Cerdika: Jurnal Ilmiah Indonesia, February 2025, 5 (2), 515-530
p-ISSN: 2774-6291 e-ISSN: 2774-6534

Available online at http://cerdika.publikasiindonesia.id/index.php/cerdika/index

MACHINE LEARNING-BASED CLASSIFICATION FOR SELECTION OF SMART SCHOLARSHIP FEE AMOUNTS IN THE CAPITAL CITY

Wiza Teguh
Universitas Bina Nusantara, Indonesia
Email: wiza.teguh@binus.ac.id

Abstract
In the management of scholarship funds such as the Jakarta Smart Card (KJP), challenges related to accuracy and efficiency are often faced, so the application of machine learning methods is expected to improve the results of data processing of scholarship recipients. This research aims to improve the accuracy of the KJP scholarship fund receipt process by applying a machine learning-based classification method. The KJP Scholarship has an important role in supporting education in Jakarta, but there is a need to improve the distribution process and the accuracy of fund distribution. By utilizing machine learning techniques, this research focuses on processing scholarship recipient data to produce better decisions. This study uses a quantitative approach with a data analysis method, with the population of all KJP scholarship recipients in 2023 and a random sample taken from the available data. Data was collected through documentation and processing of historical data of scholarship recipients, with analysis using the SEMMA (Sample, Explore, Modify, Model, Assess) approach as well as Decision Tree, Naïve Bayes, and Random Forest algorithms. The results of the analysis show that the Decision Tree model provides the best performance with an accuracy of 88.31%, compared to Naïve Bayes and Random Forest. These findings provide new insights for better decision-making in the distribution of KJP scholarships, as well as demonstrate the potential for the integration of machine learning in scholarship management in the education sector, which can improve the efficiency and accuracy of fund management.

Keywords: Jakarta Smart Card, Machine Learning, Classification, SEMMA, Decision Tree, Naïve Bayes, Random Forest.

**Correspondence Author: Wiza Teguh*
Email: wiza.teguh@binus.ac.id

INTRODUCTION

Education plays a big role in the modern industrial world as an important aspect that society needs to survive in a competitive world (Prasad & Pushpa Gupta, 2020). Education is also a way to improve the quality of human resources and is a form of investment in human capital. According to BPS data for DKI Jakarta Province (2021), the level of education is related to the poverty rate. Currently, education in Indonesia is not spread evenly throughout Indonesia, supported by Sunarti (2022) in other research, this problem is caused by poverty, lack of quality of human resources, low quality of education in Indonesia, low educational services, low literacy skills of Indonesian children (Meriyanti & Jasmina, 2022; Sunarti et al., 2022).

DKI Jakarta Province, as one of the metropolitan cities in Indonesia, also experiences inequality in terms of education (Muhaimin et al., 2022). Education in the DKI Jakarta area is still far from real expectations because there are still many children dropping out of school due to problems with their parents' limited ability to meet education costs. In realizing the process of equalizing education in the DKI Jakarta area with a 12-year compulsory education program, the DKI Jakarta government created a policy by issuing a program, namely Kartu Jakarta Pintar (KJP).

Kartu Jakarta Pintar (KJP), which is entirely sponsored by DKI Jakarta Provincial APBD funds, is a strategic program designed to give to underprivileged communities in

DKI Jakarta access to education up to and including high school or vocational school completion. Meanwhile, the KJP program was updated to KJP Plus in 2018 following the leadership of DKI Jakarta by Anies Rasheed Baswedan and Sandiaga Uno. The goal of KJP Plus was to broaden and update the benefits of Kartu Jakarta Pintar for all school-age children (6-21 years). In addition, it can be utilized for skills training, Madrasah education, Islamic boarding schools, study groups A, B, and C, and it has financial aid available for low-income families.

Since the implementation of KJP Plus, the growth and development of the school has been carried out well, so that it can improve the quality of students who will be the nation's successors in the future (statistik.jakarta.go.id, 2021). The determination of the quota for the Personal Education Fee Assistance program in the form of KJP Plus is based on standard things, namely the proportion of regions, the number of schools, and the number of students. In terms of the number of poor students in an area, a larger percentage will get priority, but this depends on the accuracy of the available data.

The DKI Jakarta Provincial Government carries out repeated verification to ensure the eligibility status of KJP Plus recipients. Currently, the KJP Plus selection and verification process has been carried out through various methods such as conducting interviews and direct visits to the residences of prospective recipients. One method that can be used to help make decisions in determining potential KJP Plus recipients is to use machine learning methods, namely by classifying existing KJP Plus selection data from previous years (Ningsih & Hardiyan, 2020). The classification process can be carried out using various algorithms in machine learning, including Naïve Bayes, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree, and Random Forest.

Several similar studies have been conducted by other researchers, such as the Classification of Jakarta Smart Card (KJP) Recipients using the ID3 algorithm method which states that home ownership criteria have the greatest role in decision making followed by the type of parent's job (Merdekawati & Kumalasari, 2022). On the other hand, research by Retnani Latifah et. al. who conducted research on determining prospective recipients of the Jakarta Smart Card (KJP) using the K-Nearest Neighbor (KNN) algorithm. Although the accuracy results obtained from this research are quite high, the data used is still too little and needs to be added (Prihatin et al., 2021). In addition, the Support Vector Machine (SVM) technique is a reliable algorithm for classification and regression that can accurately identify subtle patterns in complicated data sets which was also used by Haryanto and Hidayatullah for the Jakarta Smart Card (KJP) Fund Receipt in one of Jakarta's elementary schools (Haryanto & Hidayatullah, 2016).

Through gap analysis of existing studies, this study will use different algorithms to obtain broader research results, namely Naïve Bayes, Decision Tree, and Random Forest. The researcher will compare the three algorithms by adjusting the existing data appropriately to achieve more accurate and precise modeling in classifying Jakarta Smart Card recipients. In addition, this classification aims to determine the difference in the amount of funds received for each level of education and region, considering the difference in the amount of KJP funds received even with the same level of education and region. This research also aims to explore the feasibility of prospective KJP recipients to be more targeted and efficient.

RESEARCH METHODS

This research will adopt the SEMMA (Sample, Explore, Modify, Model, and Assess) approach as its core methodology. The first stage involves selecting an appropriate dataset, followed by exploring and visualizing the data. The next step is the modification of the dataset to ensure its readiness for the modeling process. The modeling itself will be carried out using various machine learning algorithms. Evaluation of the developed model is done by measuring its accuracy (Asriyanik & Pambudi, 2023). To validate the model, this research will apply the K-fold crossvalidation method. This aims to test the reliability of the model performance and minimize the potential for overfitting on the training data (Dutschmann et al., 2023). Model performance will be assessed using a confusion matrix table, which allows the calculation of evaluation metrics such as accuracy, precision, recall, and F1 value. Accuracy here is defined as the proportion of correctly classified instances to the total classified instances. Recall relates to how effectively the algorithm identifies a particular class, while precision relates to the classification accuracy of the entire dataset. The F1 value is a combination of recall and precision, reflecting the overall effectiveness of the method (Plotnikova et al., 2023).

Data Collection

This research was initiated with the important step of collecting a dataset of Jakarta Smart Card (KJP) recipients in Jakarta for the year 2023 as the main basis for modeling (Raja & Adlan, 2022). This process involved not only thorough data collection but also careful data screening to ensure the quality and relevance of the information collected. This data screening is important to ensure that the dataset to be used in this study is truly representative of the target population and free from distortion or bias (Martindale et al., 2020). Once the required datasets have been collected and screened, the next step is data analysis which aims to gain deep and meaningful insights in line with the research objectives. This analysis process involves applying sophisticated statistical and data modeling techniques to understand patterns and trends in the data, which will be very useful in formulating appropriate policy recommendations or interventions (Baillie et al., 2022). The results of this data selection and analysis will be the foundation for the next step of the research, which will focus more on the implementation and evaluation of policies related to KJP recipients in Jakarta, with the ultimate goal of improving the effectiveness and efficiency of this program in providing benefits to the community.

Data Exploration: Descriptiond and Visualization

Exploration of the dataset through sophisticated data visualization techniques and in-depth statistical descriptions plays a key role in this research process. This process not only helps in understanding the general characteristics and distribution of the data, but also in identifying patterns, trends, and anomalies that may exist. Data visualization allows us to see relationships and patterns that may not be immediately apparent through statistical analysis alone. Statistical descriptions provide valuable quantitative views, such as mean, median, mode, and standard deviation, all of which are important for informing the selection of data to be used in modeling (Chen et al., 2020).

Furthermore, this process is also critical in determining data quality and the need for data cleaning, such as handling missing values or outliers. The selection of appropriate

data features based on the results of this exploration is a crucial step in building a valid and reliable model. This is important because selecting the right features will increase the accuracy and effectiveness of the model in making predictions or classifications. The results of this step will be used to design and develop the model that will be tested and applied in the next phase of this research. The ultimate goal of this step is to ensure that the developed model is able to provide accurate and useful insights for decision-making related to the recipients of the Jakarta Smart Card (KJP) (Moussa & Măndoiu, 2018).

Data Modification: Data Cleaning and Data Transformation

Data transformation is an essential stage in preparing datasets for data mining analysis. This process involves the Term Frequency-Inverse Document Frequency (TF-IDF) technique, which aims to assess the importance of a word in a set of documents. TF-IDF assigns a weight to each word based on its significance in the context of the document. Inverse Document Frequency (IDF) is a part of this method that calculates the weight of a term based on how often it appears in the document set. The IDF weight of a term will decrease if the term appears more frequently in many documents (Chen et al., 2020).

Apart from data transformation, there is also another important step in data provision, which is data modification. This step includes variable selection, data cleaning, and data transformation. This process is executed to ensure that the dataset to be used in modeling is verified and ready for analysis. Feature selection involves identifying and selecting the most relevant variables, data cleaning aims to remove errors or inconsistencies, and data transformation is the process of converting data to a format more suitable for analysis. This whole process is important to ensure the quality and accuracy of the data mining analysis results (Moussa & Măndoiu, 2018).

Data Modelling using Machine Learning Alghorithm

In this research, the primary focus is on analyzing textual data through the use of advanced machine learning algorithms, specifically the Decision Tree, Naïve Bayes, and Random Forest. These methodologies are instrumental in deciphering complex patterns within text documents. The process aims at a comprehensive comparison and application of these algorithms to interpret and derive meaningful insights from the text. The ultimate objective is to leverage these insights for informed decision-making and enriching the research analysis, thereby transforming the raw text data into a valuable resource. This approach is fundamental in extracting the essence of the textual content and applying it to the research context effectively (Chen et al., 2020).

a) Decision Tree

The Decision Tree algorithm serves as a very important tool. It operates by learning from a set of independent data, represented in a tree-like model. This approach uses a 'divide and conquer' strategy, where data is divided into sections based on certain criteria, thus simplifying the problem-solving process. The effectiveness of a decision tree is encapsulated in an equation that represents the probability of a data tuple in a set D belonging to class Ci. This equation also addresses the concept of entropy in D, which measures the average information or uncertainty inherent in the predictions of the data set (Charbuty & Abdulazeez, 2021). This methodological approach in data analysis helps in breaking down complex data sets in a systematic manner, improving the understanding and classification of data such as the following equation:

$$Info\ (D) = \sum_{i=1}^{N} - p_i log_2 p_i$$

$$Info\ (D) = \sum - p_i\ n_i = 1\ log_2\ (p_i)$$

b) Naïve Bayes

Naïve Bayes is a probabilistic classification method known for its simplicity and effectiveness, especially in scenarios where the assumption of independence between features holds. The method operates based on Bayes' theorem, which describes the probability of an event, based on prior knowledge of conditions that may be associated with the event. In the context of classification, Naïve Bayes predicts the probability of a data point belonging to a particular class, given a set of features. Naïve Bayes calculates the conditional probability of each class, given each feature, and then makes predictions based on these probabilities. The classifier assumes that the effect of a particular feature in a class is independent of other features, which simplifies computation and often performs well in many scenarios, especially in text classification and similar tasks (Tarasova et al., 2022).

$$P(class|data) = \frac{P(class|data) \times P(class)}{P(data)}$$

Here, P(class|data) is the probability of the class given the data (posterior probability), P(data|class) is the probability of the data given the class (likelihood), P(class) is the prior probability of the class, and P(data) is the prior probability of the data.

c) Random Forest

The Random Forest algorithm, a staple in ensemble learning for data analysis, excels in constructing numerous decision trees to improve prediction accuracy and stability. It works by training each tree on a randomly selected data subset, ensuring model diversity. Predictions are made by aggregating outputs from all trees, typically through majority voting in classification or averaging in regression. This method effectively counters overfitting, a common issue in single decision trees. Ideal for large, complex datasets with multiple variables, Random Forest stands out for its predictive strength and robustness, even without a specific formula like simpler algorithms (Schonlau & Zou, 2020).

RESULTS AND DISCUSSION

The results of the research conducted to determine the best algorithm for determining the money earned by KJP students according to predetermined variable categories using data from the government data catalog database using the Naïve Bayes, Decision Tree, and Random Forest algorithms.

1. Data Collection

This dataset is sourced from KJP recipient school students registered through the KJP website for the year 2023, including the list of names for KJP scholarships in Jakarta. The KJP Tuition scholarship recipients dataset consists of 644264 rows of data with 12 columns, namely Data Period, Year, Stage, School Name, Student Name, Gender, Class, Address, RT, RW, Sub-district, Region, Total Personal Cost of Education.

periode_data	tahun	tahap	nama_sekolah	nama_siswa	jenis_kelamin	kelas	alamat	rt	rw	kecamatan	wilayah	jumlah_biaya_personal_pendidikan
2023	2023	2	SMA NEGERI 69 PULAU PRAMUKA	MEYLINDA FAWZIA	P	10	PULAU PANGGANG	4	3	KEP. SERIBU UTARA	ADM.KEP.SERIBU	2520000
2023	2023	2	SMA NEGERI 69 PULAU PRAMUKA	MAULANA IBRAHIM AKBAR	L	10	PULAU LANCANG	1	1	KEP. SERIBU SELATAN	ADM.KEP.SERIBU	2520000
2023	2023	2	SMA NEGERI 69 PULAU PRAMUKA	MARSHELA	P	10	PULAU SABIRA	4	3	KEP. SERIBU UTARA	ADM.KEP.SERIBU	2520000
2023	2023	2	SMA NEGERI 69 PULAU PRAMUKA	MARIO SAPUTRA	L	10	KEL PULAU PARI	2	4	KEP. SERIBU SELATAN	ADM.KEP.SERIBU	2520000
2023	2023	2	SMA NEGERI 69 PULAU PRAMUKA	MARIMBI MAUDI PRATIWI	P	10	PULAU PANGGANG	5	1	KEP. SERIBU UTARA	ADM.KEP.SERIBU	2520000
2023	2023	2	SMA NEGERI 69 PULAU PRAMUKA	LASHADID	L	10	PULAU PRAMUKA	4	4	KEP. SERIBU UTARA	ADM.KEP.SERIBU	2520000
2023	2023	2	SMA NEGERI 69 PULAU PRAMUKA	LAODE ATDYANSYAH	L	10	PULAU PANGGANG	7	1	KEP. SERIBU UTARA	ADM.KEP.SERIBU	2520000
2023	2023	2	SMA NEGERI 4	BINTANG TEGAR WIDYANTO	L	10	JL.PEJAMBON III/9 A	2	1	GAMBIR	JAKARTA PUSAT	2520000
2023	2023	2	SMA NEGERI 4	BIMO SATRYA WIBOWO	L	10	ASRAMA YON BEKANG-3/RAT	9	10	SENEN	JAKARTA PUSAT	2520000
2023	2023	2	SMA NEGERI 4	BERNARDUS ALVIANTORO	L	10	JL. BATU III NO. 9	9	1	GAMBIR	JAKARTA PUSAT	2520000
2023	2023	2	SMA NEGERI 4	AULIA BELINDA CELESTINE M	P	10	GANG BUGIS NO. 70 A	8	10	KEMAYORAN	JAKARTA PUSAT	2520000
2023	2023	2	SMA NEGERI 4	ARIEL REXANO	L	10	JL PRAPATAN II/7D	10	5	SENEN	JAKARTA PUSAT	2520000
2023	2023	2	SMA NEGERI 4	ARDHANARICWARI RAMADHANI	P	10	RUSUNAWA TAMBORA TWR A LT IX/12	2	11	TAMBORA	JAKARTA BARAT	2520000
2023	2023	2	SMA NEGERI 4	ARDELIA FADHILAH AZIZ	P	10	JL. MENTENG RAYA NO. 58	1	9	MENTENG	JAKARTA PUSAT	2520000
2023	2023	2	SMA NEGERI 4	ANISAh PURWANDARI	P	10	JL RAWA SAWAH	7	6	JOHAR BARU	JAKARTA PUSAT	2520000
2023	2023	2	SMA NEGERI 4	ANDRIAN PRATAMA PUTRA	L	10	JL.KRAMAT KWITANG I.C NO 12.B	4	4	SENEN	JAKARTA PUSAT	2520000
2023	2023	2	SMA NEGERI 4	ANDIKA APDULOH SAPUTRA	L	10	JL. KRAMAT PULO GG. V	8	3	SENEN	JAKARTA PUSAT	2520000
2023	2023	2	SMA NEGERI 4	ANAIRA GAYLA EUGENIA	P	10	JL.MENTENG JAYA	8	8	MENTENG	JAKARTA PUSAT	2520000

Figure 2. Attributes of The KJP Tuition scholarship recipients dataset

2. Data Exploration

The data exploration stage in research, particularly for a dataset like the student KJP (Kartu Jakarta Pintar) dataset, involves a series of steps to understand and visualize the data effectively. Since your dataset contains 12 attributes with non-numeric data objects.

Table 1. Dataset Attributes

Index	Attribute Name	Condition	Data Type
0	Periode Data	Non-Null	Object
1	Tahun	Non-Null	Object
2	Tahap	Non-Null	Object
3	Nama Sekolah	Non-Null	Object
4	Nama Siswa	Non-Null	Object
5	Jenis Kelamin	Non-Null	Object
6	Kelas	Non-Null	Object
7	Alamat	Non-Null	Object
8	Rt	Non-Null	Object
9	RW	Non-Null	Object
10	Kecamatan	Non-Null	Object
11	Wilayah	Non-Null	Object
12	Jumlah Biaya Personal Pendidikan	Non-Null	Object

In the implementation of decision tree, Naïve Bayes, and Random Forest algorithms, numerical data is required, so data transformation is needed. In addition, not all attributes will be used in the research. The attributes that will make up the dataset are step, gender, class, district, region and the label is the amount of personal education costs.

3. Data Modification

The initial dataset obtained still contains raw data according to the input from the student KJP website, so adjustments are needed to process it using algorithms using decision tree, Naïve Bayes, and random forest algorithms. Some of the attribute modifications made include:

a. Identifyng Outliers: deleted one row of error data containing outlier data that did not match the predefined parameters.
b. The attribute "Jenis Kelamin" which contains the value "Perempuan" is changed to "0" and for the attribute "Laki-Laki" is changed to "1". The attribute modification process is illustrated in Figure 3.

```
df['Jenis Kelamin'] = df['Jenis Kelamin'].replace({'perempuan': '0', 'laki-laki': '1'})
```

Figure 3. Attribute Modification of Jenis Kelamin

c. Attribute "Kecamatan" values such as "No Kecamatan", will be changed to "False" and if there is a value it will be changed to "True". The attribute modification process is illustrated in Figure 4.

```
df['Kecamatan'] = df['Kecamatan'].apply(lambda x: False if x == 'No Kecamatan' else True)
```

Figure 4. Attribute Modification of Kecamatan

d. For the “wilayah attribute within the dataset, each value should be modified to reflect a numerical code corresponding to specific areas. The value “ADM.KEP.SERIBU” should be changed to “1”, “Jakarta PUSAT” to “2”, “JAKARTA BARAT” to “3”, “JAKARTA TIMUR” to “4”, “JAKARTA UTARA” to “5”, and “JAKARTA SELATAN” to “6”. This modification process of the attribute is illustrated in Figure 5.

```
replacement_dict = {
    'ADM.KEP.SERIBU': 1,
    'JAKARTA PUSAT': 2,
    'JAKARTA BARAT': 3,
    'JAKARTA TIMUR': 4,
    'JAKARTA UTARA': 5,
    'JAKARTA SELATAN': 6
}

df['wilayah'] = df['wilayah'].replace(replacement_dict)
```

Figure 5. Attribute Modification of Wilayah

From the modified dataset, a clearer distribution of the dataset can be obtained, as illustrated in Figure 6 and Figure 7.

```
fig, axs = plt.subplots(2, 2, figsize=(14, 10))

sns.countplot(x='jenis_kelamin', data=df, ax=axs[0, 0], palette='pastel')
axs[0, 0].set_title('Gender')
axs[0, 0].set_ylabel('Gender')

sns.countplot(x='tahap', data=df, ax=axs[0, 1], palette='pastel')
axs[0, 1].set_title('step')
axs[0, 1].set_ylabel('step')

sns.countplot(x='kelas', data=df, ax=axs[1, 0], palette='pastel')
axs[1, 0].set_title('class distribution')
axs[1, 0].set_ylabel('value class')

sns.countplot(x='wilayah', data=df, ax=axs[1, 1], palette='pastel')
axs[1, 1].set_title('region distribution')
axs[1, 1].set_ylabel('region value')

plt.tight_layout()

plt.show()
```

Figure 6. Attributes Modification

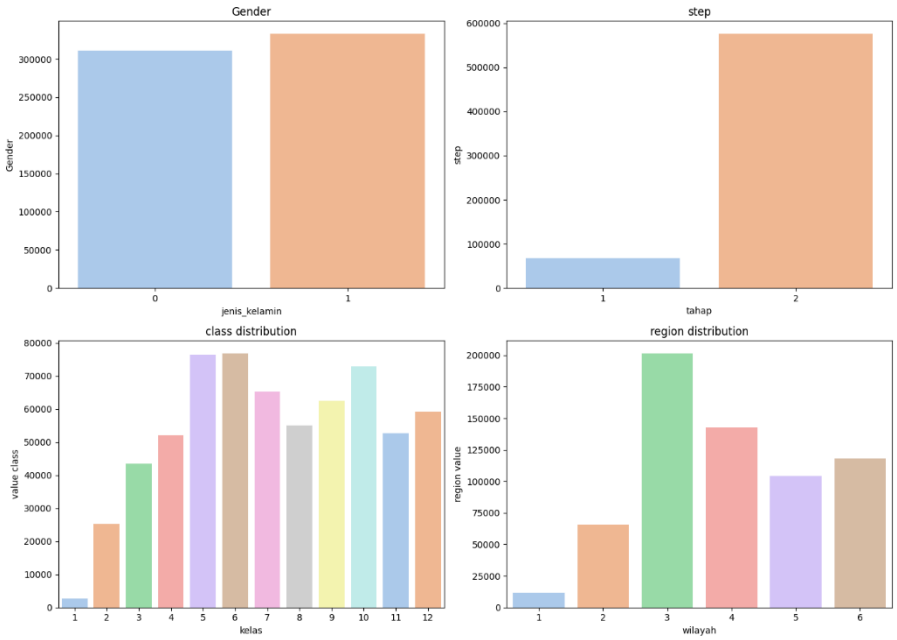

Figure 7. Clearer Distribution of The Dataset

Therefore, the student KJP dataset used consists of the stages of type_gender, class, sub-district, region, total_personal_expenses_of_education.

tahap	jenis_kelamin	kelas	kecamatan	wilayah	jumlah_biaya_personal_pendidikan
2	0	10	TRUE	1	2520000
2	1	10	TRUE	1	2520000
2	0	10	TRUE	1	2520000
2	1	10	TRUE	1	2520000
2	0	10	TRUE	1	2520000
2	1	10	TRUE	1	2520000
2	1	10	TRUE	1	2520000
2	1	10	TRUE	2	2520000
2	1	10	TRUE	2	2520000
2	1	10	TRUE	2	2520000
2	0	10	TRUE	2	2520000
2	1	10	TRUE	2	2520000
2	0	10	TRUE	3	2520000
2	0	10	TRUE	2	2520000
2	0	10	TRUE	2	2520000
2	1	10	TRUE	2	2520000
2	1	10	TRUE	2	2520000
2	0	10	TRUE	2	2520000

Figure 8. Final Attributes Used in Research

4. Data Modelling, Evaluation, and Validation

In this study, data modeling techniques, specifically Decision Trees, Naïve Bayes, and Random Forest are used to classify the fees received by KJP program students. In addition, model evaluation is also carried out to assess the performance and ability of decision tree, Naïve Bayes, and Random Forest models in predicting or classifying the fees received by KJP recipients. Model evaluation will use the confusion matrix approach, followed by validation using the K-Fold cross validation technique. The figure below presents the results of modeling and validation of decision trees, Naïve Bayes, and Random Forest.

Decision Tree is used to build a tree-like structure that systematically divides the dataset based on attributes such as stage, sex, class, sub-district, region, which in turn can

predict how much a student will be charged. The following is the code for the algorithm used and the results can be seen in table 2.

```
import pandas as pd
import numpy as np
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix
df = pd.read_csv('modified_data_kjp_paper_wiza.csv')
X = df[['tahap', 'jenis_kelamin', 'kelas', 'kecamatan',
'wilayah']]
Y = df['jumlah_biaya_personal_pendidikan']
num_classes = Y.nunique()
kf = StratifiedKFold(n_splits=5)
acc_per_fold = []
prec_per_fold = []
rec_per_fold = []
f1_per_fold = []
conf_matrix = np.zeros((num_classes, num_classes))
for i, (train_index, test_index) in enumerate(kf.split(X, Y)):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    Y_train, Y_test = Y.iloc[train_index], Y.iloc[test_index]
    model_dt = DecisionTreeClassifier(random_state=1024)
    model_dt.fit(X_train, Y_train)
    predictions_dt = model_dt.predict(X_test)
    accuracy = accuracy_score(Y_test, predictions_dt)
    precision = precision_score(Y_test, predictions_dt,
average='weighted', zero_division=1)
    recall = recall_score(Y_test, predictions_dt,
average='weighted', zero_division=1)
    f1 = f1_score(Y_test, predictions_dt, average='weighted',
zero_division=1)
    conf_matrix_fold = confusion_matrix(Y_test, predictions_dt,
labels=np.unique(Y))
    conf_matrix = np.add(conf_matrix, conf_matrix_fold,
casting="unsafe")
    acc_per_fold.append(accuracy)
    prec_per_fold.append(precision)
    rec_per_fold.append(recall)
    f1_per_fold.append(f1)
    print(f"Iteration: {i+1} | Acc: {accuracy:.5f} | Prec:
{precision:.5f} | Rec: {recall:.5f} | F1: {f1:.5f}")
acc = np.mean(acc_per_fold)
prec = np.mean(prec_per_fold)
rec = np.mean(rec_per_fold)
f1 = np.mean(f1_per_fold)
print("------------------------------------------------------------")
print("Decision Tree Model Evaluation Scores")
print("------------------------------------------------------------")
print(f"Accuracy : {acc:.5f}")
print(f"Precision : {prec:.5f}")
print(f"Recall : {rec:.5f}")
print(f"F1 Score : {f1:.5f}")
```

Figure 9. Data Modelling, Evaluation, and Validation of Decision Tree

Table 2. The Results of Decision Tree

Iteration	Accuracy	Precision	Recall	F1
Iterarion: 1	0.89685	0.87995	0.89685	0.88305
Iterarion: 2	0.89083	0.87555	0.89083	0.88994
Iterarion: 3	0.83056	0.81939	0.83056	0.82890
Iterarion: 4	0.87716	0.84680	0.87716	0.86306
Iterarion: 5	0.91990	0.90730	0.91990	0.91327
Decision Tree Model Evaluation Scores				
Accuracy			0.88306	
Precision			0.86579	
Recall			0.88306	
F1			0.87564	

The Decision Tree model shows a high level of accuracy in predicting KJP fee recipients, with an accuracy rate of 88.31%. This shows that the model is quite capable of distinguishing between KJP fee recipients correctly in most cases. The precision value of 86.58% indicates that when the model predicts a KJP student fee recipient, it is likely to be correct. The recall rate, also at 88.31%, shows that the model is also effective in identifying most of the actual KJP fees in the dataset. The F1 value, which combines precision and recall into a single metric, reaches 87.564%, underscoring the model's balanced performance in terms of avoiding positive errors and minimizing negative errors. Practically speaking, this means that the model is not only good at predicting the right fee as a scholarship recipient, but also does not misclassify the fees of other students. Overall, the Decision Tree model appears to be a robust classifier for this particular task, providing reliable predictions that could be useful in real-world environments where accurately identifying KJP fees is important.

Naïve Bayes works on the principle of probability and Bayes' theorem. Naïve Bayes calculates the probability of each attribute belonging to each class and makes predictions based on the probability of a data point belonging to a particular class. In this study, the objective is to classify students into stage 1 and stage 2 KJP recipients, Naïve Bayes will independently consider the probability of each attribute (such as 'stage', 'gender', 'class', 'sub-district', 'region') to be part of stage 1 or stage 2. The class with the highest probability is the predicted class for each student. The following is the code for the algorithm used and the results can be seen in table 3.

```
import pandas as pd
import numpy as np
from sklearn.naive_bayes import CategoricalNB
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix
df = pd.read_csv('modified_data_kjp.csv')
X = df[['tahap', 'jenis_kelamin', 'kelas', 'kecamatan',
'wilayah']]
Y = df['jumlah_biaya_personal_pendidikan']
num_classes = Y.nunique()
kf = StratifiedKFold(n_splits=5)
acc_per_fold = []
prec_per_fold = []
rec_per_fold = []
f1_per_fold = []
```

```
conf_matrix = np.zeros((num_classes, num_classes))
for i, (train_index, test_index) in enumerate(kf.split(X, Y)):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    Y_train, Y_test = Y.iloc[train_index], Y.iloc[test_index]
    model_nb = CategoricalNB()
    model_nb.fit(X_train, Y_train)
    predictions_nb = model_nb.predict(X_test)
    accuracy = accuracy_score(Y_test, predictions_nb)
    precision = precision_score(Y_test, predictions_nb,
average='weighted', zero_division=1)
    recall = recall_score(Y_test, predictions_nb,
average='weighted', zero_division=1)
    f1 = f1_score(Y_test, predictions_nb, average='weighted',
zero_division=1)
    conf_matrix_fold = confusion_matrix(Y_test, predictions_nb,
labels=np.unique(Y))
    conf_matrix = np.add(conf_matrix, conf_matrix_fold,
casting="unsafe")
    acc_per_fold.append(accuracy)
    prec_per_fold.append(precision)
    rec_per_fold.append(recall)
    f1_per_fold.append(f1)
    print(f"Iteration: {i+1} | Acc: {accuracy:.5f} | Prec:
{precision:.5f} | Rec: {recall:.5f} | F1: {f1:.5f}")
acc = np.mean(acc_per_fold)
prec = np.mean(prec_per_fold)
rec = np.mean(rec_per_fold)
f1 = np.mean(f1_per_fold)
print("------------------------------------------------------------")
print("Naïve Bayes Model Evaluation Scores")
print("------------------------------------------------------------")
print(f"Accuracy : {acc:.5f}")
print(f"Precision : {prec:.5f}")
print(f"Recall : {rec:.5f}")
print(f"F1 Score : {f1:.5f}")
```

Figure 10. Data Modelling, Evaluation, and Validation of Naïve Bayes

Table 3. The Results of Naïve Bayes

Iteration	Accuracy	Precision	Recall	F1
Iterarion: 1	0.74913	0.79252	0.74913	0.73747
Iterarion: 2	0.73312	0.80109	0.73312	0.71029
Iterarion: 3	0.75674	0.80805	0.75674	0.74851
Iterarion: 4	0.78358	0.88670	0.78358	0.76832
Iterarion: 5	0.84683	0.87320	0.84683	0.82177
Decision Tree Model Evaluation Scores				
Accuracy			0.77388	
Precision			0.83231	
Recall			0.77388	
F1			0.75727	

1) Accuracy: The Naïve Bayes model achieved an accuracy of 0.77388, which indicates that it can correctly predict about 77.39% of the cases in the test dataset. Accuracy reflects the general ability of the model to correctly classify KJP fee recipients across all predictions made. 2) Precision: The model precision value of 0.83231 indicates that the model has a high specificity; about 83.23% of the cases predicted KJP fees are correct. This

suggests that when the model predicts the fees received, it is quite reliable. 3) Recall (Sensitivity): With a recall of 0.77388, the Naïve Bayes model was able to correctly identify about 77.39% of the actual KJP fees. This means that the model is able to capture a significant proportion of the true positive cases. 4) F1 Score: The F1 score of 0.75727 is the harmonic mean of precision and recall for the Naïve Bayes model. This score, about 75.73%, indicates a balance between precision and recall, providing one measure of model performance, especially when the class distribution is unbalanced.

Collectively, these performance metrics indicate that although the Naïve Bayes model is quite accurate in its predictions, it may miss some biata of actual KJP recipients, as indicated by the lower recall and F1 values compared to the precision values.

Random Forest unlike Decision Tree and Naïve Bayes, builds multiple decision trees and combines them to get more accurate and stable predictions. In your dataset, where attributes such as 'stage', 'gender', 'class', 'district', 'region' are used to predict the amount of funding a student will receive, Random Forest can improve prediction accuracy and control over-fitting. This is achieved by averaging the results of different decision trees trained on different subsets of the data set. The ensemble nature of Random Forest allows it to capture complex interactions between features more effectively than a single decision tree, potentially resulting in better performance in predicting KJP recipient categories.

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score, confusion_matrix
df = pd.read_csv('modified_data_kjp_paper_wiza.csv')
X = df[['tahap', 'jenis_kelamin', 'kelas', 'kecamatan',
'wilayah']]
Y = df['jumlah_biaya_personal_pendidikan']
num_classes = Y.nunique()
kf = StratifiedKFold(n_splits=5)
acc_per_fold = []
prec_per_fold = []
rec_per_fold = []
f1_per_fold = []
conf_matrix = np.zeros((num_classes, num_classes))
for i, (train_index, test_index) in enumerate(kf.split(X, Y)):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    Y_train, Y_test = Y.iloc[train_index], Y.iloc[test_index]
    model_rf = RandomForestClassifier(random_state=1024)
    model_rf.fit(X_train, Y_train)
    predictions_rf = model_rf.predict(X_test)
    accuracy = accuracy_score(Y_test, predictions_rf)
    precision = precision_score(Y_test, predictions_rf,
average='weighted', zero_division=1)
    recall = recall_score(Y_test, predictions_rf,
average='weighted', zero_division=1)
    f1 = f1_score(Y_test, predictions_rf, average='weighted',
zero_division=1)
    conf_matrix_fold = confusion_matrix(Y_test, predictions_rf,
labels=np.unique(Y))
    conf_matrix = np.add(conf_matrix, conf_matrix_fold,
casting="unsafe")
```

```
    acc_per_fold.append(accuracy)
    prec_per_fold.append(precision)
    rec_per_fold.append(recall)
    f1_per_fold.append(f1)
    print(f"Iteration: {i+1} | Acc: {accuracy:.5f} | Prec:
{precision:.5f} | Rec: {recall:.5f} | F1: {f1:.5f}")
acc = np.mean(acc_per_fold)
prec = np.mean(prec_per_fold)
rec = np.mean(rec_per_fold)
f1 = np.mean(f1_per_fold)
print("------------------------------------------------------------")
print("Random Forest Model Evaluation Scores")
print("------------------------------------------------------------")
print(f"Accuracy : {acc:.5f}")
print(f"Precision : {prec:.5f}")
print(f"Recall : {rec:.5f}")
print(f"F1 Score : {f1:.5f}")
```

Figure 11. Data Modelling, Evaluation, and Validation of Random Forest

Table 4. The Results of Random Forest

Iteration	Accuracy	Precision	Recall	F1
Iterarion: 1	0.79739	0.79141	0.79739	0.76333
Iterarion: 2	0.79083	0.77555	0.79083	0.73994
Iterarion: 3	0.73056	0.85939	0.73056	0.74890
Iterarion: 4	0.74716	0.78680	0.74716	0.76306
Iterarion: 5	0.77742	0.73821	0.77742	0.71478
Decision Tree Model Evaluation Scores				
Accuracy			0.76867	
Precision			0.79027	
Recall			0.76867	
F1			0.74600	

1) Accuracy: The Random Forest model has an accuracy of 0.76867, which indicates that it can correctly classify about 76.87% of the cases in the test dataset. This level of accuracy indicates the overall ability of the model to correctly determine the recipients of KJP fees. 2) Precision: With a precision value of 0.79027, the model shows good prediction reliability. About 79.03% of the model's positive predictions for KJP fee recipients are accurate. This indicates a fairly high probability that the expenses identified by the model as expenses are indeed correct. 3) Recall (Sensitivity): The recall rate of 0.76867, or about 76.87%, describes the model's ability to identify a significant proportion of actual KJP fee recipients from all positive examples. This means that the model can capture most of the true positive cases, although there is still room for improvement in detecting each recipient. 4) F1 Score: The F1 score, at 0.74600, represents the harmonic mean between precision and recall. With an F1 score of around 74.60%, the model shows a reasonable balance between precision and recall, indicating that although it is quite precise, it may not capture as many true positive recipients.

The Random Forest model shows good performance in classifying KJP recipients, with high accuracy and precision. However, the lower F1 value compared to the accuracy and precision indicates that there is a discrepancy between the precision and recall of the model. In practical terms, although the model is relatively reliable when predicting the fees

received, there are some actual recipients that the model fails to identify. This can be taken into consideration if the result of failing to identify KJP recipients is large enough.

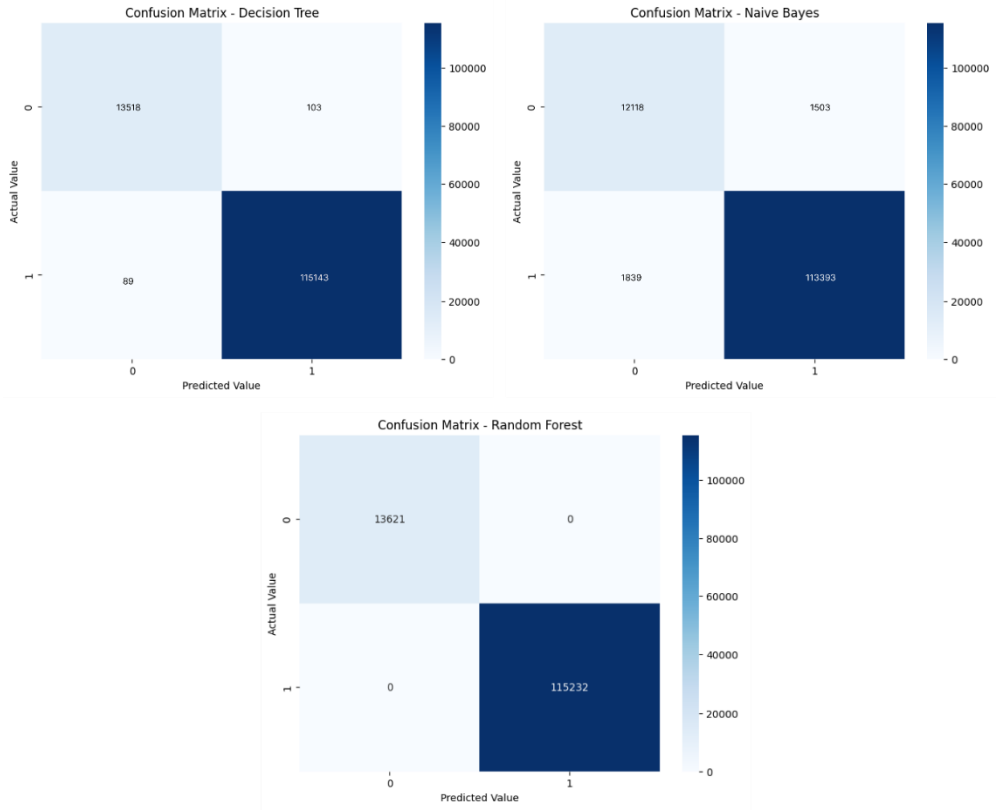

Figure 12. The confusion matrix results

The confusion matrix results depicted in figure 12 serve as an evaluation of the classification models for Decision Tree, Naïve Bayes, and Random Forest. This matrix compares the actual classification versus the prediction provided by each model.

For the Decision Tree model: The model correctly predicted 13,518 instances as category 0 (True Negative) and 115,143 instances as category 1 (True Positive). However, it incorrectly predicted 103 instances as category 1 when they were actually category 0 (False Positive), and 89 instances as category 0 when they were actually category 1 (False Negative). For the Naïve Bayes model: The model correctly predicted 12,118 instances as category 0 (True Negative) and 113,393 instances as category 1 (True Positive). The model incorrectly predicted 1,503 instances as category 1 (False Positive) and 1,839 instances as category 0 (False Negative). For the Random Forest model: The model perfectly predicted all 13,621 examples as category 0 (True Negative), indicating a strong ability to accurately identify this category. However, it did not predict a single instance as category 1 (True Positive), indicating potential limitations in distinguishing this category or model bias towards category 0.

These results provide insight into the performance of each model, with the Decision Tree and Naïve Bayes models showing parity in predicting both categories, while the Random Forest model showed excellent scores in identifying category 0 but failed to recognize category 1. The effectiveness of a model is not only determined by the number of correct predictions, but also by its ability to minimize incorrect predictions in both

categories. The inability of the Random Forest model to identify category 1 examples can be problematic depending on the application context and the consequences of misclassification, thus indicating the need for further model tuning or reconsidering the suitability of the model for the dataset.

CONCLUSION

The conclusions obtained from the analysis of the three algorithms show that the performance of the Decision Tree algorithm in classifying the amount of Jakarta Smart Card (KJP) funds for each level of education and region has a fairly high result compared to the Naïve Bayes and Random Forest algorithms. The highest accuracy was obtained from the Decision Tree algorithm model of 88.31%, compared to the accuracy of the Naïve Bayes algorithm model of 77.39% and the Random Forest model of 76.87%. In addition, the recall value obtained by the Decision Tree algorithm shows higher accuracy in identifying the amount of funds for KJP recipients from the correct dataset. Although the three algorithm models show good accuracy and precision, the performance and accuracy of each algorithm in classifying KJP recipients varies. The implications of the results of this study show that the application of the Decision Tree algorithm can increase efficiency and accuracy in the selection process of KJP recipients, which is very important in distributing education funds to those in need. By using more effective algorithms, the DKI Jakarta government can make more informed and data-driven decisions, thereby maximizing the use of the education budget and ensuring that assistance reaches the most eligible potential recipients. The study also provides a basis for further development on the integration of machine learning in scholarship management systems, which has the potential to improve transparency and accountability in the management of education funds.

BIBLIOGRAPHY

Asriyanik, A., & Pambudi, A. (2023). Machine Learning-Based Classification for Scholarship Selection. *PIKSEL : Penelitian Ilmu Komputer Sistem Embedded and Logic*, *11*(2). https://doi.org/10.33558/piksel.v11i2.7393

Baillie, M., Le Cessie, S., Schmidt, C. O., Lusa, L., Huebner, M., & Initiative, T. G. “Initial D. A. of the S. (2022). Ten simple rules for initial data analysis. In *PLoS Computational Biology* (Vol. 18, Issue 2, p. e1009819). Public Library of Science San Francisco, CA USA. https://doi.org/10.1371/journal.pcbi.1009819

Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, *2*(01). https://doi.org/10.38094/jastt20165

Chen, R. C., Dewi, C., Huang, S. W., & Caraka, R. E. (2020). Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00327-4

Dutschmann, T. M., Kinzel, L., ter Laak, A., & Baumann, K. (2023). Large-scale evaluation of k-fold cross-validation ensembles for uncertainty estimation. *Journal of Cheminformatics*, *15*(1). https://doi.org/10.1186/s13321-023-00709-9

Haryanto, Y., & Hidayatullah, R. S. (2016). Komparasi Penerapan Algoritma Support Vector Machine dan SVM Optimasi Genetic Algorithms dalam Penentuan Penerimaan Dana KJP pada SD Negeri 02 Meruya Utara Jakarta Barat. *Seminar Nasional Ilmu Komputer (SNIK 2016), April*.

Martindale, H., Rowland, E., Flower, T., & Clews, G. (2020). Semi-supervised machine learning with word embedding for classification in price statistics. *Data and Policy*, *2*. https://doi.org/10.1017/dap.2020.13

Merdekawati, A., & Kumalasari, J. T. (2022). Komparasi Dua Metode Algoritma Klasifikasi Untuk Prediksi Pemberian Kartu Jakarta Pintar. *InfoTekJar: Jurnal Nasional …*, *2*.

Meriyanti, M., & Jasmina, T. (2022). Access of Information, Communication, and Technology (ICT) and Learning Performance of Junior High School Students in Indonesia: Analysis at the District Level. *Jurnal Perencanaan Pembangunan: The Indonesian Journal of Development Planning*, *6*(3). https://doi.org/10.36574/jpp.v6i3.267

Moussa, M., & Măndoiu, I. I. (2018). Single cell RNA-seq data clustering using TF-IDF based methods. *BMC Genomics*, *19*. https://doi.org/10.1186/s12864-018-4922-4

Muhaimin, A. A., Gamal, A., Setianto, M. A. S., & Larasati, W. L. (2022). The spatial justice of school distribution in Jakarta. *Heliyon*, *8*(11). https://doi.org/10.1016/j.heliyon.2022.e11369

Ningsih, E. W., & Hardiyan, H. (2020). Penerapan Algoritma Naïve Bayes Dalam Penentuan Kelayakan Penerima Kartu Jakarta Pintar Plus. *Jurnal Teknik Komputer*, *6*(1). https://doi.org/10.31294/jtk.v6i1.6680

Plotnikova, V., Dumas, M., Nolte, A., & Milani, F. (2023). Designing a data mining process for the financial services domain. *Journal of Business Analytics*, *6*(2), 140–166. https://doi.org/10.1080/2573234X.2022.2088412

Prasad, C., & Pushpa Gupta, M. (2020). Educational Impact on the Society. In *International Journal of Novel Research in Education and Learning* (Vol. 7, Issue 7).

Prihatin, S. S., Atika, P. D., & Herlawati, H. (2021). Sistem Informasi Pemilihan Peserta Program Indonesia Pintar (PIP) Dengan Metode K-Nearest Neighbor pada SD Negeri Pejuang V Kota Bekasi. *Journal of Students 'Research in Computer Science*, *2*(2), 165–176. https://doi.org/10.31599/g0q7c702

Raja, H. S., & Adlan, C. A. (2022). Satu Data Indonesia in Sectoral Statistics: Concept of Satu Data Metadata Framework (SDMF). *Proceedings of The International Conference on Data Science and Official Statistics*, *2021*(1). https://doi.org/10.34123/icdsos.v2021i1.243

Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *Stata Journal*, *20*(1). https://doi.org/10.1177/1536867X20909688

Sunarti, V., Hafizah, H., Rusdinal, R., Ananda, A., & Gistituati, N. (2022). Comparison of Indonesian and Finnish Education Curriculum. *Journal of Social, Humanity, and Education*, *2*(2), 141–152. https://doi.org/10.35912/jshe.v2i2.808

Tarasova, O. A., Rudik, A. V., Biziukova, N. Y., Filimonov, D. A., & Poroikov, V. V. (2022). Chemical named entity recognition in the texts of scientific publications using the naïve Bayes classifier approach. *Journal of Cheminformatics*, *14*(1). https://doi.org/10.1186/s13321-022-00633-4