Basic and Applied Computing for Digital Innovation (BACODING)
Volume 01, No. 01, February, 2026, pp. 22-30
Received: 01 January 2026; Revised: 05 January 2026; Accepted: 05 February 2026
https://bacoding.id/index.php/journal/
DOI :

Research article

Estimating the Price of Koi Fish Using The K-Nearest Neighbor Method

Satria Dwi Surya[1], Gede Putra Aditya Brahmantha[2], Oki Ria Hermawan[3], Ni'matur Rohim[4], Kusrini[5], Dina Maulina[6]
[1,2,3,4,5,6] Department of Informatics, Universitas Amikom Yogyakarta, Yogyakarta, 55283, Indonesia

* **Correspondence:** Email: satria.pauwah@student.amikom.ac.id

Abstract: Ornamental fish is one of the fisheries commodities that have high economic value, in the ornamental fish business there are two types of commodities, namely marine and freshwater ornamental fish. One type of freshwater ornamental fish that is popular at this time is koi fish. This fish, which comes from the Cyprinidae family, comes from Japan, and has very attractive patterns and colors and has high and stable economic value. Broadly speaking, koi fish are estimated into 10 categories namely, Kohaku, Sanke, Showa, Bekko, Utsurimono, Asagi, Shusui, Tancho, Hikari and Koromo. Of the many types and categories of Koi fish that exist, there are difficulties for koi fish enthusiasts who are beginners in knowing the price of Koi fish. By using existing data such as the type of fish, fish length, fish age, and local or imported categories, the price of koi fish can be predicted and is expected to help people make decisions in choosing which koi fish to buy. Based on the results of the prediction experiment using the K-Nearest Neighbor algorithm, an accuracy of 67.99% was obtained and the MSE test was 745271550169.4785, MAE was 486469.26012731483, was 863291.1155395256, and R2 was around 68% with a total dataset of 640 records, and the value of k (closest neighbor) produces the highest accuracy i.e. at k=9.

Keywords: Estimation Price, k-Nearest Neighbor Regression, Fish Koi, Manhattan

1. Introduction

Cultivating freshwater ornamental fish turns out to be able to provide income for many people who do it. Apart from liking the beauty of ornamental fish, many people also depend on cultivating and marketing ornamental fish of various types. There are also a number of cultivators who were originally engaged in the cultivation of consumption fish such as tilapia, gourami, and so on, but are now turning to cultivating ornamental fish. All of this is done because the business opportunities and economic potential of ornamental fish farming are more tempting than consumption fish. Seeing this prospect, the maintenance of ornamental fish, which was originally only occupied by hobbyists, is now also a livelihood for many fish farmers, although it is only done on narrow land and with a limited amount of water [1].

Koi fish are in great demand because of the beauty of the color patterns that form on their bodies, and even koi fish are believed to bring good luck to koi lovers in Indonesia. Koi ornamental fish rearing

in Tulungagung City, East Java buys seeds from several types of koi such as kohaku, taisho, sanshoku, showa, shiro, utsuri, shusui, asagi, goromo, goshiki, bekko, tancho, kinginrin, and kawarimono with seed sizes of 5- 10 cm and feed taken from koi adifa [10]. Koi fish is a type of freshwater ornamental fish with high economic value, both in the national and international markets, so many fish enthusiasts in Indonesia are interested in keeping this fish. However, because of the many categories of koi fish, koi fans, especially those who are new, will find it difficult to know the fair prices of various types of koi fish.

The KNN algorithm calculates the predicted variable using the observed value that corresponds to the nearest K configuration predictor in the training data [11]. In this research, the K-Nearest Neighbor algorithm was chosen , used for price prediction. Based on the above problems, the authors are interested in building a system using the K-Nearest Neighbor method for modeling and predicting koi fish prices using data on koi fish species, size, age, and koi fish specifications. Data mining or data mining is an activity that includes collecting, using historical data to find regularities, patterns and relationships in large datasets. In this research implementation

Data mining algorithm uses the K-Nearest Neighbor algorithm, this algorithm can be used for price prediction. Based on the above problems, the authors are interested in building a system using the K-Nearest Neighbor method for modeling and estimating koi fish prices using data on koi fish species, size, age, number of patterns, dominant patterns and specifications of koi fish.

2. Methodology and Materials

2.1 Data Mining

Data mining is a relatively new way of extracting knowledge from very large amounts of data. Mining involves using and processing accessible data to make useful judgments or conclusions [14].

2.2 Machine learning

Machine learning (ML), as a sub-field of artificial intelligence, is optimally positioned as a predictive approach capable of considering a large number of driving variables and complex interactions between variables. Such ML models can learn from data, predict, and generalize without being explicitly programmed to do so [15].

2.3 Koi fish

Koi fish is a type of ornamental fish that has beautiful and varied colors. Koi fish body color often has several distinctive texture patterns or patterns [13]. Koi fish is a freshwater ornamental fish originating from Japan. Koi fish began to be bred in Japan in the 17th century under the name Nishikigoi which means fish of various colors. The beauty of koi fish lies in its back which has a unique color and pattern and has approximately 100 kinds of Kuroki and Tamadachi color types [4]. While entering Indonesia, it is estimated that in 1981-1982 it was brought by Hany Moniaga who lived in Cipanas, Cianjur, West Java. He then developed a koi farm named Leon and Leonny. The first Koi were 90-100 cm long, 50-57 years old. Since then, koi have become popular in Indonesia and have recently become a hobby game to this day.

2.4 Proposed Method

In figure 1, this method is designed to make it easier to carry out data collection and analysis, the research method starts from literature studies, namely looking for references related to research, such as datasets from koi fish, and references to the algorithms used to classify the prices of koi fish . In addition, it also looks at factors that need to be considered in system design, such as determining algorithms, determining

distance measurements, determining training data, determining data testing, and system design.

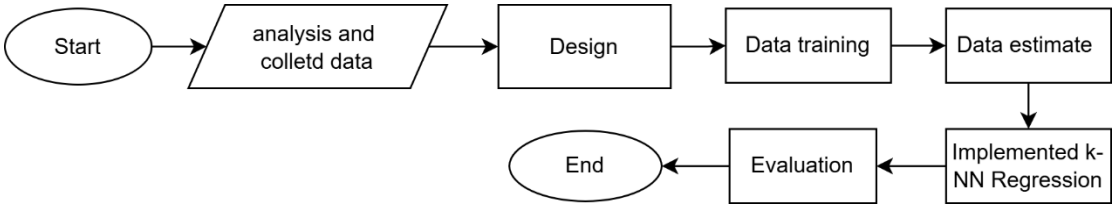

Figure 1. The K-NN Algorithm Model

2.5 Datasets

Figure 2 explains that the research data was obtained from one of the public dataset providers [5], where data from each fish were obtained as many as 10 types of Koi fish, with various forms. Among them are Goromo Fish, Kohaku, Sanke, Tancho, Shiro, Showa, Utsuri, Shusui, Chagoi and Platinum. around 640 dataset records are used, of which 640 datasets are divided into 2 parts, as much as 60% is used for test data, and 40% for training data and data preprocessing is carried out where the data preprocessing stage includes the process of cleaning data by correcting existing errors, eliminating data duplication, and checking inconsistent data [6]. The following is the dataset used.

	Jenis Ikan	Ukuran	Umur	Jumlah Corak	Corak Dominan	Spesifikasi	Harga
0	4	34	1	1	0	0	220
1	4	31	1	1	0	1	500
2	6	40	1	3	1	0	700
3	6	47	2	2	2	0	1.000.000
4	6	33	1	3	1	0	200
...	...	...	...	...	...	...	...
635	1	15	0	2	7	0	16
636	6	22	1	3	1	1	120
637	2	32	1	2	3	0	180
638	6	35	1	3	1	1	700
639	1	49	2	2	7	0	1.400.000

640 rows × 7 columns

Figure 2. Koi Fish Dataset

2.6. *KNN Model*

K -Nearest Neighbor is a supervised learning classification algorithm that aims to classify new objects based on the attributes or characteristics of existing training data samples in the system, based on the closest neighbor distance from test data to training data [3]. The following are the steps for calculating the K-NN algorithm:

3. Determine the value of K or the nearest neighbor from the training data to the test data.
4. Calculate the distance with the euclidean distance of each object to the given training data.
5. Then sort the objects from smallest to largest.
6. Groups a predefined number of K data.
7. Choose the category with the most majority or the one that appears the most.

The following is a formula from KNN [9]. Where i is Data Variable, d is distance and p is data dimension. Where d is distance and p is dimension data.

$$d_i = \sqrt{\sum_{i=1}^{p} (x_{2i} - x_{1i})^2} \quad (1)$$

The following is a calculation of the distance to Manhattan [12], d is distance between x and y, x cluster center data and y is data on attribute.

$$Dist_{MH}\,(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{2}$$

In Figure 3, the K-Nearest Neighbor Regressor will be used as an algorithm that estimates price tags, the k value used is k = 9. The distance between classes will be calculated using the Manhattan equation.

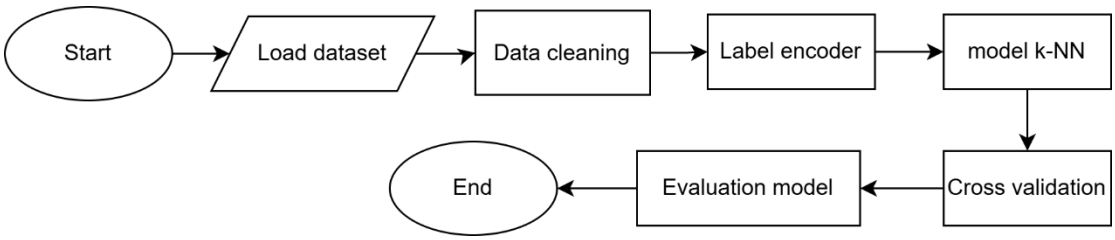

Figure 3. The K-NN Algorithm Model

2.7. *Testing Models*

K-fold Cross-validation is the most frequently used parameter setting method for traditional machine learning models. The whole KNN process is using test data to find the best K value and training data to find the nearest neighbors [2]. During the process, one of the partitions is selected to be tested, the rest are used as training data. Figure 4 explains the testing of the process of dividing data into training data and test data using the train_test_split function from the scikit-learn library. Data sharing is done by determining the proportion of test data as 60% of the total data, while the rest becomes training data. Then, the K Neighbors Regressor model is created by determining the number of nearest neighbors as much as 9, using uniform weights, using the Manhattan matrix, and p equal to 2. The model is then fit with the training data that has been divided previously. Next, predictions are made on the training data using the model that has been fit earlier and the accuracy of the model is calculated for the training data. Then, cross validation is carried out by dividing the training data into 10 parts, then carrying out the training and testing process using 9 parts of the data as training data and 1 part as test data, 10 iterations are carried out using the previously fit model. The results of the cross validation will provide a value for the accuracy of the model in each iteration.

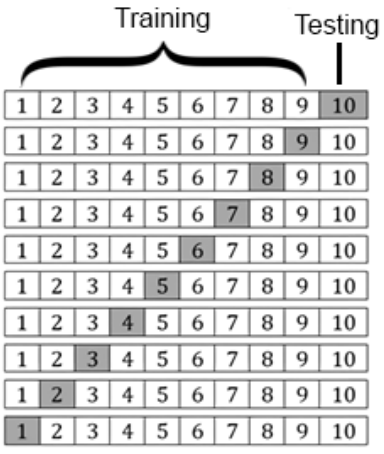

Figure 4. Illustration of K-Fold Cross Validation

3.4. Error

This study aims to determine the critical factors that must be considered in the selection of appropriate performance metrics for regression analysis, by comparing two measures that are often used, namely the coefficient of determination and absolute symmetric mean percentage error.[8].

Coefficient of determination (R^2 or *R*-squared)

$$R^2 = 1 - \frac{\sum_{i=1}^{m} (X_i - Y_i)^2}{\sum_{i=1}^{m} (\bar{Y} - Y_i)^2} \tag{3}$$

(worst value = −∞; best value = +1). The coefficient of determination (Wright, 1921) can be interpreted as the proportion of the variance of the dependent variable which can be predicted from the independent variable [8].

Mean square error (MSE) (4)

$$MSE = \frac{1}{m} \sum_{i=1}^{m} (X_i - Y_i)^2$$

(best value = 0; worst value = +∞). MSE can be used if there are outliers that need to be detected. In fact, MSE is very good at giving greater weight to such points, thanks to L2 norms: obviously, if the model ends up issuing one very bad prediction, the squared part of the function magnifies the error. Since R2 ¼ 1 MSE MST and because MST is fixed for the given data, R2 is monotonically related to MSE (negative monotonic relationship), which means that the order of the regression model based on R2 will exactly match (albeit in reverse order) to the order of the model based on MSE or RMSE [8].

Root mean square error (RMSE) (5)

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (X_i - Y_i)^2}$$

(best value = 0; worst value = +∞)

The two quantities MSE and RMSE are monotonically related (via square roots). The derivation of the regression model based on MSE will be the same as the derivation of the model based on RMSE [8].

Mean absolute error (MAE) (6)

$$MAE = \frac{1}{m} \sum_{i=1}^{m} |X_i - Y_i|$$

(best value = 0; worst value = +∞) . MAE can be used if the outlier represents a damaged part of the data. As a matter of fact, MAE doesn't scold outliers much on training (the L1 norm slightly smooths out any errors from possible outliers), thus providing a general and constrained measure of performance for the model. On the other hand, if the test set also has many outliers, the model performance will be mediocre [8].

3. Result And Discussion

Figure 5 using 640 data, the dataset is divided by 60% for random training data and 40% for random test data. Before performing the estimation, the data will be encoded in the column table which contains strings to integer form. After Encoder.

```
encoder = LabelEncoder()undefined
```

```
ikan ['Jenis Ikan'] = ikan['Jenis Ikan'].map({'Goromo':0, 'Kohaku':1, 'Sanke':2, 'Tancho':3, 'Shiro':4,
ikan ['Umur'] = ikan['Umur'].map({'Dibawah 1 tahun':0, '1 - 3 Tahun':1, 'Diatas 3 Tahun':2})
ikan ['Corak Dominan'] = ikan['Corak Dominan'].map({'Emas / Coklat':0,'Garis Hitam Di Atas Badan':1, 'N
ikan ['Spesifikasi'] = ikan['Spesifikasi'].map({'Local':0,'Import':1})
```

```
ikan['Jenis Ikan'] = encoder.fit_transform(ikan['Jenis Ikan'])
ikan['Umur'] = encoder.fit_transform(ikan['Umur'])undefined
ikan['Corak Dominan'] = encoder.fit_transform(ikan['Corak Dominan'])
ikan['Spesifikasi'] = encoder.fit_transform(ikan['Spesifikasi'])
```

```
ikan.head(150)undefined
```

Figure 5. Encoder Label Code Snippet

Figure 6 is the process of dividing training data and test data on the K-Nearest Neighbors Regressor model. This data division is done using the train_test_split function from the scikit-learn library with the following parameters, x and y are the data that will be separated into training data and test data. x_ikan, x_train, y_ikan, y_train are variable names that will be used to store training data and test data. test_size = 0.60 indicates that the test data will be 60% of all data, the remaining 40% will be training data. random_state = 42 is a number that will be used as a seed for random distribution of data. This seed is used so that the results of random distribution of data each time the program is run do not change. After the data sharing is complete, the KNN Regression model will be created using the Neighbors Regressor function and the following parameters: n_neighbors = 9 indicates that the model will use the 9 closest data to make predictions. weights = 'uniform' indicates that each closest data will give the same weight in the prediction process. Metric = 'manhattan' indicates that the model will use Manhattan distance to find the closest data. p = 2 indicates that the model will use the norm p = 2 in the search process for the closest data. After the model is created, the model will be trained using training data that has been previously divided using the fit function. Next, the model will make predictions on the test data that has been divided before using the predict function. The model accuracy will then be calculated using the score function by entering the test data as a parameter. The model accuracy results will be printed using the print function.

```
x_ikan, x_train, y_ikan, y_train = train_test_split(x,yr, test_size = 0.60, random_state = 42)
model = KNeighborsRegressor(n_neighbors = 9,  weights = 'uniform', metric = 'manhattan', p = 2)
model.fit(x_ikan, y_ikan)
pred_train = model.predict(x_train)
accuracy = model.score(x_train, y_train)
cross_validation = cross_val_score(estimator=model, X=x_train, y=y_train, cv=10)
print('Accuracy :', accuracy)
print()
print(cross_validation)
```

```
Accuracy : 0.6799235997632295

[0.59093461 0.39995216 0.64778306 0.58389998 0.60695725 0.56223255
 0.70678264 0.54295513 0.68154639 0.68565766]
```

Figure 6. KNN Model Results

From the results of the accuracy of Figure 7 of the KNN regressor model below, it can be concluded that this model has an accuracy rate of 67.99%. This level of accuracy can be said to be quite good, but

there is still the possibility of errors in the predictions made by the model. This can happen because the KNN regressor model has a weakness in handling data that is not structured properly, so it can cause errors in predictions. Therefore, further evaluation is needed to improve the accuracy of the model.

```
x_ikan, x_train, y_ikan, y_train = train_test_split(x,yr, test_size = 0.60, random_state = 42)
model = KNeighborsRegressor(n_neighbors = 9,  weights = 'uniform', metric = 'manhattan', p = 2)
model.fit(x_ikan, y_ikan)
pred_train = model.predict(x_train)
accuracy = model.score(x_train, y_train)
cross_validation = cross_val_score(estimator=model, X=x_train, y=y_train, cv=10)
print('Accuracy :', accuracy)
print()
print(cross_validation)
```

```
Accuracy : 0.6799235997632295

[0.59093461 0.39995216 0.64778306 0.58389998 0.60695725 0.56223255
 0.70678264 0.54295513 0.68154639 0.68565766]
```

Figure 7. Accuracy of KNN Regressor

Figure 8 explains that MSE (Mean Squared Error) is a metric used to measure how much error occurs in the model. A large MSE value indicates that the error in the model is also large, conversely if the MSE value is small then the error in the model is also small. Based on the MSE results above, a value of 745271550169.4785 is obtained. It can be concluded that the error in the model is quite large. This can be caused by several things, such as for example the model used does not match the existing data or there is also no proper engineering feature. Therefore, it is necessary to make improvements to the model in order to reduce the errors that occur.

Figure 9 explains that the result of the mean absolute error in the syntax above is 486469.26012731483. This means that the average absolute error of predictions made on cross validation data is 486469.26012731483. The smaller the mean_absolute_error value, the better the prediction results will be.

```
hasil_MSE = mean_squared_error(y_train, pred_train
print('Hasil MSE adalah :')
print(hasil_MSE)
```

```
Hasil MSE adalah :
745271550169.4785
```

Figure 8. MSE Result

```
hasil_MAE = mean_absolute_error(y_train,
print('Hasil MAE adalah :')
print(hasil_MAE)
```

```
Hasil MAE adalah :
486469.26012731483
```

Figure 9. MAE Result

The RMSE resultson Figure 10 are that the average error of the model used is 863291.1155395256. This means that if this model is used to predict a value, then the resulting value will be approximately 863291.1155395256 adrift from the actual value. This indicates that this model is not very good at predicting a value, and may need to be further developed or replaced with a more accurate model.

The results of R2 on Figure 11 are that the model used is able to explain about 68% of the dependent variable (y_train) that is targeted. The explanation for this result is that the value of R2 ranges from 0 to 1, where a value of 1 indicates that the model is able to explain all the dependent variables perfectly, while a value of 0 indicates that the model is not able to explain the dependent variable at all. Therefore,

the R2 results obtained in this case indicate that the model used is quite good at explaining the dependent variable, but there are still around 32% of the dependent variable that cannot be explained by the model.

```
hasil_RMSE = np.sqrt(hasil_MSE)
print('Hasil RMSE adalah :')
print(hasil_RMSE)
```

```
Hasil RMSE adalah :
863291.1155395256
```

Figure 10. RMSE Result

```
hasil_R2 = r2_score(y_train, pred_train)
print('Hasil R2 adalah :')
print(hasil_R2)
```

```
Hasil R2 adalah :
0.6799235997632295
```

Figure 11. R2 Result

4. Conclusion

Based on the research conducted, it can be concluded that the application of the K-Nearest Neighbors (KNN) algorithm in predicting Koi fish prices achieved an accuracy (R2) of 67.99%. While the model is capable of explaining approximately 68% of the variance in the dependent variable, the prediction error remains significant, as evidenced by a Mean Absolute Error (MAE) of approximately 486,469 and a Root Mean Square Error (RMSE) of 863,291. This suggests that the current model is not yet fully optimized, likely due to data noise or suboptimal feature engineering. Consequently, to enhance model performance, future research should prioritize rigorous data preprocessing, including outlier removal, missing value imputation, and proper scaling or normalization to prevent distance distortion. Furthermore, it is recommended to explore alternative architectures such as Neural Networks or Decision Trees, implement ensemble methods like Bagging or Boosting, and apply hyperparameter tuning to identify the optimal configuration for the KNN regressor.

5. Reference

[1]. Lesmana, DS and Dermawan, I, Popular Freshwater Ornamental Fish Cultivation, Independent Spreaders, Jakarta, 2001.

[2]. Bah Ibrahima, Xue Yu, “K-NN Algorithm Used for Heart Attack Detection”, 2021.

[3]. W. Yustanti, "K-Nearest Neighbor Algorithm for Predicting Land Selling Prices," J. Mat. Stats. and Computing, vol. 9, no. 1, pp.57-68, 2012.

[4]. Utomo, NBP, "The Effect of Spirulina Platensis Addition with Different Levels of Feed on the Level of Red Intensity in Kohaku Koi Fish, Journal of Indonesian Aquaculture, Vol. 5, 1-4, 2006.

[5]. Alif Apris Setiawan, "Multiclass SVM with Grid Parameter Optimizer for Predicting Student Performance", Vol.3, No1,36-31,2022.

[6]. Naisah Marito Putry, Betha Nurina Sari. M.Kom, "Comparison of KNN and Naive Bayes Algorithms for Classification of Diagnosis of Diabetes Mellitus", Vol.10, No.1, 2022.

[7]. Jaime Gonzalez-Pardo, Sandra Ceballos-Santos, Rodrigo Manzanas, Miguel Santibanez, Ignacio Fernadez-Olmo, 2022.

[8]. Davide Chicco, Matthijs J, Warrens, Giuseppe Jurman, "The Coefficient of Determination R-Squared is more Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation", 2021.

[9]. Didik Remaldi, Deden Wahiddin, Yana Cahyana, "Identification of Tilapia Freshness based on Gill Color Using the K-Nearest Neighbor Algorithm (K-NN)", Scientific Student Journal for Information, Technology and science, Vol. II,No.1, July 2021

[10]. Muhammad Hilal Aldimas, Nina Aini Mahbubah, Efta Dhartkasari, "Risk Mitigation of the Supply Chain of Ornamental Koi Fish Using the House Of Risk Method", Journal of Civilization Science, Engineering and Technology Vol.9, No.1, June 2021, Page 53 – 65

[11]. Fatkhuroji, Stefanus Aantosa, Richard Anggi Pramunendar. "Prediction of Local Soybean and Imported Soybean Prices Using the Support Vector Machine Method Based on Forward Selection", Journal of Information Technology, Vol.15, No.1, 2019.

[12]. Yumin Liang, Yiqun Pan, Xiaolei Yuan, Wenqi Jia, Zhizhong Huang, "Surogate Modeling for Long-term and High-Resolution Prediction of Building Thermal Load with a Metric-Optimized KNN Algorithm", 2022.

[13]. RA Pramunendar, DPP Prabowo, F. Alzami, RA Megantara, "Application of Random Forest for Fish Species Recognition based on Image Improvement of Clahe and Dark Channel Pior", Upgris Journal of Informatics Vol.7, No. June 1, 2021.

[14]. Rashi Rastogi, Mamta Bansal, "Diabetes Prediction Model Using Data Mining Techniques" 2022.

[15]. Jiaho You, Dan Tulpan, Mark C. Malpass, Jennifer L. Elilis, "Using Machine Learning Regression Models to Predict the Pellet Quality of Pelleted Feeds", Vol. 293, 2022.