Comparison of three selected ML models for predicting decision of the Dean

inż. Jan Łukomski
Faculty of Electrical Engineering
Warsaw University of Technology
Warsaw, Poland

inż. Paweł Podgórski
Faculty of Electrical Engineering
Warsaw University of Technology
Warsaw, Poland

***Abstract*—Accurately predicting the decisions made by the Dean in an academic institution holds immense significance due to its profound impact on students, faculty, and the institution as a whole. This article focuses on conducting a comparative analysis of three distinct machine learning models: Support Vector Machines (SVM), Decision Tree Classifier, and k-nearest neighbors (KNN). The primary objective is to assess the effectiveness of these models in predicting the Dean's decisions, specifically categorizing them as positive or negative. Through a meticulous examination of each model's performance, strengths, and limitations, this study aims to offer valuable insights that can optimize decision-making processes within academic institutions. The dataset employed in this study has undergone thorough data cleaning and transformation procedures, ensuring the accuracy and relevance of the chosen features.**

The best classifier in this case turned out to be KNN with an accuracy of 69%. The next was Decision Tree with score 68% of accuracy. The last one was SVM which achieved 64%.

***Index Terms*—ml, svm, knn, decision tree classifier**

I. Introduction

Predicting the decision of the Dean in an academic institution is a task of significant importance, as it can greatly influence the lives of students, faculty, and the overall direction of the institution. This article presents a comparative analysis of three carefully selected ML models - Support Vector Machines (SVM), Decision Tree Classifier, and k-nearest neighbors (KNN) - to determine their effectiveness in predicting the Dean's decision as either positive or negative. By examining the performance, strengths, and limitations of each model, this study aims to provide valuable insights that can enhance decision-making processes within academic institutions.

II. Data cleaning

A. Decision

The first step was to recognize if the decision of Dean is positive or negative. The conversion was made according to table I. A new column of type boolean was created which indicates whether decision is positive or negative. It turned out that the most numerous group of answers from the dean's office is status closed from unkown reasons as it is visible on Fig. 1. The dataset with decision column was very unbalanced what can be seen Fig. 2. The balanced dataset was needed so the number of positive was reduced to the negative decision quantity. This resulted in a balanced set of decisions as in Fig 3.

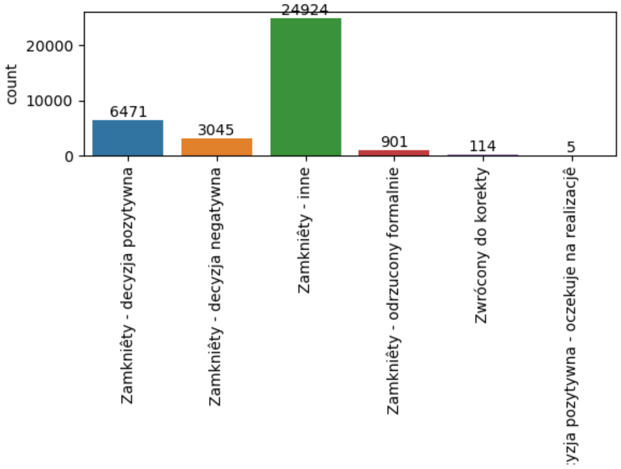

Fig. 1. Application status countplot

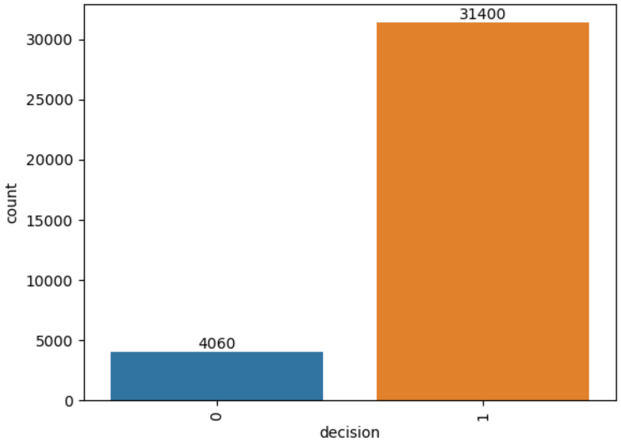

Fig. 2. Decision countplot

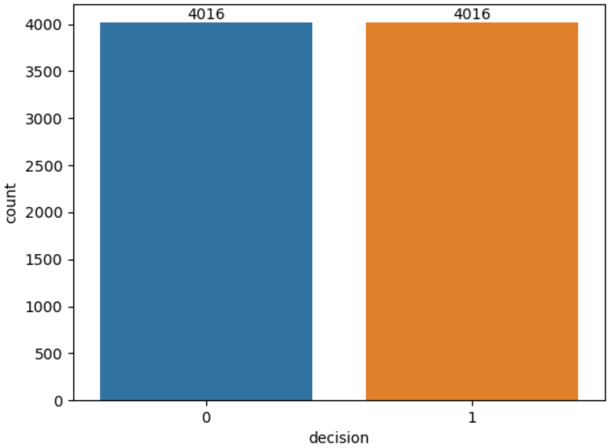

Fig. 3. Balanced decision countplot

TABLE I
CONVERT STATUS INTO DEAN’S DECISION.

positive	**negative**
Zamkniety - decyzja pozytywna	Zamkniety - decyzja negatywna
Zamkniety - inne	Zamkniety - odrzucony formalnie
Decyzja pozytywna - oczekuje na realizacje	Zwrócony do korekty

B. Data transformation

At first columns which definitely should not be significant were removed. For example ’student’ column was marked as not significant. The decision should not be performed according to ID of the person. All columns which should be significant was converted into numerical representation. There was 42 columns in dataset. Only a few of them were numerical. Other were categorical. Using *pd.factorize()* method unique values from each columns were extracted and converted into categorical codes. There was two columns that were marked as significant, but were not converted into numberical representation due to complexity of the solutions. One of those was ’Level of thesis progress’ which was not structured sentences with description about level of the thesis. The decision was made not to sink into this trying to convert it. The second column was ’Missing subjects codes’ which had a form of ‘[’A,B,C’,’C,D’]‘ where letters were subject codes.

III. SELECTING FEATURES

At the beginning a pairplot was created from all features to find some correlations. When there is so many features it is hard analyze it but it helped to select not correalted features. Drawing the image was time consuming due to the number of the columns. The result was drawn as Fig. 4. It is possible to observe that ’missing_ECTS’ and ’ECTS_in_recent_semester’ are corelated.

Then there was selected six features using the column names and intuition. The selected features was presented in the table II. The pairplot for the features is visible at Fig. 5. ”mode_of_study”, ”level_of_study”, ”specialization”, ”type_of_application” are categorical columns with limited possible option. ”semester_of_study” and ”missing_ECTS” have discrete values. In figures 6 and 7 there are boxplots of values in columns ”semester_of_study” and ”missing_ECTS”.

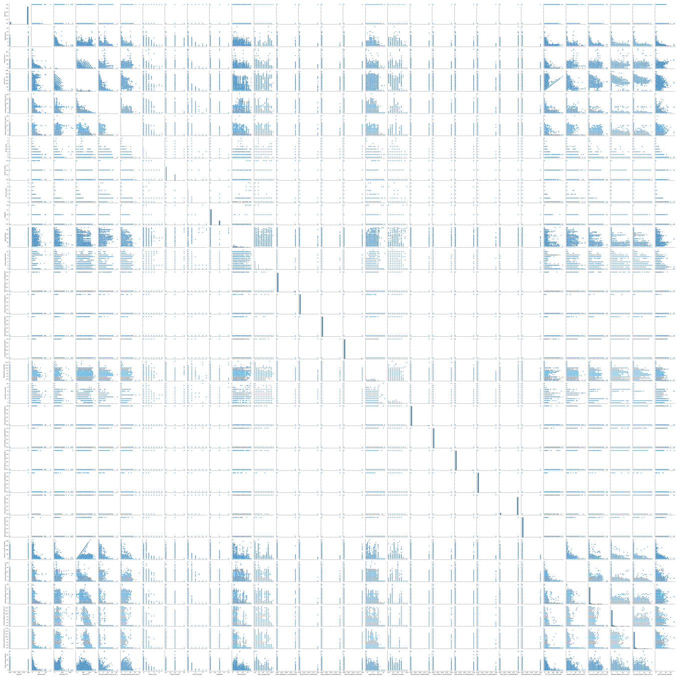

Fig. 4. Pairplot of all columns in dataset

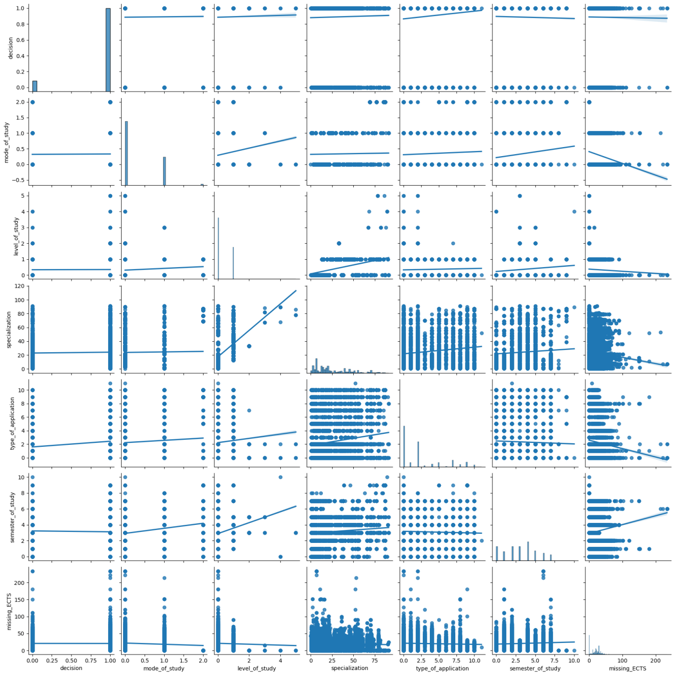

Fig. 5. Pairplot of selected columns in dataset

TABLE II
SELECTED FEATURES

Feature
mode_of_study
level_of_study
specialization
type_of_application
semester_of_study
missing_ECTS

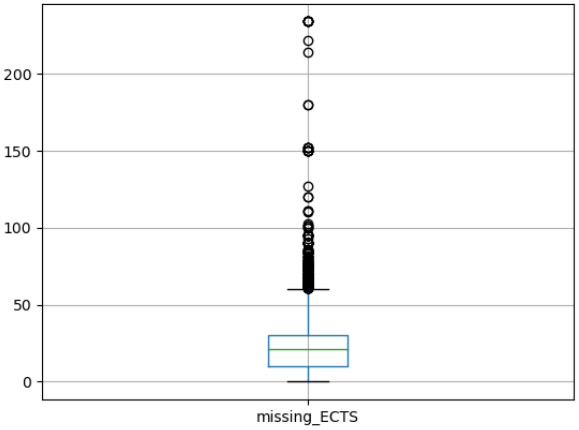

Fig. 6. Boxplot of values in missing_ECTS column

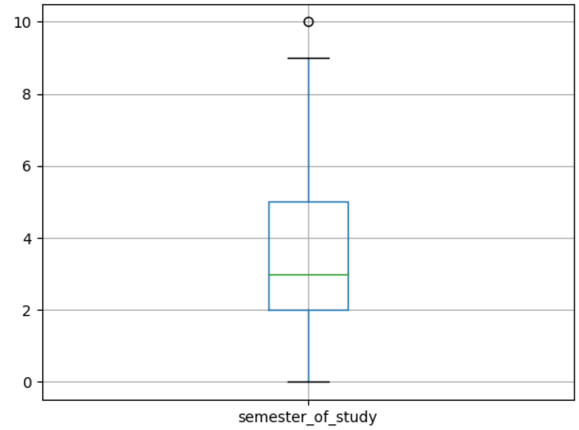

Fig. 7. Boxplot of values in ECTS_in_recent_semester column

IV. CHOOSEN ALGORITHMS

According to sklearn library [1] and other [2] recomandation, the choosen was taken to train dataset with SVM model, Decision Tree Classifier and with KNN Classificator.

A. SVM

Support Vector Machine is a supervised machine learning algorithm used for classification and regression tasks. It finds an optimal hyperplane to separate data into different classes, using techniques like the kernel trick to handle non-linearly separable data [4]. SVMs have advantages such as handling high-dimensional data and generalizing well with small datasets [3].

The basic idea behind SVM is to find an optimal hyperplane that separates the data points into different classes. A hyperplane is a decision boundary that divides the feature space into regions representing different classes. SVM aims to find the hyperplane that maximizes the margin, which is the distance between the hyperplane and the closest data points from each class. These closest data points are called support vectors [5].

B. Decision Tree Clasifier

A Decision Tree Classifier is a machine learning algorithm that creates a tree-like model to classify data [6]. It recursively splits the dataset based on features, using decision rules to determine class labels. Decision trees are interpretable and easy to understand, but may overfit and perform poorly with high-dimensional or noisy data [7].

During the training phase, the decision tree classifier learns the decision rules by fitting the data to the tree structure [8]. It continues splitting the data until a stopping criterion is met, such as reaching a maximum tree depth or when further splitting does not improve the classification accuracy.

C. KNN Classifier

K-Nearest Neighbors Classifier is a simple and versatile algorithm for classification. It assigns a class label to a new data point based on the majority vote of its k nearest neighbors [9]. KNN is easy to implement but can be computationally expensive for large datasets. It adapts well to different data types and handles multi-class classification [10].

KNN classifiers have several advantages, including their simplicity and ease of implementation. They can handle multi-class classification and can adapt to any kind of data. KNN is also a lazy learner, meaning it does not require an explicit training phase and can quickly adapt to new data. However, KNN can be computationally expensive, especially for large datasets, as it requires calculating distances for each prediction. Additionally, determining the optimal value of k and handling imbalanced data can be challenges in KNN classification [11].

V. EVALUATION

The SVM model demonstrates an accuracy, precision, and recall of 0.64. This implies that it accurately predicts the class labels for 64% of the data points. When it assigns a positive

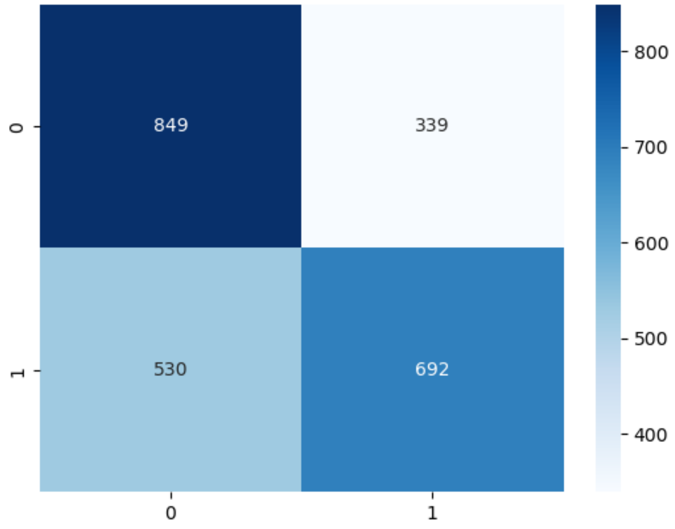

Fig. 8. SVM model prediction

TABLE III
COMPARISON OF ML MODELS

Model	Accuracy	Precision	Recall
SVM	0.64	0.64	0.64
Decision Tree Classifier	0.68	0.68	0.68
KNN Classifier	0.69	0.69	0.69

class, it is correct 64% of the time, and it successfully identifies 64% of the instances belonging to the positive class.

In the case of the Decision Tree Classifier, it achieves an accuracy, precision, and recall of 0.68. This signifies that it effectively predicts the class labels for 68% of the data points. When it assigns a positive class, it is correct 68% of the time, and it accurately identifies 68% of the positive class instances.

As for the KNN Classifier, it exhibits an accuracy, precision, and recall of 0.69. This indicates that it correctly predicts the class labels for 69% of the data points. When it assigns a positive class, it is correct 69% of the time, and it successfully identifies 69% of the positive class instances.

These metrics provide an assessment of the performance of each model in terms of their accuracy, precision, and recall, revealing their ability to correctly classify instances and predict positive class labels.

VI. SUMMARY

The SVM model achieved an accuracy, precision, and recall of 0.64, indicating that it correctly predicts the class labels for 64% of the data points. The Decision Tree Classifier performed slightly better, with an accuracy, precision, and recall of 0.68, correctly predicting the class labels for 68% of the data points. The KNN Classifier showed similar performance, with an accuracy, precision, and recall of 0.69, correctly predicting the class labels for 69% of the data points. These metrics provide insights into the models' ability to classify instances and predict positive class labels.

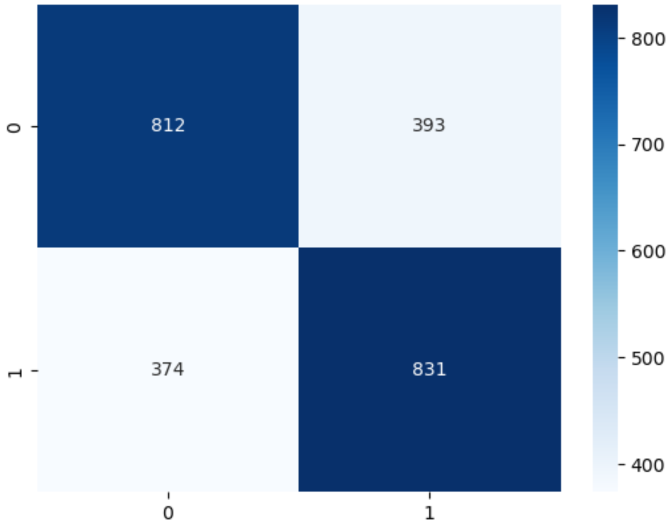

Fig. 9. Decision Tree Classifier prediction

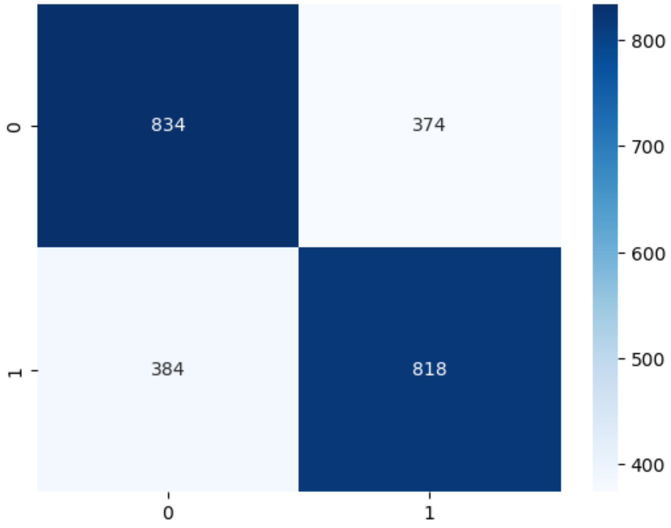

Fig. 10. KNN Classifier prediction

In summary, the KNN Classifier showed demonstrated the highest performance among the three models. These findings suggest that this classifierr may be the most effective choice for predicting the Dean's decision, considering its higher accuracy, precision and recall values.

REFERENCES

[1] https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

[2] https://towardsdatascience.com/top-10-binary-classification-algorithms-a-beginners-guide-feeacbd7a3e2

[3] Zou, H. (2019). Classification with high dimensional features. Wiley Interdisciplinary Reviews: Computational Statistics, 11(1), e1453.

[4] https://tomaszkacmajor.pl/index.php/2016/04/17/support-vector-machine/https://tomaszkacmajor.pl/index.php/2016/04/17/support-vector-machine/

[5] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5822181/

[6] https://www.ibm.com/docs/pl/spss-modeler/saas?topic=trees-decision-tree-models

[7] https://scikit-learn.org/stable/modules/tree.html

[8] Kotsiantis, S. B. (2013). Decision trees: a recent overview. Artificial Intelligence Review, 39(4), 261–283. https://doi.org/10.1007/s10462-011-9272-4

[9] Abu Alfeilat, H. A., Hassanat, A. B. A., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., & Prasath, V. B. S (2019). Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review [Doi: 10.1089/big.2018.0175]. Big Data, 7(4), 221–248. https://doi.org/10.1089/big.2018.0175

[10] Tsoumakas, G., Katakis, I., & Vlahavas, I. (2006, September). A review of multi-label classification methods. In Proceedings of the 2nd ADBIS workshop on data mining and knowledge discovery (ADMKD 2006) (pp. 99-109).

[11] Syaliman, K. U. (2021). Enhance the Accuracy of K-Nearest Neighbor (K-Nn) for Unbalanced Class Data Using Synthetic Minority Oversampling Technique (Smote) and Gain Ratio (Gr). INFOKUM, 10(1), 188-195.