

K-Nearest Neighbors Approach to Laptop Type Classification

David Mario Yohanes Samosir¹, I Gede Deindra Dwija Puridiasta², Dewa Komang Reiki Perdana Wisnu³

^{1,2,3}Universitas Pendidikan Ganesha; Jalan Udayana, telp.(0362)22570, Singaraja-Bali
e-mail: ¹david.mario@undiksha.ac.id, ²deindra@undiksha.ac.id, ³reiki@undiksha.ac.id

Abstract

This research was chosen based on the realization of the needs of many individuals who are looking for laptops, but are limited to the knowledge of technical specifications without knowing the appropriate laptop brand or model. In an effort to address this problem, we developed a laptop classification based on specifications using data from Kaggle. The methods used include data preprocessing to improve accuracy, and selection of a suitable classification model. The results show that this approach provides accurate predictions in identifying laptops based on their specifications. This classification allows users with limited technical knowledge to make better choices in selecting laptops according to their needs. This research not only provides a practical solution for consumers, but also increases understanding of the importance of technical specification knowledge in choosing a device. The conclusion of this research emphasizes the benefits of laptop classification that can increase convenience for consumers in choosing a device that suits their needs.

Keywords : Laptop, Classification, KNN, Accuracy

Abstrak

Penelitian ini dipilih berdasarkan kesadaran akan kebutuhan banyak individu yang mencari laptop, namun terbatas pada pengetahuan spesifikasi teknis tanpa mengetahui merek atau model laptop yang sesuai. Dalam usaha mengatasi masalah ini, kami mengembangkan klasifikasi laptop berdasarkan spesifikasi menggunakan data dari Kaggle. Metode yang digunakan mencakup preprocessing data untuk meningkatkan akurasi, dan pemilihan model klasifikasi yang sesuai. Hasil penelitian menunjukkan bahwa pendekatan ini memberikan prediksi yang akurat dalam mengidentifikasi laptop berdasarkan spesifikasinya. Klasifikasi ini memungkinkan pengguna dengan pengetahuan teknis terbatas untuk membuat pilihan yang lebih baik dalam memilih laptop sesuai kebutuhan mereka. Penelitian ini tidak hanya memberikan solusi praktis untuk konsumen, tetapi juga meningkatkan pemahaman tentang pentingnya pengetahuan spesifikasi teknis dalam memilih perangkat. Kesimpulan penelitian ini menekankan pada manfaat klasifikasi laptop yang dapat meningkatkan kemudahan bagi konsumen dalam memilih perangkat yang sesuai dengan kebutuhan mereka.

Kata kunci : Laptop, Klasifikasi, KNN, Akurasi

1. INTRODUCTION

Machine learning (ML) is a branch of computer science that studies how systems can learn and improve their performance without being explicitly programmed. ML has a wide range of applications, including classification, regression, clustering, and natural language processing.

Classification is one of the most common types of ML tasks. Classification involves separating data into different classes. For example, classification can be used to predict the gender of a person based on their photo, or to determine whether an email is spam or not. Classification algorithms are methods used to accomplish classification tasks. There are a wide variety of classification algorithms available, including decision tree algorithms, support vector machine

(SVM) algorithms, and K-Nearest Neighbors (KNN) algorithms. The KNN algorithm is a simple yet effective classification algorithm. The KNN algorithm works by finding the k closest data to the new data to be classified. This closest data is then used to determine the class of the new data.

Laptops are common devices used for a variety of purposes, from work and entertainment. However, finding the right laptop for you could be difficult, especially if you are not familiar with all the different types of laptops that are available. One of the ways to choose the right laptop is to figure out the type of laptop. Laptop types can be differentiated based on many factors, such as the screen size, processor, RAM, storage, and price.

In this paper, we propose a K-Nearest Neighbors (KNN) approach to predict laptop types based on their specifications. The datasets used in this study consist of laptop features from Kaggle. KNN algorithm is used for predicting the types of laptops based on their specifications.

2. METHODS

2.1 Problem analysis

The problem addressed in this research arises from the common challenge faced by individuals seeking laptops who lack knowledge about appropriate laptop brands or models. Many potential laptop buyers are limited to understanding technical specifications without being able to make informed decisions about the most suitable laptop for their needs. To tackle this issue, the research employs a machine learning approach, specifically the K-Nearest Neighbors (KNN) algorithm, to classify laptops based on their specifications using data from Kaggle.

The primary problem revolves around the difficulty consumers face in selecting the right laptop due to a lack of familiarity with diverse laptop types available in the market. Technical specifications, such as screen size, processor, RAM, storage, and price, play a crucial role in determining the suitability of a laptop for specific purposes. The research aims to bridge this knowledge gap by creating a classification system that accurately predicts laptop types based on their specifications.

2.2 Completion Stages

In the process of classifying laptop components, there are several steps to produce a good model from precise data that has been adjusted for the laptop name type prediction program. There are:



Picture 1 Completion Steps

2.2.1 Literature Study

2.2.1.1 Kaggle

Kaggle is a data science competition platform and online community of data scientists and machine learning practitioners. Kaggle was founded in 2010 and acquired by Google in 2017. Since then, Kaggle has become the world's largest data science community with over 15 million registered users as of October 2023 [1]. The platform allows users to discover and publish datasets, explore and build models in a web-based data science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges. Kaggle also provides resources for learning, such as courses and pre-trained models, making it a comprehensive platform for data science and machine learning enthusiasts.

2.2.1.2 Google Colab

Google Colab, short for Google Colaboratory, is a cloud-based platform provided by Google that allows users to write and execute Python code in a collaborative environment [2]. It is a hosted Jupyter notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs. Colab is especially well suited to machine learning, data science, and education. It allows users to combine executable code and rich text in a single document, along with images, HTML, LaTeX, and more. Additionally, it supports most popular machine learning libraries, and its notebooks are stored in Google Drive and can be easily shared with others for real-time collaboration.

2.2.1.3 Machine Learning

Machine learning (ML) is a subfield of artificial intelligence (AI) that focuses on the development and study of statistical algorithms that can effectively generalize and perform tasks without explicit instructions [3]. It enables computers to learn from data and make decisions or predictions without being explicitly programmed to perform specific tasks. Machine learning techniques have been applied to various fields, including computer vision, speech recognition, email filtering, agriculture, and medicine. In general, the way Machine Learning works is by processing a series of data called data sets, which originate from a system by determining system values, determining which attributes and which are the responses, then making a model based on those values, so that when there is data new, the expected value will be in accordance with the expectations of the model obtained [4].

There are several parts to Machine Learning, such as:

- a) Supervised learning: Supervised learning builds a knowledge base from the preclassified patterns that supports to classify new pattern [5]. Supervised learning techniques have achieved great success when there is strong supervision information like a large amount of training examples with ground-truth labels. In real tasks, however, collecting supervision information requires costs, and thus, it is usually desirable to be able to do weakly supervised learning [6].
- b) Unsupervised learning: This type of learning involves clustering data points, reducing dimensionality, and discovering patterns in the data without the use of labeled datasets.
- c) Deep learning: Deep learning is a set of algorithms of machine learning which uses multiple layers that corresponds to different level of abstraction to each level. . It consists of input layer, output layer and several hidden layer. It is used for voice synthesis, image processing, handwriting recognition, object detection, prediction analytics and decision making [7].
- d) Predictive modeling: Predictive modeling is based on one or more data instances for which we want to predict the value of a target variable. Data-driven predictive modeling generally induces a model from training data, for which the value of the target (the label) is known [8].

2.2.1.4 Python

Python is a high-level, general-purpose programming language that is interpreted, object-oriented, and dynamically typed. It emphasizes code readability with the use of significant indentation and supports multiple programming paradigms. Python is designed to be easily readable and its simple, easy-to-learn syntax reduces the cost of program maintenance. It is used in a range of applications, including data science, software and web development, automation, and system scripting. Python's popularity has grown in recent years due to its versatility, beginner-friendliness, and large and active community that contributes to its pool of modules and libraries [9].

2.2.1.5 KNN Algorithm

The k-nearest neighbors (KNN) algorithm is a non-parametric supervised learning method used for classification and regression [10]. In KNN, the output is determined by the majority vote of its neighbors for classification, and by the average of the values of the k nearest neighbors for regression. It is based on the principle that similar points are located near each other in a feature space. The "k" in KNN refers to the number of nearest neighbors to include in the majority vote or averaging process. It is a simple algorithm that stores all available cases and classifies new data based on a similarity measure.

2.2.1.6 Laptop Specification

Laptop specifications refer to the internal components and features of a laptop computer, including processor, memory (RAM), storage, and graphics capabilities. These specifications determine the laptop's performance and ability to handle various tasks [11]. Key components and features to consider when looking at laptop specifications are:

- a) **Processor:** The central processing unit (CPU) is responsible for processing information. Common processor manufacturers include Intel and AMD, with models such as i3, i5, i7, and i9. A higher processor number indicates a more powerful processor.
- b) **Memory (RAM):** RAM is used for multitasking and running multiple applications simultaneously. More RAM can provide a speed boost, with 8GB being the minimum to aim for, 16GB or 32GB being recommended for high-end machines.
- c) **Storage:** Laptops typically have either a hard disk drive (HDD) or solid-state drive (SSD) for storing files, programs, and data. HDDs are more affordable but slower, while SSDs are faster but more expensive.
- d) **Graphics Card:** An additional graphics card is used for tasks like image editing and gaming. Some laptops have dedicated graphics cards, while others rely on integrated graphics.
- e) **Display:** The screen size and resolution are important factors to consider, as they can affect the overall user experience and productivity.
- f) **Operating System:** The operating system, such as Windows or macOS, is the software that runs on the laptop and determines its functionality and compatibility with various programs.

2.2.2 Obtaining Data

Data about laptop prices has been successfully obtained through the Kaggle platform using a dataset titled "Laptop Price". After reading the dataset, it was found that the dataset consists of 13 columns, namely Company, TypeName, RAM, Weight, Price, TouchScreen, IPS, Ppi, Cpu_brand, HDD, SSD, Gpu_brand, and Os. Each of these columns is further divided into categories to make the analysis easier, such as Company with categories (Acer, Apple, Asus, Chuwi, Dell, Fujitsu, Google, HP, Huawei, Lenovo, LG, Mediacom, Microsoft, MSI, Razer, Samsung, Toshiba, Vero, Xiaomi), TypeName with categories (2 in 1 Convertible, Gaming, Netbook, Notebook, Ultrabook, Workstation), Cpu_brand with categories (AMD Processor, Intel Core i3, Intel Core i5, Intel Core i7, Other Intel Processor), Gpu_brand with categories (AMD, Intel, Nvidia), and Os with categories (Mac, Others, Windows). The dataset totals 1273 entries, providing a rich and comprehensive framework for

conducting in-depth analysis on various aspects related to laptops, such as hardware specifications, brand, and operating system.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Company	TypeName	Ram	Weight	Price	TouchScreen	Ips	Ppi	Cpu_brand	HDD	SSD	Gpu_brand	Os
2	Apple	Ultrabook	8	1.37	11.17575455	0	1	226.9830047	Intel Core i5	0	128	Intel	Mac
3	Apple	Ultrabook	8	1.34	10.77677732	0	0	127.6779401	Intel Core i5	0	0	Intel	Mac
4	HP	Notebook	8	1.86	10.32993107	0	0	141.2119981	Intel Core i5	0	256	Intel	Others
5	Apple	Ultrabook	16	1.83	11.81447594	0	1	220.5346239	Intel Core i7	0	512	AMD	Mac
6	Apple	Ultrabook	8	1.37	11.47310097	0	1	226.9830047	Intel Core i5	0	256	Intel	Mac
7	Acer	Notebook	4	2.1	9.967025573	0	0	100.4546699	AMD Processor	500	0	AMD	Windows
8	Apple	Ultrabook	16	2.04	11.64410812	0	1	220.5346239	Intel Core i7	0	0	Intel	Mac
9	Apple	Ultrabook	8	1.34	11.03061499	0	0	127.6779401	Intel Core i5	0	0	Intel	Mac
10	Asus	Ultrabook	16	1.3	11.28544251	0	0	157.3505121	Intel Core i7	0	512	Nvidia	Windows

Picture 2 Dataset From Kaggle

2.2.3 Data Cleaning

In the data cleaning stage, preprocessing steps are performed for handling missing values, normalization, and data transformation or data format adjustment to suit the needs of the analysis, thus ensuring that the data used for model training is of high quality.

2.2.4 Data Visualization

Involves creating informative data visualizations to provide better insight into the characteristics of the dataset. Graphs and plots are created to understand variable distributions, correlations between features, and trends that may be useful in the selection of appropriate classification models.

2.2.5 Data Evaluate

Model evaluation is performed using confusion matrix to measure the performance of the model in laptop classification. Metrics such as accuracy, precision, recall, and F1-score are calculated to provide a comprehensive picture of the model's effectiveness in identifying laptop types based on their specifications. The results of this evaluation will be the basis for drawing conclusions and recommending further development steps.

3. RESULT AND DISCUSSION

3.1 Data Cleaning

The provided Python code utilizes the scikit-learn library to perform label encoding on categorical columns within a DataFrame. A `LabelEncoder` object is initialized and applied to encode specific columns such as 'Company,' 'Cpu_brand,' 'Gpu_brand,' 'TypeName,' and 'Os.' The encoded values are stored in new columns, and the code subsequently prints the original categorical values along with their corresponding numerical encodings for each column. This label encoding process is essential for preparing categorical data to be compatible with machine learning algorithms that require numerical input.

✓ Create Encoding Labels for Categorical Columns

```
# Create a new column for the encoded
col_encoder = LabelEncoder()
df['Company_label'] = col_encoder.fit_transform(df['Company']) #change into numerik data
df['Cpu'] = col_encoder.fit_transform(df['Cpu_brand']) #change into numerik data
df['Gpu'] = col_encoder.fit_transform(df['Gpu_brand']) #change into numerik data
df['TypeName_label'] = col_encoder.fit_transform(df['TypeName']) #change into numerik data
df['Os_label'] = col_encoder.fit_transform(df['Os']) #change into numerik data
```

Picture 3 Cleaning Data

3.2 Data Visualization

In the K-Nearest Neighbors approach to Laptop Type Classification, data visualization plays a crucial role to provide a deep understanding of the relationship between various categories and related variables, with TypeName as the prediction goal. Through this data visualization, it is possible to clearly observe how a large amount of data is distributed among the relevant categories, providing rich insights into the characteristics of each laptop type.



Picture 4 Quantity of Data per Category



Picture 5 Quantity of Category data to TypeName

```

# Check the relationship between the laptop type label and other variables in the DataFrame.
print(df.columns)
list_value = ["Company_label", "TypeName_label", "Ram", "Weight", "Price", "TouchScreen", "Ips", "Ppi", "Cpu", "HDD", "SSD", "Gpu", "Os_label"]

corr_matrix = df[list_value].corr()
print(corr_matrix["TypeName_label"].sort_values(ascending=False))

Index(['Company', 'TypeName', 'Ram', 'Weight', 'Price', 'TouchScreen', 'Ips',
      'Ppi', 'Cpu_brand', 'HDD', 'SSD', 'Gpu_brand', 'Os', 'Company_label',
      'Cpu', 'Gpu', 'TypeName_label', 'Os_label'],
      dtype='object')
TypeName_label    1.000000
Company_label     0.005373
Ppi              -0.028490
SSD              -0.073101
Os_label         -0.110535
Cpu              -0.121850
Price            -0.125212
Ips              -0.154569
HDD              -0.199795
Ram              -0.244020
Gpu              -0.252458
Weight           -0.278346
TouchScreen      -0.404706
Name: TypeName_label, dtype: float64

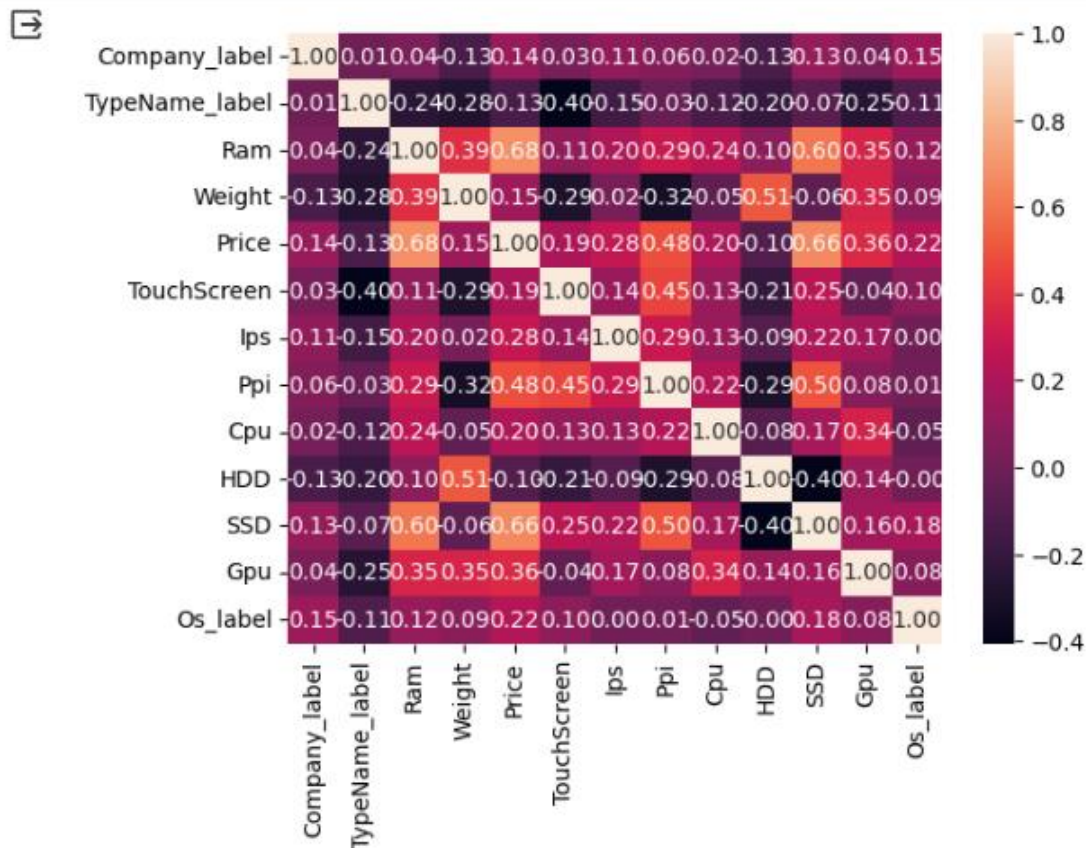
```

Picture 6 Category data relation to TypeName_label

```

sns.heatmap(df[list_value].corr(), annot=True, fmt=".2f")
plt.show()

```



Picture 7 Category Data Relationship

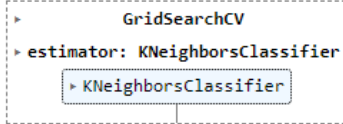
3.3 Data Modelling

In the data modeling stage, the first step is to adjust the hyperparameters to prevent overfitting or underfitting, optimize model performance, and select an appropriate model. This is important to ensure that the model built can provide accurate and effective predictions.

Once the hyperparameter tuning process is complete, the next step involves finding the confusion matrix. Confusion matrix is used to evaluate the extent to which the model can predict well for each class. By looking at the confusion matrix, it can be measured how efficient the model is in identifying and classifying instances into the correct class.

```
[ ] # Perform GridSearchCV for hyperparameter tuning (Find the best param)

param_grid = {'n_neighbors': [1, 3, 5, 7, 9, 11], 'weights': ['uniform', 'distance']}
knn = KNeighborsClassifier()
grid_search = GridSearchCV(knn, param_grid, cv=5)
grid_search.fit(x, y)
```



```
[ ] # Get the best hyperparameters
best_params = grid_search.best_params_
best_params

{'n_neighbors': 11, 'weights': 'uniform'}
```

Picture 8 Hyperparameter Tuning

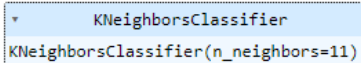
The next step is to split the data into two parts, data training and data testing. In this context, an 80:20 ratio is used, where 80% of the data is used as training data to train the model, while the remaining 20% is used as testing data to test how well the model can make predictions on data that has never been seen before. This process aims to objectively evaluate the performance of the model and ensure the reliability of the model when faced with new data.

Split The Data for Train and Test

```
[ ] # Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

```
[ ] # Create a new KNN model with the best parameters and train it
knn_model = KNeighborsClassifier(**best_params)
```

```
[ ] knn_model.fit(x_train, y_train)
```



Picture 9 Data Train and Data Test

Next, create the model. The model has been created through tests and training, with the import of the joblib module. Use the 'dump' function of joblib to save the model object into a file. With an output file named 'knn_model.joblib'.

Create a Model

```
[ ] import joblib

# Save the model to a file
joblib.dump(knn_model, 'knn_model.joblib')

['knn_model.joblib']
```

Picture 10 Create a Model

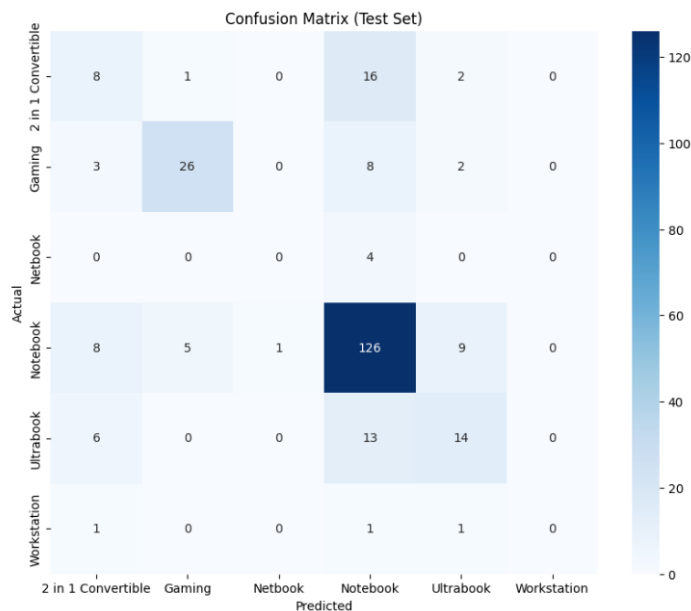
3.4 Data Evaluate

Picture 11 explains that the overall accuracy rate of this model is 68.24%. In the Gaming and Notebook category, the model got an accuracy rate of 81% for the Gaming category and 75% for the Notebook category. In both of these categories, models work well. As for the 2 in 1 Convertible, Netbook and Workstation categories. The model gets an accuracy rate of 30% for the 2 in 1 Convertible category, 0% for the Netbook and Workstation category. Of the three categories, the model works difficulty. As well as for the Ultrabook category, the model gets an accuracy of 50% which means the model works moderately.

Accuracy on the Test Set: 68.24%				
Classification Report for Test Set:				
	precision	recall	f1-score	support
2 in 1 Convertible	0.31	0.30	0.30	27
Gaming	0.81	0.67	0.73	39
Netbook	0.00	0.00	0.00	4
Notebook	0.75	0.85	0.79	149
Ultrabook	0.50	0.42	0.46	33
Workstation	0.00	0.00	0.00	3
accuracy			0.68	255
macro avg	0.40	0.37	0.38	255
weighted avg	0.66	0.68	0.67	255

Picture 11 Model Accuracy

In Picture 12 there is a confusion matrix that explains the categories of 2 in 1 Convertible, Gaming, Netbook, Notebook, Ultrabook and Workstation. On the y-axis 'Actual' which has actual information related to the category. On the x-axis 'Predicted' is the prediction from the model. From the data used, it can be seen that the model predicts the Notebooks corresponding to the actual Notebook data are 126 in total. The highest total. The rest of the model predicts 1 Notebook in Workstation, 13 Notebook in Ultrabook, 4 Notebook in Netbook, 8 Notebook in Gaming and 16 Notebook in 2 in 1 Convertible.



Picture 12 Confusion Matrix

In Picture 13, explain the example of the model. By entering a value of 'ram' worth 6, 'HDD' worth 0, 'SSD' worth 500, 'CPU brand' worth AMD Processor, Lenovo's 'Company Name' and a Windows-valued 'Os type', the model classifies the specification as Ultrabook.

```
Enter RAM size in GB: 6
Enter HDD size in GB: 0
Enter SSD size in GB: 500
Enter CPU brand: AMD Processor
Enter Company Name: Lenovo
Enter Os type: Windows
['Ultrabook']
The predicted laptop type is: Ultrabook
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning:
```

Picture 13 Result Prediction

4. CONCLUSION

From the analysis of making models to classify the type of laptop that has been made. The model uses Python programming language with a tool called Google Collaboratory. The algorithm used to create the model is the KNN algorithm or K-Nearest Neighbor is one of the algorithms that is useful for classification. In this case, it is for the classification of laptop types. The accuracy rate of this model is 68.24% which has a percentage already above 50% means good.

BIBLIOGRAPHY

- [1] M. Romzi and B. Kurniawan, "Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma," *JTIM J. Tek. Inform. Mahakarya*, vol. 03, no. 2, pp. 37–44, 2020.
- [2] Dr. M.J. Garbade, "What is Google Colab?," *Education Ecosystem Blog*. <https://educationecosystem.com/blog/what-is-google-colab/>
- [3] IBM, "What is machine learning?," *IBM*, 2023. <https://www.ibm.com/topics/machine-learning>
- [4] I. Rahardjo, N. Hanafiah, and Y. Setiawan, "Performance of Information Technology Infrastructure Prediction using Machine Learning," *Procedia Comput. Sci.*, vol. 179, no. 2020, pp. 515–523, 2021, doi: 10.1016/j.procs.2021.01.035.
- [5] V. Pulabaigari, "Semi - supervised learning : a brief review," *Int. J. Engineeing Technol.*, no. July, pp. 81–85, 2018, doi: 10.14419/ijet.v7i1.8.9977.
- [6] Z. Zhou, "A brief introduction to weakly supervised learning," *Natl. Sci. Rev.*, vol. 5, pp. 44–53, 2018, doi: 10.1093/nsr/nwx106.
- [7] A. Chahal and P. Gulia, "Machine Learning and Deep Learning," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 12, pp. 4910–4914, 2022, doi: 10.35940/ijitee.L3550.1081219.
- [8] D. Martens and F. Provost, "PREDICTIVE WITH BIG DATA :," vol. 1, no. 4, pp. 215–226, 2013, doi: 10.1089/big.2013.0037.
- [9] Coursera, "What Is Python Used For? A Beginner's Guide," *Coursera*. <https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python>
- [10] IBM, "K-Nearest Neighbors Algorithm," *www.ibm.com*, 2023. <https://www.ibm.com/topics/knn>
- [11] A. Williams, "Laptop specs and terms explained: what to look for when buying a laptop," *T3*. <https://www.t3.com/features/laptop-specs-and-terms-explained>