# JTI Skripsi Sinta 2 - Dimas Trianda (2101011) - Turnitin.docx

*by* Jaidyn Sylvester

**Submission date:** 06-Nov-2024 03:30AM (UTC-0500)
**Submission ID:** 2510220271
**File name:** JTI_Skripsi_Sinta_2_-_Dimas_Trianda_2101011_-_Turnitin.docx (4.47M)
**Word count:** 4075
**Character count:** 22667

Jurnal Teknik Informatika

Homepage : http://journal.uinjkt.ac.id/index.php/ti

# OPTIMIZATION OF K-NEAREST NEIGHBOR (KNN) PERFORMANCE IN PREDICTING HYPERTENSION RISK USING GRIDSEARCHCV

Dimas Trianda[1], Dedy Hartama[2], Solikhun[3]

[1,2,3]Informatics Engineering Study Program, STIKOM Tunas Bangsa, Pematang Siantar, Indonesia
[1,2,3]Jl. Sudirman, Pematangsiantar, Indonesia
E-mail: [1]trianda28721@gmail.com, [2]dedyhartama@amiktunasbangsa.ac.id, [3]solikhun@amiktunasbangsa.ac.id

## ABSTRACT

Hypertension is one of the leading causes of cardiovascular disease, necessitating early prediction and detection of hypertension risk. The objective of this study is to optimize the *K-Nearest Neighbor* (KNN) algorithm in predicting hypertension risk using *GridSearchCV*. The dataset used was sourced from Kaggle and includes 2,340 samples with 11 health attributes such as age, gender, and cholesterol levels. After data processing, the KNN model was optimized with *GridSearchCV*, evaluating combinations of hyperparameters such as the number of neighbors (k), weight type, and distance metric. As a result, the best model is an accuracy of 86% for the testing data with *GridSearchCV*, representing a 4% increase compared to KNN without *GridSearchCV*, which only achieved an accuracy of 82%. Additionally, the *ROC AUC Score* of 0.9197 indicated good classification performance. This study concludes that hyperparameter optimization using *GridSearchCV* can significantly enhance the predictive performance of the KNN model in classifying hypertension risk.

**Keywords:** *Optimization, Hypertension, Machine Learning, KNN, GridSearchCV*

## 1. INTRODUCTION

Hypertension, also called hypertension, it is a long-term condition characterized by increased arterial pressure, which plays a significant role in causing cardiovascular diseases like heart attacks and strokes [1] ,[2]. Worldwide, more than one billion people suffer from hypertension and its prevalence is increasing, particularly in countries with low and middle incomes. China has the highest number of hypertension cases, with 226 million people affected, followed by India with 200 million. On the other hand, the number of hypertension cases in developed countries has decreased [3], [4]. The increasing burden of hypertension underscores the need for effective early prediction and detection to enable timely medical interventions and improve patient health outcomes [5]. With the growing reliance on data-driven approaches in healthcare, predictive models have become essential tools in managing chronic diseases like hypertension.

One such data-driven approach is data mining, which has proven effective in analyzing large datasets and uncovering hidden patterns

[6]. In the healthcare sector, data mining techniques are applied to predict, classify, and analyze disease outcomes, providing valuable insights that can guide clinical decision-making [7]. Machine learning, an integral part of data mining, is particularly useful for disease prediction, including hypertension, by processing patient data to identify key risk factors [8], [9]. Among the various *machine learning* algorithms, *K-Nearest Neighbor* (KNN) is widely appreciated for its simplicity and effectiveness in classification tasks [10]. KNN's ability to handle complex multidimensional data makes it a suitable choice for predicting hypertension risk based on factors such as age, gender, and cholesterol levels [11], [12].

However, despite its effectiveness, one limitation of KNN lies in determining the optimal hyperparameters, such as the number of neighbors or the distance metric, which greatly affects the model's performance [13], [14]. This is why *hyperparameter tuning* is crucial [15], [16]. *Hyperparameter tuning* involves selecting the best set of parameters to enhance the accuracy, precision, overall model performance [17]. Traditional manual tuning methods are often inefficient and prone to errors, especially when dealing with large data sets and complex models [18]. To address these challenges, automated techniques have been developed to expedite this tuning process [19].

One of the most popular techniques for hyperparameter optimization is *GridSearch Cross-Validation* (*GridSearchCV*). This method systematically searches through a pre-defined set of hyperparameter combinations, evaluating each combination using cross-validation to minimize *overfitting* is performed to ensure that the model generalizes well to the new data [20], [21]. By assessing model performance across multiple data subsets, *GridSearchCV* identifies the optimal hyperparameter combination to maximize metrics like accuracy [22]. This comprehensive search process makes it highly effective for optimizing machine learning algorithms such as KNN, particularly in healthcare applications [23].

Previous studies that serve as references for this research include the study by Sudriyanto, which focused on neural network optimization for hypertension prediction using the Particle Swarm Optimization (PSO) algorithm. This research combined Neural Networks (NN) with PSO to optimize weights and biases, resulting in improved prediction performance. The dataset used included hypertension data with variables such as age, gender, blood pressure, cholesterol levels, and others. Experimental results showed that Neural Networks with PSO produced a lower Root Mean Squared Error (RMSE) value (0.170) compared to Neural Networks without PSO (0.197), indicating improved accuracy in predicting hypertension risk [24].

Another study by Ongkosianbhadra and Lestari developed a hypertension risk predictive model using the Gradient-Boosting Decision Tree (GBDT) algorithm optimized with various *hyperparameter tuning* methods, including Tree Parzen Estimation, which achieved the highest accuracy of 74.43%. The dataset used comprised 70,693 rows of information provided by the Centers for Disease Control and Prevention (CDC), with 17 attributes covering behavior, medical history, and health status. The results demonstrated that hyperparameter optimization with Tree Parzen Estimation improved model performance, producing more accurate hypertension risk predictions compared to other tuning methods such as GridSearch and *Bayesian Optimization* [25].

Unlike these approaches, this research uses the KNN algorithm optimized with GridSearch Cross-Validation, which represents a novel approach in classifying hypertension data. Additionally, this study focuses on KNN hyperparameter optimization through *GridSearchCV*, a technique that has not been widely applied to KNN in the context of hypertension prediction.

This research focuses on enhancing the performance of the KNN algorithm for predicting hypertension risk by utilizing *GridSearchCV* for *hyperparameter tuning*. The goal is to optimize the KNN model by finding the best hyperparameters that can improve its ability to classify individuals based on key risk factors. By combining the strengths of KNN and the systematic optimization provided by *GridSearchCV*, the goal of this study is to create a reliable predictive model that can assist healthcare professionals in identifying high-risk individuals, thereby enabling earlier intervention and reducing the health burden associated with hypertension.

## 2. RESEARCH METHODE

The dataset used in this research was obtained from an open-source relevant to hypertension risk factors, including information such as age, gender, body mass index, cholesterol levels, blood pressure, and other health history. The data was sourced from Kaggle via the following link: https://www.kaggle.com/datasets/khan1803115/hypertension-risk-model-main?resource=download, comprising a sample of 2,340 patient records.

Before analysis, the data underwent a cleaning and *preprocessing* stage, where each entry was checked to ensure there were no missing or incomplete data. Subsequently, the data was split into two parts: the *training set* and the *test set*, to ensure that the model could adapt and provide accurate results for previously unseen data. This process guarantees that the dataset used can yield reliable and representative predictions for hypertension.

All analyses were conducted using Jupyter Notebook. The research process is further explained through the steps presented in Figure 1.

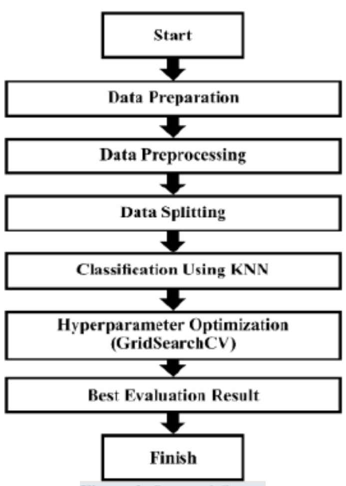



**Figure 1.** *Research Stages*

### 2.1. Data Preparation

This stage involves collecting a hypertension dataset that includes various features related to patient health metrics, such as gender, age, BMI, cholesterol levels, blood pressure, and more. The data was obtained from open sources, such as Kaggle. This dataset forms the foundation of the research, making it essential to ensure that the data used is relevant and representative.

### 2.2. Data *Preprocessing*

Data *preprocessing* is crucial for enhancing model accuracy and ensuring consistent results. At this stage, missing values in the dataset are handled using *SimpleImputer*, which can fill gaps with statistical measures such as the mean. Additionally, *MinMaxScaler* is used to scale features so that all variables have a uniform range (usually between 0 and 1), which allows the model to process data more efficiently without being affected by large numbers.

### 2.3. Splitt Data

After *pre-processing*, the dataset is split into two parts: *training* data and *testing* data, using the *train_test_split* function. The *training* data is utilized to train the KNN model, while the *test* data is reserved for evaluating the model's performance. This division is crucial to ensure that the model can generalize well to new and unseen data.

### 2.4. Classification Using KNN

At this stage, the *K-Nearest Neighbor* (KNN) algorithm is used to classify patients as either at risk or not at risk of hypertension. A *pipeline* is created to streamline the classification process, integrating data *preprocessing* steps and the KNN algorithm into a single workflow. This *pipeline* simplifies the process and ensures that all steps are applied consistently during both the training and testing phases.

### 2.5. Hyperparameter Optimization (*GridSearchCV*)

*Hyperparameter tuning* is performed using *GridSearchCV* to enhance the performance of the KNN model. This process involves testing various values for key parameters, such as the number of neighbors (k), type of weights, and distance metrics, to identify the combination that provides the best accuracy. By systematically exploring these parameters, the model is optimized to be more effective for the specific dataset.

### 2.6. Validation and Evaluation

To evaluate the performance of the applied model, validation is conducted using k-fold cross-validation. In this study, k-fold cross-validation is implemented with a training-to-testing data ratio of 8:2, with variations in the value of k ranging from 1 to 50. The model will be evaluated by assessing the performance of an optimized KNN model using *GridSearchCV* for *hyperparameter tuning*.

## 3. RESULTS AND DISCUSSION

The study comprises seven main stages: data preparation, *preprocessing*, data splitting into training and testing sets, classification using the KNN method, hyperparameter optimization using *GridSearchCV* for the KNN model, comparing KNN performance before and after hyperparameter optimization, and finally, obtaining the best classification results based on evaluation metrics such as accuracy.

### 3.1. Preparing Data For Classification

This research utilizes a 2024 dataset sourced from Kaggle. The dataset contains several key features for predicting hypertension, including gender (coded as 0 for female, 1 for male - X1), age (X2), smoking status (coded as 0 for non-smoker, 1 for smoker - X3), daily cigarette consumption (X4), diabetes status (coded as 0 for non-diabetic, 1 for diabetic - X5), cholesterol level (X6), systolic blood pressure (X7), diastolic blood pressure (X8), body mass index (X9), heart rate per minute (X10), glucose level (X11), and the final outcome of hypertension risk (coded as 0 for not at risk, 1 for at risk).

As shown in Table 1, the dataset consists of 12 features with a total of 2,340 data entries. Out of these 12 features, one serves as the target variable, while the remaining 11 act as inputs. The feature designated as the target is "risk," while the 11 features analyzed in this study include gender, age, smoking status, daily cigarette consumption, diabetes presence, cholesterol level, systolic blood pressure, diastolic blood pressure, body mass index, heart rate per minute, and glucose level.

Before building the machine learning model, the dataset must go through *preprocessing* steps and ensure that the classes are balanced, making it ready for further analysis.

**Tabel 1.** *Dataset used*

| No. | X1 | X2 | X3 | X4 | X5 | ... | X11 | Risk |
|---|---|---|---|---|---|---|---|---|
| 1. | 1 | 39 | 0 | 0 | 0 | ... | 77 | 0 |
| 2. | 0 | 46 | 0 | 0 | 0 | ... | 76 | 0 |
| 3. | 1 | 48 | 1 | 20 | 0 | ... | 70 | 0 |
| 4. | 0 | 61 | 1 | 30 | 0 | ... | 103 | 1 |
| 5. | 0 | 46 | 1 | 23 | 0 | ... | 85 | 0 |
| 6. | 0 | 43 | 0 | 0 | 0 | ... | 99 | 1 |
| 7. | 0 | 63 | 0 | 0 | 0 | ... | 85 | 0 |
| 8. | 0 | 45 | 1 | 20 | 0 | ... | 78 | 0 |
| 9. | 1 | 52 | 0 | 0 | 0 | ... | 79 | 1 |
| 10. | 1 | 43 | 1 | 30 | 0 | ... | 88 | 1 |
| 11. | 0 | 50 | 0 | 0 | 0 | ... | 76 | 0 |
| 12. | 0 | 43 | 0 | 0 | 0 | ... | 61 | 0 |
| 13. | 1 | 46 | 1 | 15 | 0 | ... | 64 | 1 |
| 14. | 0 | 41 | 0 | 0 | 0 | ... | 84 | 1 |
| 15. | 0 | 38 | 1 | 20 | 0 | ... | 70 | 1 |
| 16. | 1 | 48 | 1 | 10 | 0 | ... | 72 | 1 |
| 17. | 0 | 46 | 1 | 20 | 0 | ... | 89 | 0 |
| 18. | 0 | 38 | 1 | 5 | 0 | ... | 78 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 2.340. | 1 | 40 | 0 | 0 | 0 | ... | 72 | 1 |

### 3.2. Data Splitting

The data is divided into two parts: the *training set* and the *test set*. This division is made so that the model can learn from the training data and then be tested with the test data to evaluate its performance. The data split is shown in Figure 2.

### Split Data

```
[238]: X_train, X_test, y_train, y_test = train_test_split
       (X, y, test_size=0.2, stratify=y, random_state=42)
       X_train.shape, X_test.shape, y_train.shape, y_test.shape

[238]: ((1872, 11), (468, 11), (1872,), (468,))
```

**Figure 2.** *Data Split Process*

In the code above, the input variables are stored in X, while the target variable is stored in y. The data is split use an 80:20 ratio, 80% of the data for *training* and 20% for *testing*, using the *train_test_split* method in the *sklearn.model_selection* library to split the data randomly, with the parameter *stratify*=y to ensure that the class distribution remains balanced between the *training* and *test sets*. A total of 1,872 data points are used for training, and 468 data points are used for testing.

## 3.3. Building the KNN Model Without *GridSearchCV*

Before applying *GridSearchCV* for hyperparameter optimization, an initial model is built using the KNN algorithm with K = 5. At this stage, the model is trained using the dataset that has been split into 80:20 training and test data.

The KNN model can be seen in Figure 3.

### KNN without GridSearchCV

```
[250]: knn_model = KNeighborsClassifier(n_neighbors=5)
       knn_model.fit(X_train_scaled, y_train)

[250]: KNeighborsClassifier
       KNeighborsClassifier()
```

**Figure 3.** *Building a KNN Model Without GridSearchCV*

Next, in the *confusion matrix* for KNN without *GridSearchCV*, as shown in Figure 4, the results from the *confusion matrix* indicate that out of the total 468 test data, there are 199 *true positives*, 187 *true negatives*, 35 *false positives*, and 47 *false negatives*.

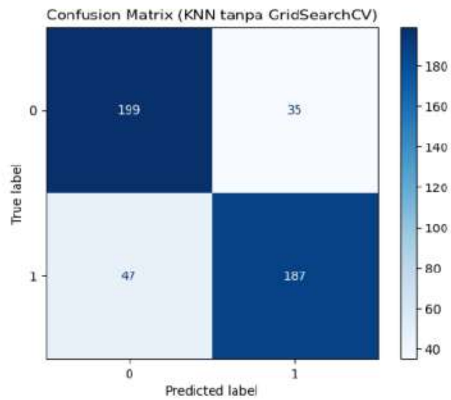



**Figure 4.** *Confusion Matrix KNN Without GridSearchCV*

Next are the prediction results and evaluation generated from the KNN model without *GridSearchCV*.

### Predict and Evaluate

```
[253]: y_pred_knn = knn_model.predict(X_test_scaled)
       print("Classification Report (KNN tanpa GridSearchCV):\n",
             classification_report(y_test, y_pred_knn))

Classification Report (KNN tanpa GridSearchCV):
              precision    recall  f1-score   support

           0       0.81      0.85      0.83       234
           1       0.84      0.80      0.82       234

    accuracy                           0.82       468
   macro avg       0.83      0.82      0.82       468
weighted avg       0.83      0.82      0.82       468
```

**Figure 5.** *Prediction Results and Evaluation of KNN Without GridSearchCV*

The training and testing scores for the KNN model without *GridSearchCV* can be seen in Figure 6.

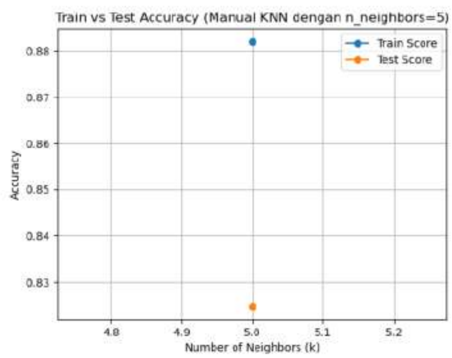


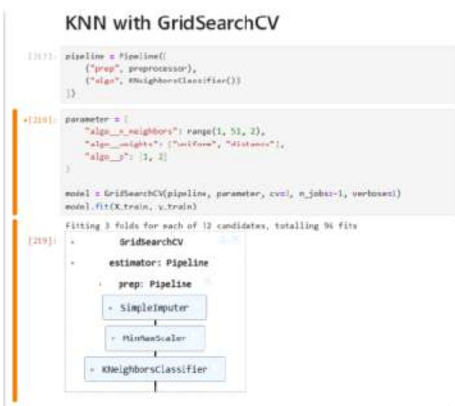

**Figure 6.** *Plot Training vs Testing Accuracy KNN Without GridSearchCV*

As shown in Figure 6, the accuracy obtained from the training data reached 88%, indicating that the model can recognize patterns in the training data well. However, when the model was tested using the test data, the accuracy dropped to 82%. This difference in accuracy suggests the presence of *overfitting*, where the model is too fitted to the training data and struggles to generalize to new, unseen data.

*Overfitting* occurs when the model learns too many details and noise from the training data, leading to decreased performance on the test data. Therefore, the next step is to perform hyperparameter optimization using *GridSearchCV* to find the optimal value for K, in order to improve accuracy and reduce the risk of *overfitting* with the help of K-Fold *Cross Validation*.

### 3.4. KNN with *GridSearchCV*

At this stage, *Hyperparameter Tuning* is performed using *GridSearchCV* to find the best combination of parameters for the KNN model. The range of values explored for hyperparameters includes K values from 1 to 50, weight options of uniform and distance, and p values tested as 1 (Manhattan) and 2 (Euclidean). This process is conducted using cross-validation with 3 different scenarios. The KNN model with *GridSearchCV* can be seen in Figure 7.


KNN with GridSearchCV

```
[217]: pipeline = Pipeline([
           ("prep", preprocessor),
           ("algo", KNeighborsClassifier())
       ])

[219]: parameter = {
           "algo__n_neighbors": range(1, 51, 2),
           "algo__weights": ["uniform", "distance"],
           "algo__p": [1, 2]
       }

       model = GridSearchCV(pipeline, parameter, cv=3, n_jobs=-1, verbose=1)
       model.fit(X_train, y_train)

       Fitting 3 folds for each of 12 candidates, totalling 96 fits
[219]: GridSearchCV
       estimator: Pipeline
       prep: Pipeline
       SimpleImputer
       MinMaxScaler
       KNeighborsClassifier
```

**Figure 7**. *Building a KNN Model with GridSearchCV*

Next, in the *confusion matrix* for KNN with *GridSearchCV*, as shown in Figure 8, the results indicate that out of a total of 468 *testing data*, there are 204 *true positive* cases, 199 *true negative* cases, 30 *false positive* cases, and 35 *false negative* cases.

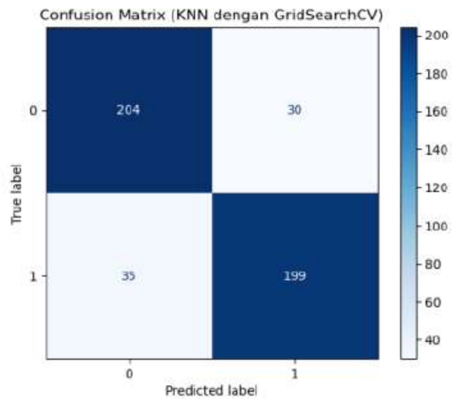
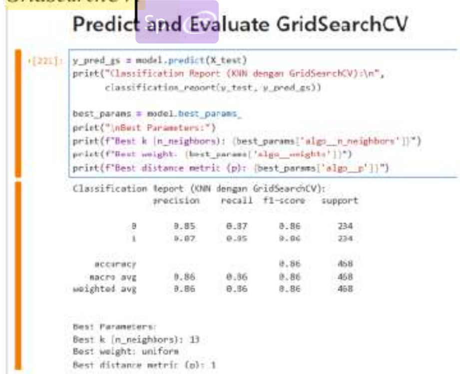



**Figure 8**. *Confusion Matrix KNN with GridSearchCV*

The following are the prediction results and evaluation produced by the KNN model with *GridSearchCV*


Predict and Evaluate GridSearchCV

```
[221]: y_pred_gs = model.predict(X_test)
       print("Classification Report (KNN dengan GridSearchCV):\n",
             classification_report(y_test, y_pred_gs))

       best_params = model.best_params_
       print("\nBest Parameters:")
       print(f"Best k (n_neighbors): {best_params['algo__n_neighbors']}")
       print(f"Best weight: {best_params['algo__weights']}")
       print(f"Best distance metric (p): {best_params['algo__p']}")

Classification Report (KNN dengan GridSearchCV):
              precision    recall  f1-score   support

           0       0.85      0.87      0.86       234
           1       0.87      0.85      0.86       234

    accuracy                           0.86       468
   macro avg       0.86      0.86      0.86       468
weighted avg       0.86      0.86      0.86       468


Best Parameters:
Best k (n_neighbors): 13
Best weight: uniform
Best distance metric (p): 1
```

**Figure 9.** *Prediction Results and Evaluation of KNN With GridSearchCV*

The training and testing scores for the KNN model with *GridSearchCV* can be seen in Figure 10.


**Figure 10.** *Plot Training vs Testing Accuracy KNN With GridSearchCV*

The tuning results indicate that the best parameters for the model are K value of 13, p of

1 (Manhattan), and uniform weights. The accuracy achieved by this model is 85% for the training data and 86% for the testing data, indicating that there is no *overfitting* as the training and testing accuracies are fairly balanced.

The model also achieved an *ROC AUC Score* of 0.9197, which demonstrates good performance in classification. The *ROC Curve* can be seen in Figure 11.

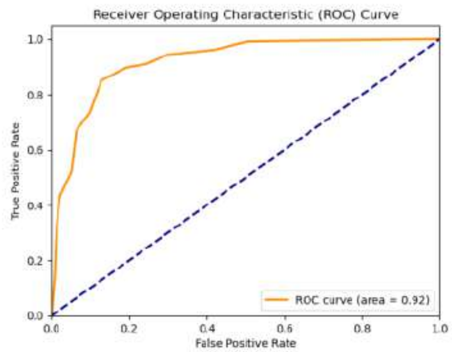



**Figure 11.** *ROC Curve*

3.5. Comparison Between KNN Without *GridSearchCV* and KNN With *GridSearchCV*

To compare the best accuracy between KNN and KNN optimized with *GridSearchCV*, it is essential to evaluate their training results and performance. The standard KNN classification process involves calculating the distance between the test data and its nearest neighbors and manually selecting the optimal number of neighbors (k).

In contrast, KNN optimized with *GridSearchCV* automatically searches for the best parameter combinations, including the k value and distance metric, using cross-validation. This technique enables the model to find the parameter configuration that yields the highest accuracy, reduces the risk of *overfitting*, and enhances the stability of model performance.

To compare the performance of both approaches, the first step is to train both the standard KNN and the KNN with *GridSearchCV* using the same dataset. The next step is to evaluate their accuracy on the test data. The test results show that KNN with *GridSearchCV* achieved an accuracy of 86%, which is higher than the standard KNN's accuracy of 82%. This confirms that parameter optimization using *GridSearchCV* can significantly improve model performance. The accuracy comparison between the KNN method and the KNN optimized with *GridSearchCV* is displayed in Table 2.

**Tabel 2.** *Comparison Between KNN Without GridSearchCV and KNN With GridSearchCV*

| Metode | Akurasi |
|---|---|
| KNN without *GridSearchCV* | 82% |
| KNN with *GridSearchCV* | 86% |

## 3.6. *Exploratory Data Analysis (EDA)*

Next, the researcher analyzed the relationship between condition attributes and the target using *exploratory data analysis (EDA)* to identify the conditions that contribute to the highest risk of hypertension. The EDA process can be seen in Figure 12.

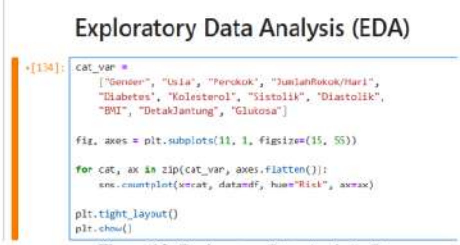



**Figure 12.** *Exploratory Data Analysis Process*

From the EDA process, the data identified the highest-risk factors as follows:

**Tabel 3.** *Most Risky Conditions*

| *Exploratory Data Analysis (EDA)* **Plot Results** | |
|---|---|
| Gender | Women |
| Age | 46 – 63 years |
| Smoker | Do not smoke |
| Number of cigarettes/day | 0 |
| History of diabetes | No diabetes |
| Cholesterol | 200 - 250 |
| Systolic blood pressure | 140 - 180 |
| Diastolic blood pressure | 88 - 98 |
| BMI | 24 - 30 |
| Heart rate | 75 / minute |
| Glucose | 60 - 80 |

## 4. CONCLUSION

This study successfully optimized the performance of the *K-Nearest Neighbor* (KNN) algorithm for predicting hypertension risk using *GridSearchCV*. With this approach, the optimized KNN model achieved an accuracy of

85% on the training data and 86% on the testing data, indicating the model's ability to generalize well without *overfitting*. The *ROC AUC Score* of 0.9197 demonstrates that the model has excellent classification capability. These findings underscore the importance of *hyperparameter tuning* in enhancing the performance of machine learning models, especially in the context of predicting hypertension risk.

This research is expected to aid healthcare professionals in early detection of hypertension risk and help reduce the public health burden. For future research, it is recommended that this model be evaluated with a larger and more diverse dataset to improve the generalizability of the results. The use of other optimization methods, such as *Random Search* or *Bayesian Optimization*, could also be considered to compare the effectiveness of *hyperparameter tuning*. Implementing additional features or conducting deeper analysis on the influence of specific attributes on predictions could provide more valuable insights for predicting hypertension risk.

## BIBLIOGRAPHY


[1] E. A. Setyarini, A. Monica, F. Sinaga, F. X. Widiantoro, L. T. Listianingsih, and C. Cecilia, "Illness Perception and Self-management of Elderly With Hypertention," *Malaysian J. Med. Heal. Sci.*, vol. 19, no. Supplement 9, pp. 97–101, 2023, doi: 10.47836/MJMHS.19.S9.14.

[2] M. C. Anvarovna and A. G. Abdunabievna, "THE STUDY OF THE CHARACTERISTICS OF CARDIOMETABOLIC RISKFACTORS IN YOUNG PEOPLE WITH ARTERIAL HYPERTENTION," *Int. Sci. J.*, vol. 2, no. 7, pp. 3–8, 2023.

[3] S. A. Nasir, "Labetalol against Alpha Methyldopa for the treatment gestational hypertention," *Rev. Latinoam. Hipertens.*, vol. 18, no. 8, pp. 394–399, 2023, doi: 10.5281/zenodo.10214339.

[4] M. Milasari, Anita Agustina, Y. S. Pramono, M. Rauf, and D. K. Wulandari, "Modality Therapy : Calisthenics for Elderly with Hypertention In Tanipah Village," *OMNICODE J. (Omnicompetence Community Dev. Journal)*, vol. 1, no. 2, pp. 87–89, 2022, doi: 10.55756/omnicode.v1i2.99.

[5] T. Elisabhet, "Analysis of the Level Anxiety Related To the Self Efficacy of Eldelry with Hypertention," *Indones. J. Glob. Heal. Res.*, vol. 6, no. 2, pp. 851–856, 2024, doi: https://doi.org/10.37287/ijghr.v6i2.3055.

[6] A. Nandy, C. Duan, and H. J. Kulik, "Using Machine Learning and Data Mining to Leverage Community Knowledge for the Engineering of Stable Metal-Organic Frameworks," *J. Am. Chem. Soc.*, vol. 143, no. 42, pp. 17535–17547, 2021, doi: 10.1021/jacs.1c07217.

[7] A. Namoun and A. Alshanqiti, "Predicting student performance using data mining and learning analytics techniques: A systematic literature review," *Appl. Sci.*, vol. 11, no. 1, pp. 1–28, 2021, doi: 10.3390/app11010237.

[8] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A Survey on Distributed Machine Learning," *ACM Comput. Surv.*, vol. 53, no. 2, 2020, doi: 10.1145/3377454.

[9] S. Sharma and P. Chaudhary, "Machine learning and deep learning," *Springer*, pp. 71–84, 2023, doi: 10.1515/9783110791402-004.

[10] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, *A survey on missing data in machine learning*, vol. 8, no. 1. Springer International Publishing, 2021. doi: 10.1186/s40537-021-00516-9.

[11] Richa Nanda Fitria, Wahyu Sugianto, and Amalia Cemara Nur'aidha, "Prediksi Penyakit Diabetes Mellitus Tipe I dan Tipe II Menggunakan Metode KNN di Klinik Dharma Husada," *Antigen J. Kesehat. Masy. dan Ilmu Gizi*, vol. 2, no. 3, pp. 39–53, 2024, doi: 10.57213/antigen.v2i3.303.

[12] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction," *Sci. Rep.*, vol. 12, no. 1, pp. 1–11, 2022, doi:

10.1038/s41598-022-10358-x.

[13] N. Alturki *et al.*, "Improving Prediction of Chronic Kidney Disease Using KNN Imputed SMOTE Features and TrioNet Model," *C. - Comput. Model. Eng. Sci.*, vol. 139, no. 3, pp. 3513–3534, 2024, doi: 10.32604/cmes.2023.045868.

[14] P. Dinesh, A. S. Vickram, and P. Kalyanasundaram, "Medical image prediction for diagnosis of breast cancer disease comparing the machine learning algorithms: SVM, KNN, logistic regression, random forest and decision tree to measure accuracy," *AIP Conf. Proc.*, vol. 2853, no. 1, 2024, doi: 10.1063/5.0203746.

[15] S. F. M. Radzi, M. K. A. Karim, M. I. Saripan, M. A. A. Rahman, I. N. C. Isa, and M. J. Ibahim, "Hyperparameter tuning and pipeline optimization via grid search method and tree-based autoML in breast cancer prediction," *J. Pers. Med.*, vol. 11, no. 10, 2021, doi: 10.3390/jpm11100978.

[16] W. Pannakkong, K. Thiwa-Anont, K. Singthong, P. Parthanadee, and J. Buddhakulsomsiri, "Hyperparameter Tuning of Machine Learning Algorithms Using Response Surface Methodology: A Case Study of ANN, SVM, and DBN," *Math. Probl. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/8513719.

[17] J. Rijsdijk, L. Wu, G. Perin, and S. Picek, "Reinforcement learning for hyperparameter tuning in deep learning-based side-channel analysis," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2021, no. 3, pp. 677–707, 2021, doi: 10.46586/tches.v2021.i3.677-707.

[18] A. Stuke, P. Rinke, and M. Todorovic, "Efficient hyperparameter tuning for kernel ridge regression with Bayesian optimization," *Mach. Learn. Sci. Technol.*, vol. 2, no. 3, 2021, doi: 10.1088/2632-2153/abee59.

[19] J. Isabona, A. L. Imoize, and Y. Kim, "Machine Learning-Based Boosted Regression Ensemble Combined with Hyperparameter Tuning for Optimal Adaptive Learning," *Sensors*, vol. 22, no. 10, 2022, doi: 10.3390/s22103776.

[20] S. T. Kusuma and T. B. Sasongko, "Optimasi K-Nearest Neighbor dengan Grid Search CV pada Prediksi Kanker Paru-Paru," *Indones. J. Comput. Sci.*, vol. 12, no. 4, pp. 2162–2171, 2023, doi: 10.33022/ijcs.v12i4.3267.

[21] Ankit Kumar and Dr. Deepak Sharma, "Intrusion Detection System using Multi-Layer Perceptron with Grid Search CV," *Int. J. Mod. Trends Sci. Technol.*, vol. 8, no. 7, pp. 98–101, 2022, doi: 10.46501/ijmtst0807016.

[22] K. Kavitha, N. Tangudu, S. R. Sahu, G. V. L. Narayana, and V. Anusha, "Detection of PCOS using Machine Learning Algorithms with Grid Search CV Optimization," *Int. J. Eng. Trends Technol.*, vol. 71, no. 7, pp. 201–208, 2023, doi: 10.14445/22315381/IJETT-V71I7P219.

[23] W. Nugraha and A. Sasongko, "Hyperparameter Tuning pada Algoritma Klasifikasi dengan Grid Search," *Sist. J. Sist. Inf.*, vol. 11, no. 2, pp. 391–401, 2022, doi: https://doi.org/10.32520/stmsi.v11i2.1750.

[24] S. Sudriyanto, "Optimizing Neural Networks Using Particle Swarm Optimization (PSO) Algorithm for Hypertension Disease Prediction," *JEECOM J. Electr. Eng. Comput.*, vol. 5, no. 2, pp. 278–284, 2023, doi: 10.33650/jeecom.v5i2.6759.

[25] F. V. Ongkosianbhadra and C. C. Lestari, "Pengembangan Model Prediksi Risiko Hipertensi Menggunakan Algoritma Gradient Boosting Decision Tree yang Dioptimalkan dengan Hyperparameter Tuning Tree Parzer Estimation," *J. Inform. dan Sist. Inf.*, vol. 9, no. 2, pp. 35–44, 2023, doi: 10.37715/juisi.v9i2.4403.

# JTI Skripsi Sinta 2 - Dimas Trianda (2101011) - Turnitin.docx

ORIGINALITY REPORT

| 17% | 11% | 10% | 7% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to UIN Syarif Hidayatullah Jakarta<br>Student Paper | 2% |
| 2 | Submitted to University of Sunderland<br>Student Paper | 1% |
| 3 | keep.lib.asu.edu<br>Internet Source | 1% |
| 4 | Mahadi Hasan Imran, Mohammad Ilyas Khan, Shahrizan Jamaludin, Ibnul Hasan et al. "A critical analysis of machine learning in ship, offshore, and oil & gas corrosion research, part I: Corrosion detection and classification", Ocean Engineering, 2024<br>Publication | 1% |
| 5 | Pawan Singh Mehra, Dhirendra Kumar Shukla. "Artificial Intelligence, Blockchain, Computing and Security - Volume 2", CRC Press, 2023<br>Publication | 1% |
| 6 | Submitted to Politeknik Manufaktur Negeri Bangka Belitung<br>Student Paper | 1% |

| | | |
|---|---|---|
| 7 | www.science.gov<br>Internet Source | 1 % |
| 8 | www.yosnalab.com<br>Internet Source | 1 % |
| 9 | www.researchgate.net<br>Internet Source | 1 % |
| 10 | Submitted to Liverpool John Moores University<br>Student Paper | <1 % |
| 11 | www.dovepress.com<br>Internet Source | <1 % |
| 12 | www.frontiersin.org<br>Internet Source | <1 % |
| 13 | Submitted to The University of the West of Scotland<br>Student Paper | <1 % |
| 14 | Submitted to Northcentral<br>Student Paper | <1 % |
| 15 | journal.unipdu.ac.id<br>Internet Source | <1 % |
| 16 | www.scitepress.org<br>Internet Source | <1 % |
| 17 | Submitted to Queensland University of Technology<br>Student Paper | <1 % |

| | | |
|---|---|---|
| 18 | www.researchsquare.com<br>Internet Source | <1 % |
| 19 | journal.uinjkt.ac.id<br>Internet Source | <1 % |
| 20 | jurnal.iaii.or.id<br>Internet Source | <1 % |
| 21 | www.simplilearn.com<br>Internet Source | <1 % |
| 22 | Alex X. Wang, Stefanka S. Chukova, Binh P. Nguyen. "Ensemble k-nearest neighbors based on centroid displacement", Information Sciences, 2023<br>Publication | <1 % |
| 23 | Submitted to University of Huddersfield<br>Student Paper | <1 % |
| 24 | Yikai Lu, Lingbo Tong, Alison Cheng. "Advanced Knowledge Tracing for Intelligent Tutoring Systems: Incorporating Process Data and Curricula Information via an Attention-Based Framework", Open Science Framework, 2023<br>Publication | <1 % |
| 25 | Yuan Zhao, Weiguo Zhang, Xiufeng Liu. "Grid search with a weighted error function: Hyper-parameter optimization for financial time | <1 % |

series forecasting", Applied Soft Computing, 2024
Publication

| | | |
|---|---|---|
| 26 | journal.unnes.ac.id<br>Internet Source | <1 % |
| 27 | m.moam.info<br>Internet Source | <1 % |
| 28 | pubmed.ncbi.nlm.nih.gov<br>Internet Source | <1 % |
| 29 | ruidera.uclm.es<br>Internet Source | <1 % |
| 30 | Balqis Albreiki, Nazar Zaki, Hany Alashwal. "A Systematic Literature Review of Student' Performance Prediction Using Machine Learning Techniques", Education Sciences, 2021<br>Publication | <1 % |
| 31 | Grant Charles Mwakipunda, Norga Alloyce Komba, Allou Koffi Franck Kouassi, Edwin Twum Ayimadu et al. "Prediction of hydrogen solubility in aqueous solution using modified mixed effects random forest based on particle swarm optimization for underground hydrogen storage", International Journal of Hydrogen Energy, 2024<br>Publication | <1 % |

| | | |
|---|---|---|
| 32 | Keshinro, Babatunde Ibrahim. "Human Activity Recognition Using Deep Learning Methods for Human-Robot Interaction", North Carolina Agricultural and Technical State University, 2024<br>Publication | <1 % |
| 33 | Md. Mehedi Hassan, Md. Mahedi Hassan, Farhana Yasmin, Md. Asif Rakib Khan et al. "A comparative assessment of machine learning algorithms with the Least Absolute Shrinkage and Selection Operator for breast cancer detection and prediction", Decision Analytics Journal, 2023<br>Publication | <1 % |
| 34 | arxiv.org<br>Internet Source | <1 % |
| 35 | cointelegraph.com<br>Internet Source | <1 % |
| 36 | journal2.uad.ac.id<br>Internet Source | <1 % |
| 37 | jurnal.stikeskesosi.ac.id<br>Internet Source | <1 % |
| 38 | thesai.org<br>Internet Source | <1 % |
| 39 | www.igi-global.com<br>Internet Source | <1 % |

| | | | |
|---|---|---|---|
| 40 | www.mdpi.com<br>Internet Source | | <1 % |
| 41 | Christian Micheau. "Modes of invasion of cancer of the larynx.A statistical, histological, and radioclinical analysis of 120 cases", Cancer, 07/1976<br>Publication | | <1 % |
| 42 | Satya Tegar Kusuma, Theopilus Bayu Sasongko. "Optimasi K-Nearest Neighbor dengan Grid Search CV pada Prediksi Kanker Paru-Paru", The Indonesian Journal of Computer Science, 2023<br>Publication | | <1 % |

| | | | |
|---|---|---|---|
| Exclude quotes | Off | Exclude matches | Off |
| Exclude bibliography | On | | |

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Missing "," You may need to place a comma after this word.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Article Error You may need to remove this article.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Missing "," You may need to place a comma after this word.

Missing "," You may need to place a comma after this word.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Article Error** You may need to use an article before this word.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



**Article Error** You may need to remove this article.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Missing ","** You may need to place a comma after this word.

**P/V** You have used the passive voice in this sentence. Depending upon what you wish to emphasize in the sentence, you may want to revise it using the active voice.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Article Error** You may need to use an article before this word.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Article Error** You may need to use an article before this word.

**Article Error** You may need to use an article before this word.

**Missing ","** You may need to place a comma after this word.

**Article Error** You may need to remove this article.



**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Article Error** You may need to use an article before this word.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**S/V** This subject and verb may not agree. Proofread the sentence to make sure the subject agrees with the verb.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Article Error** You may need to remove this article.

**Missing ","** You may need to place a comma after this word.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Confused You have used **an** in this sentence. You may need to use **a** instead.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

Sp. This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



---

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.

**Sp.** This word is misspelled. Use a dictionary or spellchecker when you proofread your work.



---



---