Selenge Tuvshin, Iftikhar Fakhar, Haider Tobias Abraham
Machine Learning
19th of November 2023

Exercise 1

Classification experiments

Introduction

In this exercise, various models are built for four different classification problems. The following sections describe the process of cleaning the used datasets as well as training of different classifiers. With the resulting performance metrics, a comparison of the best and worst classifiers for each dataset is made to be able to determine the difficulty of the classification tasks and the fitness of the applied classifiers for those tasks.

Dataset 1: Absenteeism at work

The first dataset used represents notes taken by the HR department about the absences of mail couriers in Brazil. Many behavioral, physical and social features about employees are collected and attached to information about the absences of employees. As a target, either the hours of absence or the absence reason can be chosen. This results in either a regression task or classification task. For the purpose of this assignment, the latter is used as a prediction target.

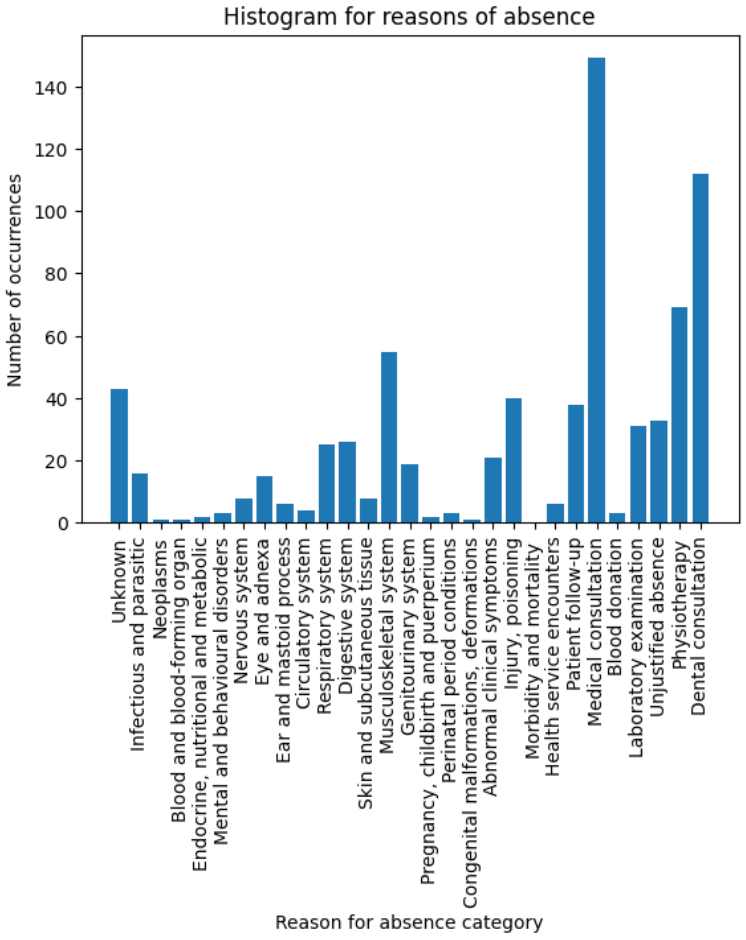
Column Name	**Min value**	**Max value**
Reason for absence	0.000	28.000
Month of absence	0.000	12.000
Day of the week	2.000	6.000
Seasons	1.000	4.000
Transportation expense	118.000	388.000
Distance from Residence to Work	5.000	52.000
Service time	1.000	29.000
Age	27.000	58.000
Work load Average/day	205.917	378.884
Hit target	81.000	100.000
Disciplinary failure	0.000	1.000

Column Name	Min value	Max value
Reason for absence	0.000	28.000
Month of absence	0.000	12.000
Day of the week	2.000	6.000
Seasons	1.000	4.000
Transportation expense	118.000	388.000
Distance from Residence to Work	5.000	52.000
Service time	1.000	29.000
Age	27.000	58.000
Work load Average/day	205.917	378.884
Hit target	81.000	100.000
Education	1.000	4.000
Son	0.000	4.000
Social drinker	0.000	1.000
Social smoker	0.000	1.000
Pet	0.000	8.000
Weight	56.000	108.000
Height	163.000	196.000
Body mass index	19.000	38.000
Absenteeism time in hours	0.000	120.000

Pre-processing

Although the dataset seems very clean and usable as is, some kind of preprocessing and interpretation is required for most features.
The target variable is encoded as integers and represents one of 28 classes defined by the company as absence reasons.

It is, like other categorical variables like month, day of week, season and education, mapped to its string representation and defined as a categorical data type.
Many boolean features are encoded as 0 and 1. To prevent models from misinterpreting them as continuous variables, they are explicitly set to true and false respectively.
Some suspected data input entries were removed. This is the case for the month of absence, which should only contain values from 1 to 12, but has some 0 values in the dataset. As seen in the histogram for the reason of absence above, missing values are kept and encoded as an unknown class.



Classifiers

Random Forest

This classifier is an ensemble learning method. It constructs multiple decision trees during training which are all different because of some randomness in the fitting algorithm. It is a simple and robust algorithm that can often be directly applied to various datasets.
The main training parameter is the number of trees. (*Sklearn.ensemble.RandomForestClassifier — Scikit-Learn 1.3.2 Documentation*, n.d.)

K-nearest neighbors

KNN is a lazy learning algorithm that simply classifies samples as the majority of the nearest neighbors. It works well for datasets with non-linear decision boundaries and requires little tuning. The main model parameter is the number of neighbors chosen for classification. (*Sklearn.neighbors.KNeighborsClassifier — Scikit-Learn 1.3.2 Documentation*, n.d.)

Support vector machine

SVC is a method of defining decision hyperplanes to separate classes in the feature space. It is highly performant with many input features. It has many optional model parameters such as kernels, regularization methods that allow the training of nonlinear boundaries and increase robustness. (*1.4. Support Vector Machines — Scikit-Learn 1.3.2 Documentation*, n.d.)

Performance

The performance of all classifiers is rather bad. This is the case, because the learning problem is hard and many features that could explain the reason for absence are not present as input features. Also, the target has 28 different classes where some are not present at all in the data.

Classifier/ Performance metric	Random Forest trees = 90	KNN n = 4	SVC
Accuracy	0.331522	0.277174	0.288043
Precision	0.322665	0.235118	0.255435
Recall	0.331522	0.277174	0.288043
F1 Score	0.312193	0.237398	0.204382

All in all, the random forest classifier seems to perform best. It is able to learn some absenteeism patterns of the individuals (employees) from which the samples are drawn and is able to predict some absence reasons with impressive accuracy. However, it seems obvious that high accuracy can only be achieved in this dataset by overfitting and generalization of the model to new data would result in very poor accuracy. This becomes visible when looking at the cross validation accuracy scores of the random forest classifier, which are much higher than the accuracies for the test set in the end. The mean of the prediction accuracy is 42 % while the test accuracy is only 33 %.

CV	fold 1	fold 2	fold 3	fold 4	fold 5
Accuracy for each fold	0.459459	0.387387	0.427273	0.400000	0.400000

It can be expected that data, as it is recorded in the absenteeism dataset and resulting models, would not work for other companies, even in the same sector.

Dataset 2: Predict students' dropout and academic success

The dataset on student dropout and academic success is a comprehensive collection of data points covering various aspects of a student's academic journey and background. It includes personal information such as marital status, gender, and age at enrollment, as well as academic details like the course of study, attendance type and previous qualifications with grades. Additionally, it encompasses socioeconomic indicators such as the student's nationality, parents' qualifications and occupations, and financial status including tuition fee payments and scholarship holdings. Academic performance is detailed across semesters, including credits, enrollments, evaluations, approvals, and grades. The dataset also integrates broader economic factors like unemployment rates, inflation, and GDP, potentially to study their impact on academic outcomes. The target variable in this dataset likely represents the student's academic status, indicating whether they have dropped out or succeeded, making it a valuable resource for analyzing and predicting factors influencing student dropout and academic success.

Pre-processing

In this preprocessing step, the code is checking for missing values in the dataset named students_dropout_and_academic_success. It first creates a boolean mask (missing_values) that identifies whether each entry in the dataset is null (True) or not (False). The subsequent line sums up these boolean values for each column, providing a count of missing values in each respective column.

```
Marital status                                  0
Application mode                                0
Application order                               0
Course                                          0
Daytime/evening attendance                      0
Previous qualification                          0
Previous qualification (grade)                  0
Nacionality                                     0
Mother's qualification                          0
Father's qualification                          0
Mother's occupation                             0
Father's occupation                             0
Admission grade                                 0
Displaced                                       0
Educational special needs                       0
Debtor                                          0
Tuition fees up to date                         0
Gender                                          0
Scholarship holder                              0
Age at enrollment                               0
International                                   0
Curricular units 1st sem (credited)             0
Curricular units 1st sem (enrolled)             0
Curricular units 1st sem (evaluations)          0
Curricular units 1st sem (approved)             0
Curricular units 1st sem (grade)                0
Curricular units 1st sem (without evaluations)  0
Curricular units 2nd sem (credited)             0
Curricular units 2nd sem (enrolled)             0
Curricular units 2nd sem (evaluations)          0
Curricular units 2nd sem (approved)             0
Curricular units 2nd sem (grade)                0
Curricular units 2nd sem (without evaluations)  0
Unemployment rate                               0
Inflation rate                                  0
```

Fortunately, the result indicates that there are no missing values in any of the columns.

Classifiers

RandomForestClassifier

It specifies 90 decision trees (`n_trees`) in the ensemble, and the classifier is trained on the training data (`X_train`, `y_train`). Predictions are then made on the test data (`X_test`), and performance metrics, including overall accuracy and per-class accuracy, are calculated. The Random Forest model is known for its robustness and ability to handle complex datasets, making it a popular choice for classification tasks. The specified parameters, such as the number of trees, impact the model's performance and are crucial for achieving accurate predictions. (*Sklearn.ensemble.RandomForestClassifier — Scikit-Learn 1.3.2 Documentation*, n.d.)

Stochastic Gradient Descent (SGD)

The code utilizes a Stochastic Gradient Descent (SGD) classifier for a classification task. The SGD classifier is trained on X_train and y_train, and predictions are made on X_test. Performance metrics, encompassing overall accuracy and per-class accuracy, are then computed.

The choice of n_neighbors is pivotal for the classifier's behavior and its ability to discern patterns in the data. *(Stochastic Gradient Descent (SGD) Classifier, n.d.)*

Neural Network Classification

The classifier is instantiated with default settings and trained on X_train and y_train. Predictions are made on X_test, and performance metrics are computed using a custom function, likely calculate_performance_metrics. Fine-tuning hyperparameters, such as the neural network architecture and learning rate, is crucial for optimizing the model's performance in capturing intricate patterns within the data. *(Types of Neural Networks and Definition of Neural Network, n.d.)*

Performance

These metrics provide a comprehensive overview of the classifiers' performance. The RandomForestClassifier exhibits the highest accuracy and F1 score, indicating a robust overall performance. On the other hand, the Stochastic Gradient Descent classifier appears to have lower performance across all metrics, suggesting potential challenges in capturing patterns within the data. The MLP classifier falls in between, showcasing a moderate level of accuracy and precision. Understanding these metrics helps in selecting the most suitable classifier for the given classification task and can guide further optimization efforts.

Classifier/ Performance metric	RandomForestClassifier	Stochastic Gradient Descent	MLP
Accuracy	0.768535	0.521700	0.572333
Precision	0.749933	0.498813	0.693209
Recall	0.768535	0.521700	0.572333
F1 Score	0.750671	0.377824	0.598548

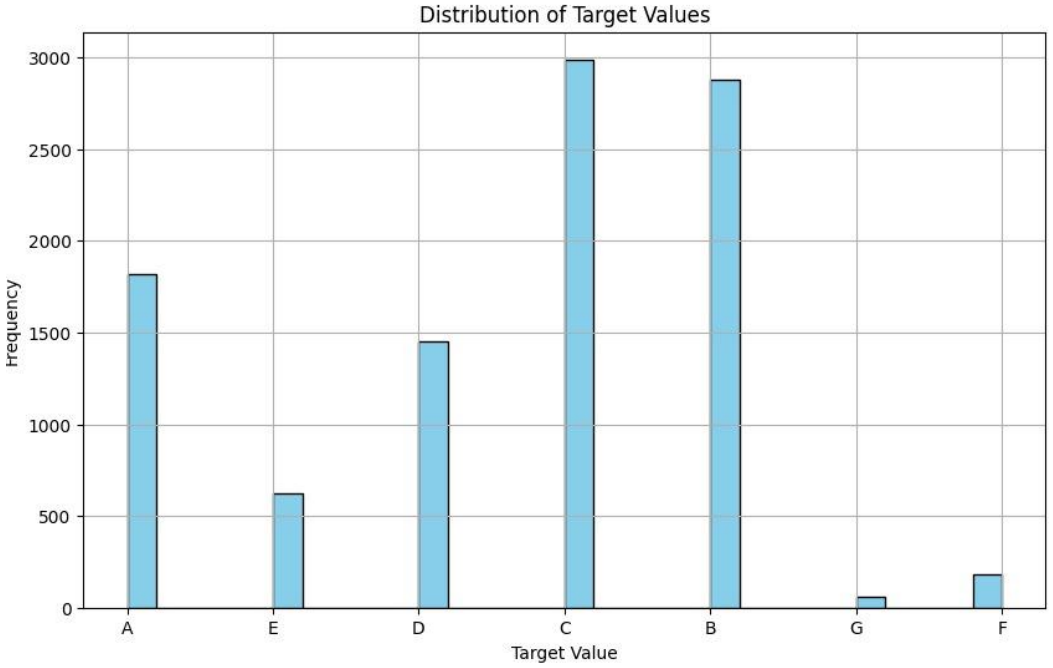
Cross-validation for the best performing classifier in the students' dropout and academic success dataset: RandomForestClassifier

CV	fold 1	fold 2	fold 3	fold 4	fold 5
Accuracy for each fold	0.7936747	0.78614458	0.76656627	0.76168929	0.76621418

Dataset 3: Loan

Description

The loan dataset from Kaggle, containing 90 columns, offers a detailed representation of various loan attributes. Among these columns, “loan_amnt” records the amount of each loan, which varies from 1,000 to 40,000. Additionally, the dataset includes categorical variables, such as the loan term options of 36 months and 60 months. These columns are utilized as features for classification models. The dataset provides target values in the “grade” column, where each loan is assigned a grade. This dataset is particularly robust for machine learning purposes as it is complete, with no missing values, and is also used for a Kaggle competition.



Pre-Processing

Our data processing and model training process began with an in-depth analysis of the distribution of the target value, giving us a clear understanding of the data structure. Before we pre-processed the data we divided the Kaggle training dataset into distinct training and validation subsets, ensuring a robust model evaluation later on.

During the data cleaning phase, we found no missing values in either the training or validation datasets, eliminating the need for any data removal or imputation. To enhance the interpretability of the data, we re-mapped certain values to more intuitive ones. This included converting the n/N or y/Y values in the "pymnt_plan", "debt_settlement_flag", and "hardship_flag" columns to TRUE or FALSE values. Similarly, we replaced the “w” and ”f” values in the “initial_list_status" column with “whole loan” and “fractional loan”, respectively.

Next, we addressed the issue of scale in our numerical features. We first identified the integer-type variables, and then applied the StandardScaler from scikit-learn to normalize those continuous variables that exhibited significant range differences.

For the boolean variables in our dataset, we transformed them into dummy variables using the OneHotEncoder from scikit-learn. This step was crucial to ensure that these categorical features could be effectively utilized by our machine learning algorithms. Finally, we trained our data using various classifiers, marking the culmination of our rigorous data processing and preparation process. This comprehensive approach allowed us to build a model that is both robust and interpretable.

Classifiers

Perceptron

The perceptron is a fundamental type of artificial neuron and the simplest form of a neural network, primarily used for binary classification tasks in supervised learning. It operates by taking multiple binary inputs, each multiplied by a weight, and then summing these weighted inputs. This sum is passed through a step function, a type of activation function, to yield a single binary output. This model exemplifies one of the earliest algorithms in machine learning. (*Perceptron Definition*, n.d.)

Gaussian Naive Bayes

Naive Bayes, a classification algorithm in machine learning, is grounded in Bayes theorem and known for its simplicity and effectiveness in a wide range of applications. The Gaussian Naïve Bayes variant is a specific adaptation that assumes features follow a normal distribution. Unlike other Naïve Bayes models that might use different methods to estimate data distribution, Gaussian Naïve Bayes simplifies this by calculating the mean and standard deviation of the training data, making it particularly straightforward and efficient for handling continuous data. (Vats, 2021)

Decision Tree

A decision tree is a popular machine-learning algorithm used for both classification and regression tasks. It's a flowchart-like structure where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome. The topmost node in a decision tree is known as the root node. The decision tree algorithm tries to solve the problem by using tree representation, where it breaks down a dataset into smaller and smaller subsets while at the same time developing an associated decision tree. The final decision tree can be used to predict the outcome for unseen instances. (Chakure, 2022)

Performance

In evaluating the Perceptron, Gaussian Naive Bayes, and Decision Tree classifiers, significant performance discrepancies were noted. The Perceptron and Gaussian Naive Bayes yielded subpar results in key metrics such as accuracy, precision, recall, and F1 score, indicating a potential mismatch with the dataset. In contrast, the Decision Tree classifier achieved remarkably high performance metrics, reaching 0.98. This initially raised concerns about overfitting. However,

when subjected to 5-fold cross-validation, the Decision Tree maintained a high mean accuracy of 0.98, with a very low standard deviation of 0.004 across folds. This consistency suggests that the Decision Tree classifier is indeed well-suited for the loan dataset, providing reliable predictions.

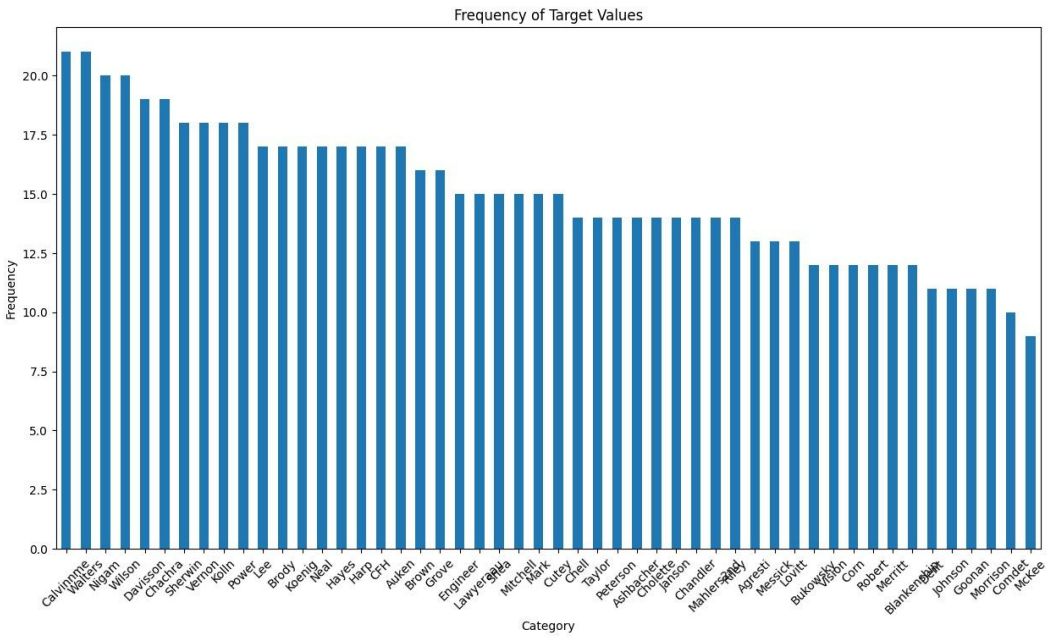
Classifier/ Performance metric	Perceptron	Gaussian Naive Bayes	Decision Tree
Accuracy	0.255200	0.240000	0.985200
Precision	0.148889	0.276171	0.986581
Recall	0.255200	0.240000	0.985200
F1 Score	0.159871	0.160899	0.985577

Cross-validation for the best performing classifier in the reviews dataset:
Decision Tree

CV	fold 1	fold 2	fold 3	fold 4	fold 5
Accuracy for each fold	0.97266667	0.98533333	0.98333333	0.98533333	0.97866667

Dataset 4: Reviews

Description



The reviews dataset, sourced from Kaggle, consists of a large matrix with 750 rows and 10,000 columns. Each of these columns is a continuous variable, with values ranging from 0 to 32, indicating a wide variety of data points. The dataset also includes a categorical “class” column, distinguished by different words as we visualized in our figure above.

Pre-processing

In the preprocessing phase, we proceeded to split the data into training and test sets, ensuring that we had separate data for model training and validation.

A notable aspect of this dataset is the absence of missing values, which streamlined the preprocessing stage. The dataset comprises a mix of continuous variables and categorical class information as our target, presenting a comprehensive yet potentially complex scenario for analysis. The minimal data sparsity in our dataset eliminated the need for using any scaling techniques. Furthermore, since all variables were numerical, there was no requirement for one-hot encoding or the creation of dummy variables for categorical data.

With the preprocessing step complete, we directly moved towards training our models. This approach allowed us to efficiently handle the data without overcomplicating the preprocessing pipeline.

Classifiers

Neural Network Classification (Multi-Layer Perceptrons)

Multi-Layer Perceptrons (MLPs) are a type of neural network known for their depth and complexity, featuring multiple layers of neurons. They are particularly effective in tasks like speech recognition and machine translation due to their fully connected architecture, where each neuron in one layer is connected to all neurons in the subsequent layer. An MLP consists of input, output, and multiple hidden layers. It employs a process called backpropagation for learning, adjusting weights based on the error in predictions. Nonlinear activation functions in these layers allow MLPs to capture complex patterns, often culminating in a softmax function in the output layer for classification tasks. (*Types of Neural Networks and Definition of Neural Network*, n.d.)

Stochastic Gradient Descent

Stochastic Gradient Descent (SGD) is a variant of the traditional gradient descent algorithm, widely used in machine learning for optimizing models like support vector machines, logistic regression, and neural networks. Differing from standard gradient descent, which computes the gradient using all data points, SGD updates parameters using the gradient from just one randomly selected data point at a time. This approach significantly speeds up the learning process, particularly in large-scale and sparse data scenarios common in text classification and Natural Language Processing. SGD's efficiency and effectiveness make it a go-to method, especially when combined with backpropagation for neural network training. (*Stochastic Gradient Descent (SGD) Classifier*, n.d.)

Nearest Centroid Classifier

The Nearest Centroid Classifier, also known as the Nearest Prototype Classifier, is a straightforward machine learning method for classifying observations. It works by assigning a data point to the class whose centroid, or average point, is nearest to it. This approach is particularly effective in text classification tasks, where documents are represented as word vectors with weights. (*Nearest Centroid Classifier*, n.d.)

Performance

In the comparative analysis of the Neural Network (MLP), Stochastic Gradient Descent (SGD), and Nearest Centroid Classifier, it was observed that none of these classifiers achieved a high level of performance in predicting the target value. Among them, SGD showed the most promise, yet its overall effectiveness was still limited. This was further explored through a 5-fold cross-validation of the SGD classifier, which yielded a mean accuracy of just 0.34. Notably, a significant drop in accuracy to 0.15 in the fifth fold, coupled with a high standard deviation of 0.10, points to potential issues in model stability and consistency. This suggests that the model's performance was notably uneven across different subsets of the data, an issue that requires further investigation and possibly refinement of the model or its training process.

Classifier/ Performance metric	Neural Network Classification (Multi-layer Perceptron)	Stochastic Gradient Descent (SGD) Classification	Nearest Centroid Classifier
Accuracy	0.436170	0.473404	0.218085
Precision	0.447453	0.567675	0.339010
Recall	0.436170	0.473404	0.218085
F1 Score	0.404578	0.464175	0.232033

Cross-validation for the best performing classifier in the reviews dataset:
Stochastic Gradient Descent (SGD)

CV	fold 1	fold 2	fold 3	fold 4	fold 5
Accuracy for each fold	0.40707965	0.33628319	0.375	0.44642857	0.15178571

Comparison

Dataset/Classifier	Abenteeism at work		Student dropouts		Loans		Reviews	
Best classifier	0.33	Random Forest	0.76	Random Forest	0.98	Decision Tree	0.47	SGD
Worst classifier	0.27	KNN	0.52	SGD	0.24	Gaussian Naive Bayes	0.21	Nearest Centroid Classifier

For the absenteeism at work dataset, the Random Forest classifier emerges as the most effective with a score of 0.33, indicating its moderate success in handling the complexities of this particular dataset. Its ability to ensemble multiple decision trees likely contributes to its superior performance over the KNN classifier, which scores the lowest at 0.27. The KNN's less effective performance might be due to its reliance on proximity-based methods, which may not be optimal for this dataset's characteristics.

For the students' dropout and academic success dataset, the Random Forest classifier shows remarkable efficacy with a score of 0.76. This high performance is likely due to Random Forest's ability to handle the complexities and variabilities associated with educational data, which often includes a mix of numerical and categorical variables. In contrast, the SGD (Stochastic Gradient Descent) classifier scores lower at 0.52, indicating a lesser degree of effectiveness. The SGD classifier's linear approach might not capture the nuances and interdependencies of factors influencing student dropouts as effectively as the Random Forest's ensemble method.

In the loans dataset, the Decision Tree classifier stands out with an impressive score of 0.98. This high accuracy can be attributed to the Decision Tree's ability to break down a large dataset into simpler pieces, making it effective for categorical data prevalent in loan-related information. On the contrary, the Gaussian Naive Bayes classifier, scoring only 0.24, struggles possibly due to its assumption of feature independence, which is often not the case in complex financial datasets.

The reviews dataset, characterized by over 10.000 features with continuous variables and categorical target values, sees the SGD (Stochastic Gradient Descent) classifier as the top performer with a score of 0.47. The SGD classifier's robustness and efficiency in handling large-scale and continuous data make it well-suited for this dataset. In stark contrast, the Nearest Centroid Classifier, with a score of 0.21, falls short in effectiveness, likely due to its simplistic approach that struggles with high dimensionality and continuous nature of the data.

This comprehensive analysis highlights that the effectiveness of classifiers is heavily influenced by the dataset's nature and structure. While Random Forest and Decision Tree excel in certain scenarios, their superiority isn't universal. Similarly, the underperformance of classifiers like KNN, Gaussian Naive Bayes, and Nearest Centroid in these contexts does not diminish their potential applicability in other types of datasets, underscoring the importance of choosing the right classifier based on specific dataset characteristics.

Summary

In conclusion, this analysis offers insightful observations on the suitability of various classifiers for distinct types of datasets. It underscores the critical importance of aligning classifier choice with the specific characteristics and complexities of each dataset. The performance disparities among classifiers like Random Forest, Decision Tree, and SGD across different datasets highlight the need for careful consideration in model selection. This research serves as a valuable guide and emphasizes the necessity of a tailored approach in classifier selection to achieve optimal results.

Individual contributions from team members

Tuvshin:

- Code and report for dataset 3
- Code and report for dataset 4
- Comparison
- Summary

Tobias:

- Jupyter notebook template
- Code and report for dataset 1

Fakhar:

- Code and report for dataset 2

References

Chakure, A. (2022, February 10). *Components of Decision Tree Classification*. Built In. Retrieved November 18, 2023, from https://builtin.com/data-science/classification-tree

Martiniano, A., & Ferreira, R. (2018, April 4). *Absenteeism at work*. UCI Machine Learning Repository. Retrieved October 8, 2023, from https://archive.ics.uci.edu/dataset/445/absenteeism+at+work

Nearest centroid classifier. (n.d.). Wikipedia. Retrieved November 18, 2023, from https://en.wikipedia.org/wiki/Nearest_centroid_classifier

1.4. Support Vector Machines — scikit-learn 1.3.2 documentation. (n.d.). Scikit-learn. Retrieved November 19, 2023, from https://scikit-learn.org/stable/modules/svm.html

Perceptron Definition. (n.d.). DeepAI. Retrieved November 18, 2023, from https://deepai.org/machine-learning-glossary-and-terms/perceptron

Predict students' dropout and academic success. (n.d.). UCI Machine Learning Repository. Retrieved October 8, 2023, from https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success

sklearn.ensemble.RandomForestClassifier — scikit-learn 1.3.2 documentation. (n.d.). Scikit-learn. Retrieved November 19, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

sklearn.neighbors.KNeighborsClassifier — scikit-learn 1.3.2 documentation. (n.d.). Scikit-learn. Retrieved November 19, 2023, from https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Stochastic Gradient Descent (SGD) Classifier. (n.d.). The Click Reader. Retrieved November 18, 2023, from https://www.theclickreader.com/stochastic-gradient-descent-sgd-classifier/

Types of Neural Networks and Definition of Neural Network. (n.d.). Great Learning. Retrieved November 18, 2023, from https://www.mygreatlearning.com/blog/types-of-neural-networks

Vats, R. (2021, February 21). *Gaussian Naive Bayes: What You Need to Know?* upGrad. Retrieved November 18, 2023, from https://www.upgrad.com/blog/gaussian-naive-bayes