The existing system establishes a group consisting of school management, teachers, and support staff. This group collaboratively works to establish indicators for the Early Warning System (EWS).

Identification and prioritization of risk factors: Based on contextual evidence, the school staff group determines and ranks the key risk factors and predictors of dropout. To comprehend dropout trends and causes, they study data on dropout from prior years that is currently available.

Formulation of indicators: Indicators are formulated based on the identified predictors of dropout. Each predictor is translated into an indicator that can be measured or observed. For example, indicators may include absenteeism, academic performance, behaviour issues, socio-economic status, special alerts for specific circumstances, and a sense of belonging to the school.

Data collection methodology: The existing system determines how data will be collected for each indicator. It

considers existing data collection systems, such as electronic or paper-based systems, to ensure the accessibility and reliability of the necessary data. Special attention is given to reliable absenteeism data, including justified and non-justified absences, tardiness, and reasons for these.

Weighting and thresholds: The school staff group may give various indications varying weights in order to increase the EWS's sensitivity to dropout concerns. They concur on cutoff points that specify when pupils will be classified as "at risk" or "at high risk" of failing to graduate. This aids in classifying students according to their degree of danger.

Based on data obtained on early leaver profiles and the most significant risk factors, the current system continuously adjusts the indicator sets, weighting, and scoring systems. Through this iterative approach, the system's capacity to forecast student dropouts can be improved continuously.

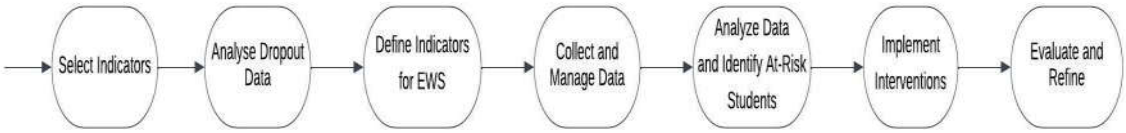


Fig.1. Existing System flowchart

PROPOSED SYSTEM

There is a need to identify multiple dimensions that influence dropout, including individual, family, community, and school factors. By considering a wide range of indicators, EWS can effectively capture the complexity of dropout risks. The existing system being a manual procedure takes a lot of time and effort into finding out the drop outs leaving none for the rectification of problems causing the dropouts.

Machine learning (ML) approaches can significantly aid Early Warning Systems (EWS) by automating and enhancing the process of predicting student dropouts. This way, the efforts of the school/ university committees can be used where it is actually needed. Here's how ML can assist EWS and reduce the need for manual work:

1. **Feature selection:** ML algorithms can automatically identify the most relevant features or indicators for predicting student dropout. Instead of relying on manual selection and prioritization of indicators, ML algorithms can analyse large amounts of data to identify patterns and relationships that are predictive of dropout.
2. **Data preprocessing:** ML techniques can handle data preprocessing tasks such as handling missing values, normalizing data, and encoding categorical variables. This automation streamlines the data preparation phase, saving time and effort compared to manual data cleaning and transformation.
3. **Predictive modelling:** ML algorithms, such as logistic regression, decision trees, random forests, or neural networks, can be trained on historical data to create predictive models. These models can learn complex relationships between predictors and dropout outcomes, providing accurate predictions for individual students.
4. **Continuous learning:** ML models can continuously learn and adapt based on new data. As more data becomes available, the models can be updated to incorporate the latest information, ensuring that the predictions remain relevant and accurate over time. This dynamic learning capability reduces the need for manual model adjustments.
5. **Identification of complex patterns:** ML algorithms can uncover hidden patterns and interactions among predictors that may not be evident through manual analysis. This ability to detect complex

relationships can lead to more accurate predictions and identification of at-risk students who may not fit conventional risk profiles.

6. **Scalability:** ML approaches allow for the efficient analysis of large-scale datasets. They can process vast amounts of student data quickly, enabling EWS to handle data from multiple schools or districts. This scalability is especially beneficial when dealing with extensive student populations.
7. **Performance evaluation:** ML techniques provide automated evaluation metrics to assess the performance of the predictive models. These metrics, such as accuracy, precision, recall, and F1 score, help quantify the effectiveness of the ML-based EWS and guide further improvements.

It's important to note that while ML approaches offer significant advantages, they still require careful consideration and domain expertise. Interpretability of ML models and ethical considerations should be taken into account to ensure transparency, fairness, and accountability in the decision-making process of EWS.

By leveraging ML techniques, EWS can streamline the process of predicting student dropouts, reduce manual efforts in feature selection and indicator formulation, and improve the accuracy and efficiency of dropout risk identification. Here are the steps:

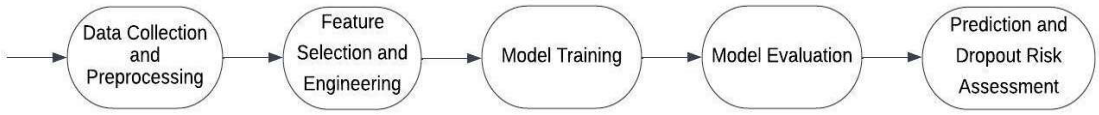


Fig.2. Proposed System Flowchart

Step 1: Data Collection and Preprocessing

- Gather student data (demographics, academic records, attendance, behavior, etc.)
- Clean and preprocess the data (handle missing values, normalize features, etc.)
- Split the data into training and testing sets
- Label encoding and ordinal encoding of all the values
- Data scaling into the interval -1

Step 2: Feature Selection and Engineering

- Identify relevant features for dropout prediction
- Perform feature selection techniques (e.g., correlation analysis, feature importance)

Step 3: Model Training

- Apply SVM algorithm for training the model
- Choose appropriate SVM kernel (linear, polynomial, radial basis function, etc.)
- Optimize hyper parameters through techniques like cross-validation

Step 4: Model Evaluation

- Evaluate the trained model using the testing dataset
- Evaluating the model using a confusion matrix
- Calculate performance metrics (accuracy, precision, recall, F1-score, etc.)
- Analyse the model's performance and identify any shortcomings

Step 5: Prediction and Dropout Risk Assessment

- Use the trained SVM model to predict dropout probabilities for new student data
- Classify students into risk levels (low, medium, high) based on predicted probabilities
- Identify students at high risk of dropout for intervention

Next steps that can be followed

Step 1: Intervention and Support

- Implement targeted interventions and support strategies for at-risk students

- Provide personalized counselling, academic assistance, or mentoring
- Collaborate with relevant stakeholders to address identified risk factors

Step 2: Monitoring and Refinement

- Monitor the effectiveness of interventions in reducing dropout rates
- Collect feedback from students, teachers, and other stakeholders
- Continuously refine the ML model and update the training data

SYSTEM ARCHITECTURE

The process begins at the time of admissions. When a student joins into the student, he/she submits details like the salary of parents, education history, etc. For this model, along with this information, it needs more info like the marital status of the student, the relationship status of the parents, etc. The admin office can give a form to be filled to every student explaining the need.

Thus, all the details of the student are right in the hands of the student which are needed to execute this model. The university can now appoint an EWS analyzer who needs to make a separate dataset with all the student details required for the model. The next task for the analyzer is to keep updating the dataset with course grade with every semester.

Now, the analyzer has to keep training and executing the model every semester in order to predict the students with a high chance of dropping out in the next semester. the analyzer and the college management can work out the most effective features which are determining the dropout rate with statistical analysis techniques like correlation matrix, hypothesis testing to reduce the rate of drop outs.

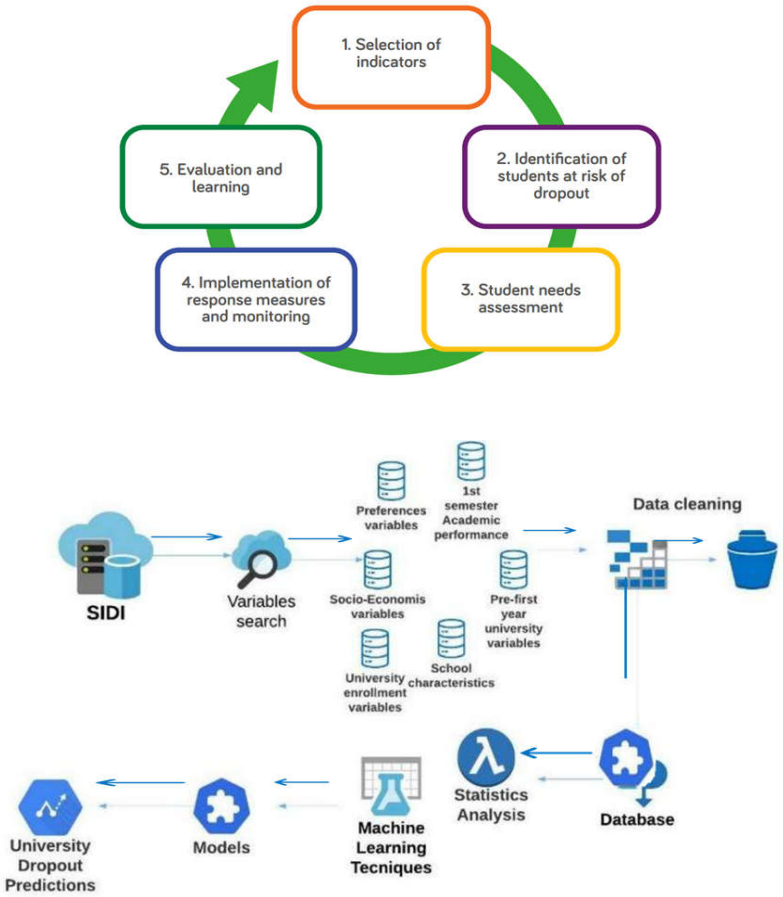


Fig.3. System Architecture

DEFINE MODULES & FUNCTIONALITIES:

From the library Scikit-learn (sklearn):

a. **MinMaxScaler:** A module for data normalization or standardization.
b. **train_test_split:** A function for splitting the data into training and testing sets.
c. **LabelEncoder:** A class for encoding categorical variables as integers.
d. **OrdinalEncoder:** A class for encoding categorical variables as integers with an ordinal relationship.
e. **svm:** A module for Support Vector Machine algorithms.
f. **datasets:** A module providing various datasets for testing and practicing machine learning algorithms.
g. **GridSearchCV:** A class for performing hyperparameter tuning using grid search and cross-validation.
h. **csv:** The csv module provides functionality for reading and writing CSV (Comma-Separated Values) files. It is a built-in module in Python.
i. **Axes3D:** Axes3D is a class in the mpl_toolkits.mplot3d module of Matplotlib that allows plotting in 3D.
j. **Pyplot:** Pyplot is a module within the Matplotlib library. Matplotlib is a plotting library in Python that provides a wide range of functions for creating static, animated, and interactive visualizations.

IMPLEMENTATION

LIBRARIES:

1. **SEABORN**:
   Seaborn is a data visualization library built on top of Matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics.We used this library to construct a heatmap

2. **PANDAS:**
   Pandas is a powerful library for data manipulation and analysis. It provides data structures like DataFrames that allows us to efficiently work with structured data.
3. **MATPLOTLIB:**
   Matplotlib is a plotting library that provides a flexible and comprehensive set of plotting functions. It is widely used for creating static, animated, and interactive visualizations in Python.
4. **SCIKIT-LEARN (SKLEARN):**
   Scikit-learn is a popular machine learning library in Python. It provides a wide range of algorithms for classification, regression, clustering, dimensionality reduction, and more. Our project implementation is mostly dependent on the modules of this library.

5. **NUMPY:**
   NumPy is a fundamental library for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions.

ALGORITHMS

Multi-dimensional multi-class Support Vector Classifier:

The Multi-dimensional multi-class Support Vector Classifier is a supervised machine learning algorithm used for classification problems. It is an extension of the Support Vector Machine (SVM) algorithm. SVM aims to find an optimal boundary between different output classes by performing complex data transformations using kernel functions. These transformations maximize the separation between data points based on the defined classes or labels.

In its basic form, SVM seeks to find a line that maximizes the separation between two classes in a two-dimensional space. However, in the case of multi-dimensional data, the objective is to find a hyperplane that maximizes the separation of data points into their respective classes in an n-dimensional space. The data points that are closest to the hyperplane are known as Support Vectors.In the case of multi-class classification with

three target classes (dropout, graduate, and enrolled), binary classification cannot be directly applied. However, the Support Vector Classifier (SVC) algorithm provides two approaches for multi-class classification:

One-to-One approach

This approach breaks down the multi-class problem into multiple binary classification problems. For each pair of classes, a binary classifier is trained to separate the instances of one class from the instances of the other class. In this approach, multiple hyperplanes are created to separate each pair of classes, ignoring the points belonging to the third class.

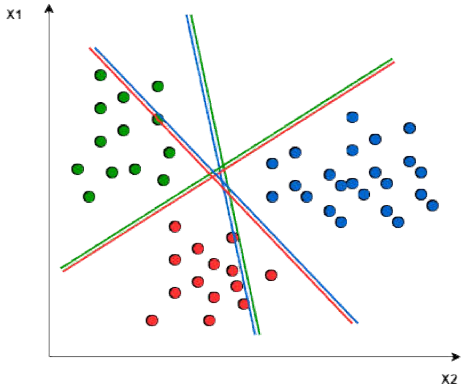


Fig.4. One-to-One approach:

One-to-Rest approach

In this approach, a hyperplane is created to separate one class from the rest of the classes all at once. The separation takes into account all the points and divides them into two groups: one group for the points of the class under consideration and another group for all the remaining points representing the other classes.

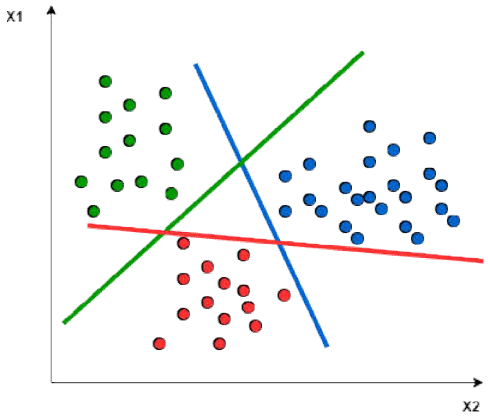


Fig.5. One-to-Rest approach:

To implement these approaches, you can use the SVC function provided by the sklearn library. This function allows you to specify the desired kernel functions for the SVM algorithm, such as linear, polynomial, or radial basis function (RBF), among others.

Logistic Regression

Based on a given dataset of independent variables, logistic regression calculates the likelihood that an event will occur, such as voting or not voting. A supervised ML model is what it is. Given that the result is a probability, the dependent variable's range is 0 to 1. In logistic regression, the odds—that is, the probability of success divided by the probability of failure—are transformed using the logit formula.

In contrast to linear regression, logistic regression does not simply fit a straight line to the data. Instead, we used the Sigmoid curve to fit an S-shaped curve to our observations. The following formulas are used to represent this logistic function, which is also referred to as the log odds or the natural logarithm of odds:

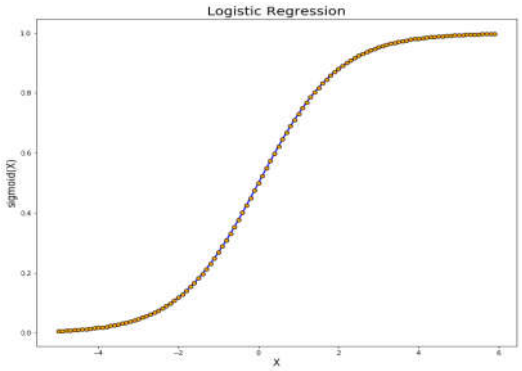


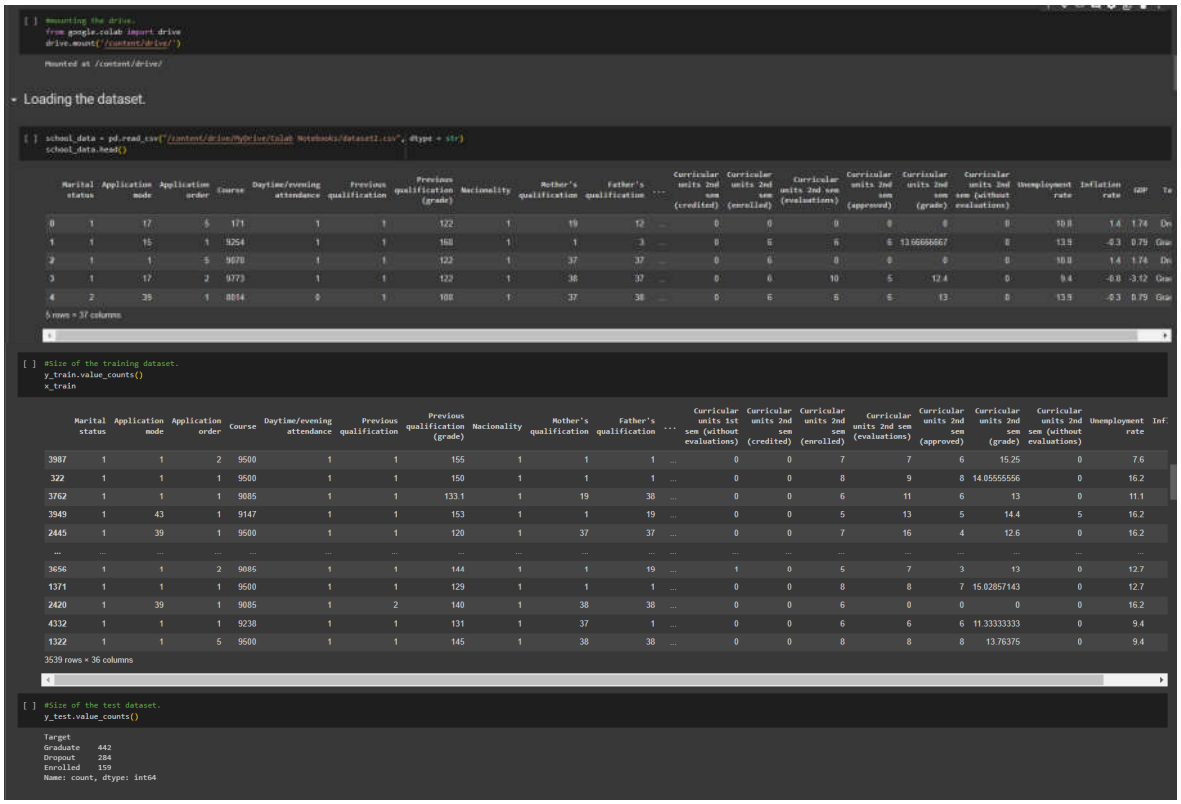
Fig.6. Logistic regression

$$\ln(pi/(1-pi)) = Beta\_0 + Beta\_1*X\_1 + \ldots + B\_k*K\_k$$

Logit(pi) is the dependent or response variable in this logistic regression equation while x is the independent variable. Most frequently, maximum likelihood estimation (MLE) is used to estimate the beta parameter, or coefficient, in this model. In order to find the best fit for the log odds, this approach iteratively evaluates various beta values. The log likelihood function is created after each of these iterations, and logistic regression aims to maximise this function to get the most accurate parameter estimate.

The conditional probabilities for each observation can be calculated, logged, and added together to produce a forecast probability once the best coefficient (or coefficients, if there are multiple independent variables) has been identified. If the categorization is binary, a probability of less than.5 predicts 0 and a probability of more than 0 predicts 1. It is recommended to assess the model's goodness of fit, or how well it predicts the dependent variable, once the model has been computed.

RESULT



Visualization

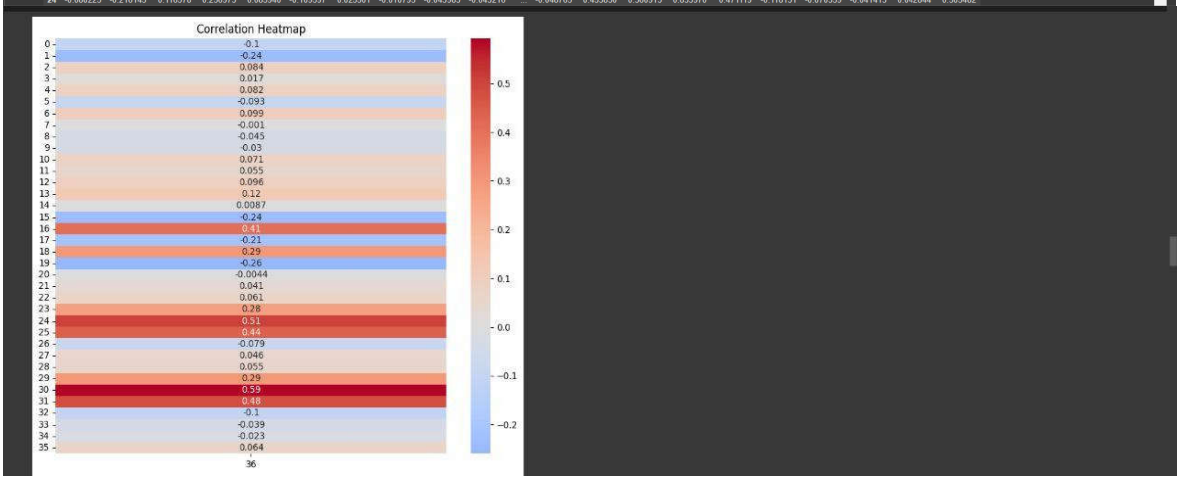
```
[ ] temp = [[i] for i in y_train_transf]
    arr = np.append(x_train_rescaled, temp, axis=1)
    df = pd.DataFrame(arr)
    df
```

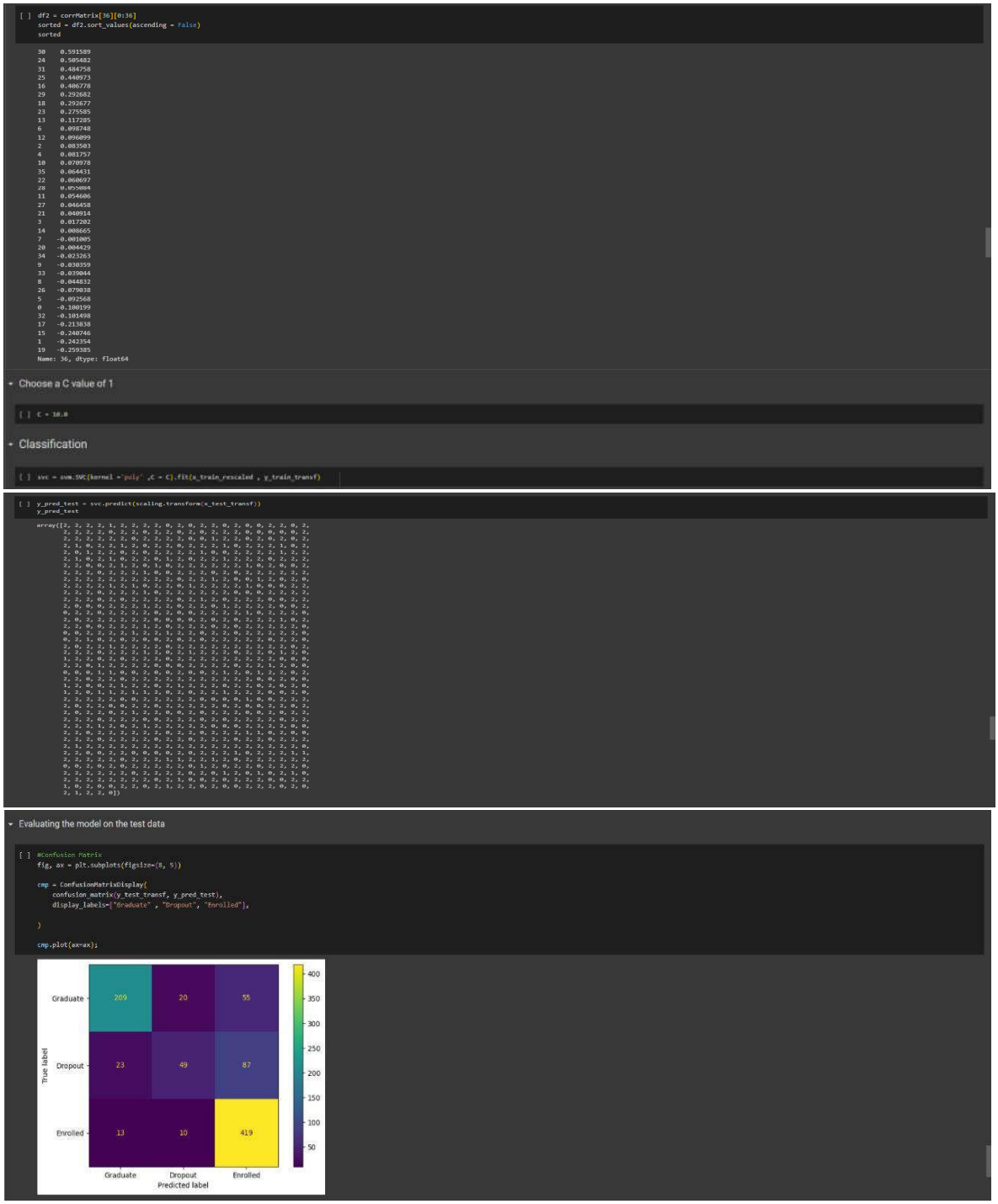
	0	1	2	3	4	5	6	7	8	9	...	27	28	29	30	31	32	33	34	35	36
0	-1.0	-1.000000	-0.428571	0.375	1.0	-1.00	0.376344	-1.0	-1.000000	-1.000000	...	-1.0	0.8	0.862069	0.684211	0.805389	-1.000000	0.555556	0.50	-0.111111	2.0
1	-1.0	-1.000000	-0.714286	0.375	1.0	-1.00	0.247312	-1.0	-1.000000	-1.000000	...	-1.0	0.9	1.000000	0.894737	0.356287	-1.000000	0.333333	-0.50	-1.000000	2.0
2	-1.0	-1.000000	-0.714286	-0.375	1.0	-1.00	-0.204301	-1.0	-0.615385	0.419355	...	-1.0	0.7	-0.793103	0.684211	-0.221557	-1.000000	-0.777778	0.00	0.777778	2.0
3	-1.0	0.294118	-0.714286	0.000	1.0	-1.00	0.311828	-1.0	-1.000000	-0.548387	...	-1.0	0.6	-0.655172	0.578947	0.517964	0.333333	0.333333	-0.50	-1.000000	2.0
4	-1.0	0.058824	-0.714286	0.375	1.0	-1.00	-0.548387	-1.0	0.153846	0.354839	...	-1.0	0.8	-0.448276	0.473684	-0.437126	-1.000000	0.333333	-0.50	-1.000000	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3534	-1.0	-1.000000	-0.428571	-0.375	1.0	-1.00	0.075269	-1.0	-1.000000	-0.548387	...	-1.0	0.6	0.862069	0.368421	-0.221557	-1.000000	-0.333333	1.00	-0.777778	2.0
3535	-1.0	-1.000000	-0.714286	0.375	1.0	-1.00	-0.311828	-1.0	-1.000000	-1.000000	...	-1.0	0.9	0.931034	0.789474	0.751497	-1.000000	-0.333333	1.00	-0.777778	2.0
3536	-1.0	0.058824	-0.714286	-0.375	1.0	-0.25	-0.010753	-1.0	0.230769	0.419355	...	-1.0	0.7	-1.000000	-1.000000	-1.000000	-1.000000	0.333333	-0.50	-1.000000	0.0
3537	-1.0	-1.000000	-0.714286	0.125	1.0	-1.00	-0.268817	-1.0	0.153846	-1.000000	...	-1.0	0.7	0.793103	0.684211	-0.880240	-1.000000	1.000000	-0.75	-0.555556	2.0
3538	-1.0	-1.000000	0.428571	0.375	1.0	-1.00	0.096774	-1.0	0.230769	0.419355	...	-1.0	0.9	0.931034	0.894737	0.212575	-1.000000	1.000000	-0.75	-0.555556	2.0

3539 rows × 37 columns

```
[ ] corrMatrix = df.corr()
    corrMatrix
```

	0	1	2	3	4	5	6	7	8	9	...	27	28	29	30	31	32	33	34	35	36
0	1.000000	0.251782	-0.127549	0.014457	-0.255111	0.098102	-0.017977	-0.030878	0.144253	0.113437	...	0.059768	0.001562	-0.033290	-0.080217	-0.089182	0.012637	0.020622	0.012333	-0.054526	-0.100199
1	0.251782	1.000000	-0.297002	-0.092397	-0.287456	0.582386	-0.021217	-0.029327	0.096010	0.121512	...	0.346486	-0.080290	-0.129970	-0.200694	-0.128466	0.055226	0.020332	-0.033064	-0.034459	-0.242354
2	-0.127549	-0.297002	1.000000	0.118429	0.160305	-0.197197	-0.054212	-0.035742	-0.057642	-0.088823	...	-0.146907	0.115436	0.082477	0.110096	0.069265	-0.008020	0.012496	-0.009585	0.037019	0.083503
3	0.014457	-0.092397	0.118429	1.000000	-0.081978	-0.135164	-0.077563	-0.022756	0.014257	0.019362	...	-0.124840	0.400384	0.212827	0.234207	0.118456	-0.017105	-0.013489	0.026182	-0.036229	0.017202
4	-0.255111	-0.287456	0.160305	-0.081978	1.000000	-0.098348	0.048618	0.017460	-0.164603	-0.142609	...	-0.160067	0.059196	-0.009608	0.077421	0.102276	0.006035	-0.042988	-0.013520	0.058385	0.081757
5	0.098102	0.582386	-0.197197	-0.135164	-0.098348	1.000000	0.075661	-0.038770	0.029745	0.032647	...	0.301499	-0.035857	-0.084513	-0.080759	0.001178	0.011778	0.021148	-0.055645	0.028088	-0.092568
6	-0.017977	-0.021217	-0.054212	-0.077563	0.048618	0.075661	1.000000	0.028430	-0.012519	0.000070	...	-0.014764	-0.025660	0.023161	0.027004	0.184918	-0.017801	0.029611	-0.000948	-0.104630	0.098748
7	-0.030878	-0.029327	-0.035742	-0.022756	0.017460	-0.038770	0.028430	1.000000	-0.013246	-0.045574	...	0.018658	-0.057036	-0.017311	-0.030123	-0.011133	-0.008109	-0.003041	0.011635	0.069202	-0.001005
8	0.144253	0.096010	-0.057642	0.014257	-0.164603	0.029745	-0.012519	-0.013246	1.000000	0.380193	...	0.011785	-0.016666	-0.025830	-0.037755	-0.056269	0.026610	0.057539	0.028998	-0.094227	-0.044832
9	0.113437	0.121512	-0.088823	0.019362	-0.142609	0.032647	0.000070	-0.045574	0.380193	1.000000	...	0.026619	-0.025414	-0.020052	-0.034005	-0.062311	-0.003208	0.053149	0.037831	-0.083594	-0.030359
10	0.033516	-0.021295	0.000617	0.041208	0.011110	-0.028930	-0.007277	0.002194	0.031584	0.060516	...	-0.022460	0.027813	0.064923	0.077542	0.037313	-0.017035	-0.118235	-0.005788	0.015186	0.070978
11	0.014148	-0.063021	0.024337	0.033824	0.015603	-0.050587	0.006431	0.020180	0.024177	0.097883	...	-0.039975	0.021769	0.034639	0.058653	0.021811	-0.062220	-0.083797	0.044794	0.053883	0.054606
12	0.000844	0.040852	-0.101359	-0.095826	-0.000973	0.174871	0.553855	-0.004404	0.014942	-0.008490	...	0.021542	-0.077955	0.021071	0.002000	0.181779	-0.030019	-0.003382	-0.002111	-0.045136	0.096099
13	-0.229618	-0.310220	0.332750	0.005407	0.252577	-0.152766	-0.005578	-0.010925	-0.085665	-0.081415	...	-0.120013	-0.001768	0.025470	0.096359	0.071675	-0.029317	0.010353	-0.003332	0.078683	0.117285
14	-0.026743	-0.039660	0.019243	-0.002630	0.028384	-0.022260	-0.007992	0.002227	-0.013978	-0.000098	...	-0.021570	-0.014447	-0.032944	0.009688	-0.010709	-0.012668	-0.024947	0.009448	-0.002250	0.008665
15	0.037284	0.116291	-0.084128	-0.055033	0.003301	0.091767	-0.026160	0.076624	0.011371	0.013750	...	0.032399	-0.028441	-0.036696	-0.151565	-0.133152	0.039709	0.019060	-0.021780	0.072797	-0.240746
16	-0.095277	-0.153926	0.059847	0.032092	0.047741	-0.097809	0.042962	-0.047681	-0.005750	-0.031758	...	0.020069	0.012476	0.068895	0.305626	0.212487	-0.074371	-0.024988	-0.015546	-0.017238	0.406778
17	-0.013744	0.121418	-0.087152	-0.110206	-0.013754	0.055358	-0.042474	-0.026519	-0.034640	-0.040258	...	0.014475	-0.126518	-0.072707	-0.227700	-0.148915	0.068229	0.041116	0.020354	-0.022701	-0.213838
18	-0.053570	-0.195553	0.069995	0.044517	0.093164	-0.106976	0.040703	-0.024464	-0.034682	0.048388	...	-0.065377	0.077847	0.150186	0.224127	0.157828	-0.038279	-0.060584	-0.015704	0.069812	0.292677
19	0.512825	0.535173	-0.277939	-0.040690	-0.447784	0.235438	-0.098414	-0.005660	0.248790	0.205277	...	0.196896	-0.077729	-0.083678	-0.245708	-0.152713	0.048142	0.041738	0.018820	-0.096007	-0.259385
20	-0.024448	-0.033448	-0.034526	-0.008347	0.018519	-0.046011	0.041694	0.940735	-0.012300	-0.048677	...	0.009251	-0.043855	-0.008768	-0.025942	-0.009332	-0.010955	-0.001965	0.013295	0.061728	-0.004429
21	0.061747	0.371380	-0.150108	-0.141853	-0.160609	0.309141	0.007062	0.038741	0.015511	0.032112	...	0.849790	-0.291163	-0.151095	0.018289	0.043606	0.009659	0.063259	-0.011010	0.055734	0.040914
22	-0.010264	-0.124758	0.121839	0.396156	0.075014	-0.082074	-0.042477	-0.028784	-0.002787	-0.030340	...	-0.434615	0.864278	0.374984	0.362991	0.183174	0.007834	-0.085763	-0.008664	-0.010793	0.060697
23	-0.060561	-0.157721	0.100584	0.220410	0.039213	-0.081800	0.027234	-0.033679	-0.030960	-0.047680	...	-0.182425	0.362321	0.518835	0.364818	0.331684	-0.006103	-0.060818	0.018873	0.078685	0.275585
24	-0.080225	-0.216145	0.116578	0.256973	0.085940	-0.109597	0.025501	-0.010795	-0.043983	-0.043216	...	-0.048765	0.453830	0.360913	0.833970	0.471119	-0.118131	-0.070359	-0.041413	0.042844	0.505482





Using Logistic Regression:

```
[3] from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive
```

```
[5] # Importing and cleaning the data
    train_data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/DBS.csv', sep=';')
    test_data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/DBS_2020.csv', sep=';')
    train_data.head()
```

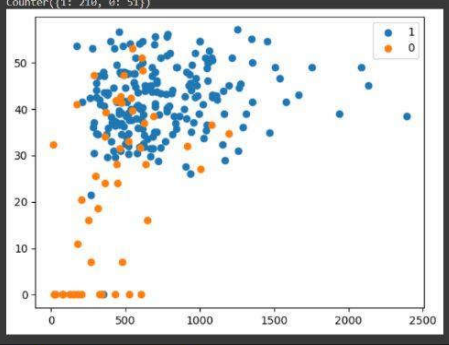
	access	tests	tests_grade	exam	project	project_grade	assignments	result_points	result_grade	graduate	year	acad_year
0	1256	57.00	A	19	91.54	A	40.0	189.92	A	1	2019	2019/2020
1	985	42.87	B	19	75.96	A	13.7	189.43	A	1	2017	2017/2018
2	1455	54.50	A	16	96.79	A	40.0	188.91	A	1	2019	2019/2020
3	998	54.50	A	16	93.36	A	40.0	186.85	A	1	2019	2019/2020
4	1347	55.00	A	16	92.86	A	39.0	186.38	A	1	2019	2019/2020

```
# Data normalization with sklearn
from sklearn import preprocessing
standardized_X = preprocessing.scale(X_test)
df = pd.DataFrame(standardized_X)
df.describe()
```

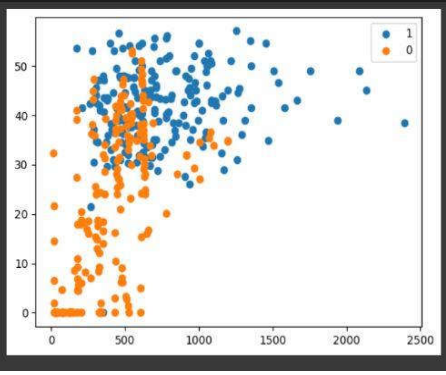
	0	1	2
count	6.000000e+01	6.000000e+01	6.000000e+01
mean	-1.850372e-17	4.218847e-16	1.591320e-16
std	1.008439e+00	1.008439e+00	1.008439e+00
min	-2.201088e+00	-2.870371e+00	-2.841506e+00
25%	-5.506751e-01	-6.288193e-01	-8.207221e-01
50%	-1.676396e-01	3.510590e-01	2.428486e-01
75%	5.836475e-01	7.641450e-01	8.809911e-01
max	2.820037e+00	1.491048e+00	1.306419e+00

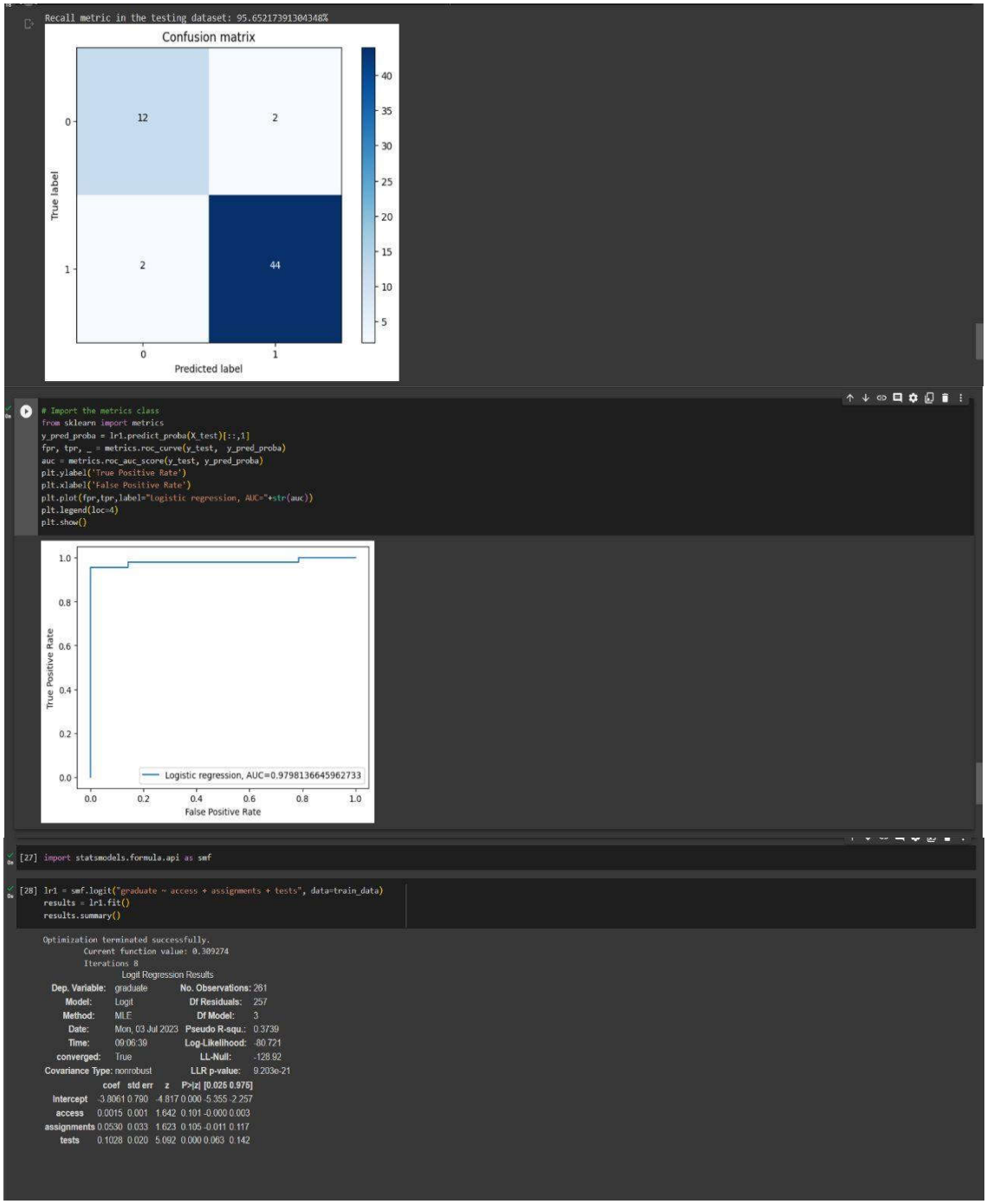
```
# Summarize class distribution
counter = Counter(y_train)
print(counter)

for label, _ in counter.items():
  row_ix = where(y_train == label)[0]
  plt.scatter(X_train[row_ix, 0], X_train[row_ix, 1], label=str(label))
plt.legend()
plt.show()
```

```
Counter({1: 210, 0: 51})
```




```
[17] for label, _ in counter.items():
       row_ix = where(y_train_res == label)[0]
       plt.scatter(X_train_res[row_ix, 0], X_train_res[row_ix, 1], label=str(label))
     plt.legend()
     plt.show()
```





CONCLUSION

The whole idea of this project is to use Machine learning approaches in understanding the problems students face to keep their education at pace. If a university is able to help a student in even a single way, it will help make a big change in a student's life. This model is a tool for the universities to make this change possible.

The model makes the analysis part easy for EWS (Early Warning System) analyzers. Using this simple model we executed, the analyzer will be able to pose questions about different features and find the answers using the model.

Using this SVM model at the simplest level:

1. The Analyzer can get details of the students on the verge of dropping out.

2. Analysis of the top features affecting the dropout rate can be understood

FUTURE SCOPE

Predictive Models for Different Student Groups:

Customize predictive models for specific student populations, such as first-generation college students, international students, or students from underrepresented backgrounds. These tailored models can account for unique challenges and provide more accurate predictions for targeted groups.

Cross-Institution Collaboration:

Foster collaboration among universities and educational institutions to share data, best practices, and research findings related to dropout prediction and prevention. Collaborative efforts can lead to a broader understanding of dropout factors and the development of standardized frameworks.

Comparative Analysis:

Conduct comparative studies across different universities, courses, and student demographics to identify common trends and factors contributing to dropout rates. This analysis can help institutions develop targeted interventions and policies to reduce attrition.

Policy Formulation:

Dropout prediction models can provide valuable insights into the factors contributing to student attrition. Government agencies responsible for education can use this information to formulate policies that target the identified risk factors. For example, if the data shows that financial constraints are a major reason for dropout, the government can introduce scholarships or financial aid programs to support students in need.

Addition of Psychological Features:

Include psychological assessments or surveys to collect data on student's psychological characteristics such as personality traits, self-efficacy, motivation, resilience, and mental health indicators. These features can be administered periodically to gather information throughout student's academic journey.

REFERENCES

[1]. (baeldung, Multiclass Classification Using Support Vector Machines, 2022) https://www.baeldung.com/cs/svm-multiclass-classification

[2]. (Valentim Realinho, Predict students' dropout and academic success, 2021) https://archive-beta.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success

[3]. (Samuel-Njoroge, Machine Learning with Support Vector Classifier., 2022) https://github.com/Samuel-Njoroge/School_Drop_out_Prediction_with_SVMs/blob/main/School_Dropout_.ipynb

[4]. (Wagavkar, Introduction to the Correlation Matrix, 2023) https://builtin.com/data-science/correlation-matrix

[5]. (Vinayak Hegde, Higher education student dropout prediction and analysis through educational data mining, 2018) https://ieeexplore.ieee.org/document/8398887

[6]. (pypi,pandas.DataFrame,2023)https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html