UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

October 19, 2023

Author:*Lic. Ijtsi Dzaya Ramos Morales*

Machine Learning , Dr. Marco Antonio Aceves Fernández

Report 3: Clustering Algorithms

Abstract

This study explores the application of the K-nearest neighbors (KNN) algorithm to analyze the Titanic dataset, addressing its complexities, including missing data, categorical attributes, and numerical variations. Through data preprocessing and selective attribute identification, the dataset was prepared for analysis. The Synthetic Minority Over-sampling Technique (SMOTE) was employed to balance the dataset effectively. Both the customized and sklearn library KNN algorithms that were implemented yielded similar high accuracy results, which was an accuracy of 85%, affirming KNN's efficacy in this context.

Introduction

Clustering and classification algorithms serve as indispensable tools for uncovering latent patterns within datasets. The focus of this task turns to data analysis as KNN is employed in the context of the Titanic dataset.

The dataset itself presents certain complexities, including missing data, categorical variables, and variations in numerical scales. These nuances serve as our guiding framework. The primary goal is to demonstrate the practical use of KNN, a lazy learning approach, within this dataset, revealing its efficacy and potential contributions to the field of data analysis.

Theoretical Foundation

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm utilized for classification and regression tasks. It relies on a simple yet powerful principle: data points that are close to each other in feature space are likely to have similar target values. KNN is categorized as a lazy learning algorithm, which means it doesn't build an explicit model during training but rather memorizes the training dataset for subsequent predictions.

For classification tasks, KNN works as follows:

Given a new data point to be classified, KNN identifies the K-nearest data points from the training dataset using a distance metric, often the Euclidean distance formula:

$$\text{Euclidean Distance} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Here, x_i and y_i represent the feature values of the data points being compared, and n is the number of features.

The algorithm then assigns the class label that is most common among the K-nearest neighbors to the new data point. The value of K is a crucial parameter that affects the algorithm's performance, with small values leading to more flexible (noisy) decision boundaries and large values resulting in smoother (but possibly biased) decision boundaries.

KNN's simplicity, effectiveness, and flexibility make it an attractive choice for various classification tasks, including the analysis of the Titanic dataset.

Chi-Square Test for Correlation

The chi-square test is a statistical method employed to determine the independence or association between categorical variables. In the context of data analysis, it is often used to assess the correlation between categorical attributes and the target variable.

For our Titanic dataset, the chi-square test can be utilized to gauge the relationship between categorical attributes (e.g., 'Pclass' or 'Sex') and the binary target variable, 'Survived.' The chi-square statistic is computed as:

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

Here, O represents the observed frequency, E is the expected frequency, and the summation is performed across all categories of the categorical variable.

A high chi-square statistic and a low associated p-value indicate a significant association between the variables, suggesting that the categorical attribute is relevant for classification tasks like predicting passenger survival.

SMOTE for Data Balancing

In many datasets, class imbalances can be a challenge, as one class may significantly outnumber the other. The Synthetic Minority Over-sampling Technique (SMOTE) is a method used to address class imbalances in datasets, particularly beneficial for classification problems.

SMOTE generates synthetic instances of the minority class by interpolating between existing minority class instances. The technique operates as follows:

- For each minority class instance, SMOTE selects its K-nearest neighbors using the Euclidean distance metric.
- A random neighbor is chosen among the K-nearest neighbors, and a synthetic instance is created by combining attributes of the selected instance and the original instance.
- The process is repeated until the desired balance between the classes is achieved.

SMOTE effectively mitigates class imbalances, enhancing the performance of classification algorithms like KNN when applied to datasets with unequal class distribution.

Materials and methods

1. **Dataset Information:**

   The dataset used comprises a version of the Titanic dataset. It includes a total of 12 columns:

   - `PassengerId` (integer): A unique identifier for each passenger.
   - `Survived` (integer): Binary variable indicating survival (0 for not survived, 1 for survived).
   - `Pclass` (integer): Passenger class (1st, 2nd, or 3rd).
   - `Name` (string): Passenger's name.
   - `Sex` (string): Gender of the passenger.
   - `Age` (float): Age of the passenger.
   - `SibSp` (integer): Number of siblings or spouses aboard.
   - `Parch` (string): Number of parents or children aboard.
   - `Ticket` (string): Ticket number.
   - `Fare` (string): Passenger fare.
   - `Cabin` (string): Cabin number.
   - `Embarked` (string): Port of embarkation (C, Q, S).

   It is essential to note that this dataset can be obtained from Kaggle, however, only a reduced version with fewer instances compared to the original Titanic dataset is available.

2. **Software:**

   Various Python libraries were employed for data analysis and classification, including:

   - `numpy`: A library for numerical computations.
   - `pandas`: A data manipulation library for handling tabular data.
   - `sklearn`: The scikit-learn library, featuring machine learning algorithms and tools.
   - `matplotlib`: A data visualization library for creating plots and charts.
   - `seaborn`: A data visualization library for creating informative and attractive statistical graphics.
   - `imblearn`: A library for addressing class imbalance, featuring SMOTE for oversampling.

   Custom machine learning functions were also used, and the dataset was imported from a separate file named `ML_functions`.

3. **Methods:**

   The methods and techniques employed in this project are as follows:

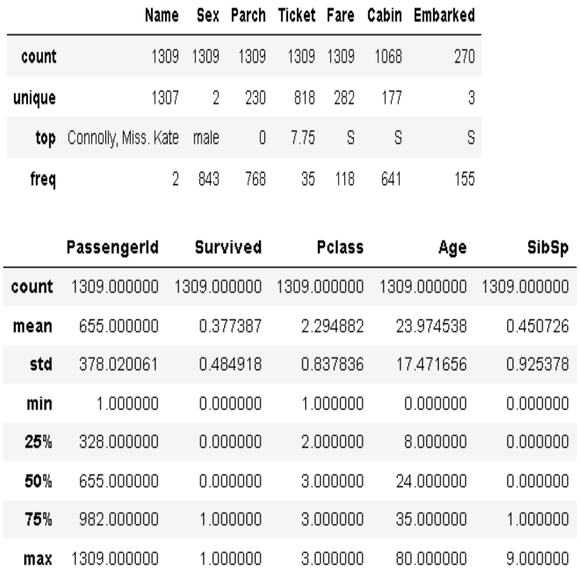
   - **K-Nearest Neighbors (KNN) Implementation:** A KNN algorithm was developed from scratch to classify passengers into survived or not survived categories.
   - **KNN from scikit-learn:** The scikit-learn library was used to implement a KNN classifier.
   - **Elbow Method:** The elbow method was applied to determine the optimal number of clusters (K) for KNN.
   - **SMOTE for Data Balancing:** The Synthetic Minority Over-sampling Technique (SMOTE) was used to address class imbalance in the dataset.
   - **Min-Max Normalization:** Data normalization was applied to scale numerical attributes.

- **One-Hot Encoding:** Categorical values were transformed into binary vectors for use in machine learning models.
- **Data Imputation:** Missing values in categorical attributes were imputed using the mode value.
- **Chi-Square Test:** The chi-square test was utilized to assess the relevance of variables for classification tasks and determine which variables should be dropped from the dataset.

Results

The results presented herein shed light on various aspects of the analysis.

Figure 1 shows the description of the numerical and categorical values, including descriptive statistics.

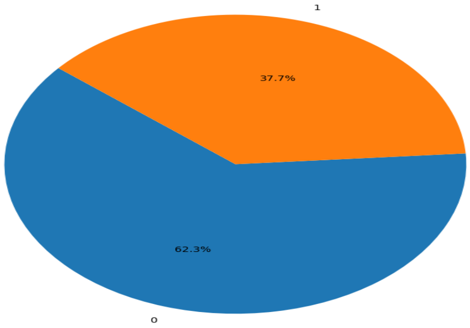


	Name	Sex	Parch	Ticket	Fare	Cabin	Embarked
count	1309	1309	1309	1309	1309	1068	270
unique	1307	2	230	818	282	177	3
top	Connolly, Miss. Kate	male	0	7.75	S	S	S
freq	2	843	768	35	118	641	155

	PassengerId	Survived	Pclass	Age	SibSp
count	1309.000000	1309.000000	1309.000000	1309.000000	1309.000000
mean	655.000000	0.377387	2.294882	23.974538	0.450726
std	378.020061	0.484918	0.837836	17.471656	0.925378
min	1.000000	0.000000	1.000000	0.000000	0.000000
25%	328.000000	0.000000	2.000000	8.000000	0.000000
50%	655.000000	0.000000	3.000000	24.000000	0.000000
75%	982.000000	1.000000	3.000000	35.000000	1.000000
max	1309.000000	1.000000	3.000000	80.000000	9.000000

Figure 1: Dataset description

Figure 2 presents a pie chart for the distribution of the instances according to the target attribute for the before and after the data augmentation via SMOTE.



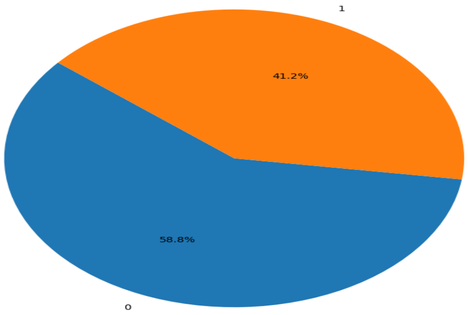
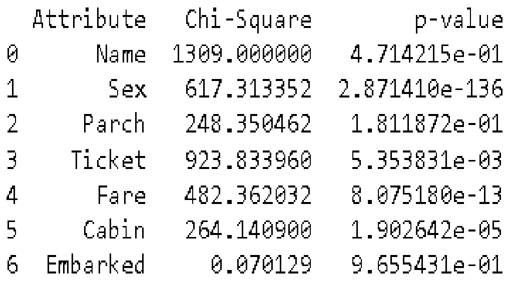


Figure 2: Distribution before and after SMOTE

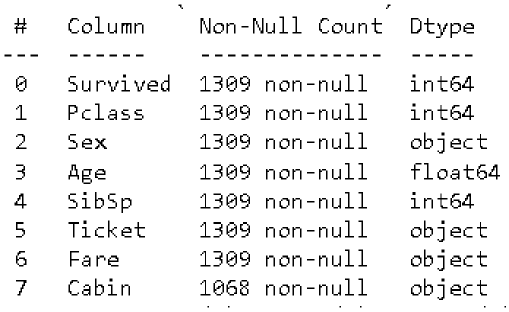
Figure 3 displays results of applying the chi-square test to the categorical attributes and the target attribute.



```
  Attribute  Chi-Square        p-value
0      Name 1309.000000  4.714215e-01
1       Sex  617.313352 2.871410e-136
2     Parch  248.350462  1.811872e-01
3    Ticket  923.833960  5.353831e-03
4      Fare  482.362032  8.075180e-13
5     Cabin  264.140900  1.902642e-05
6  Embarked    0.070129  9.655431e-01
```

Figure 3: Chi-quare results

Figure 4 shows the null-value count and the data type of the attributes.



```
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  1309 non-null   int64
 1   Pclass    1309 non-null   int64
 2   Sex       1309 non-null   object
 3   Age       1309 non-null   float64
 4   SibSp     1309 non-null   int64
 5   Ticket    1309 non-null   object
 6   Fare      1309 non-null   object
 7   Cabin     1068 non-null   object
```

Figure 4: Elbow method results for Clarans

Figure 5 displays the elbow method and the confusion matrix of the KNN customized method and the confusion matrix of the sklearn KNN method, respectively.

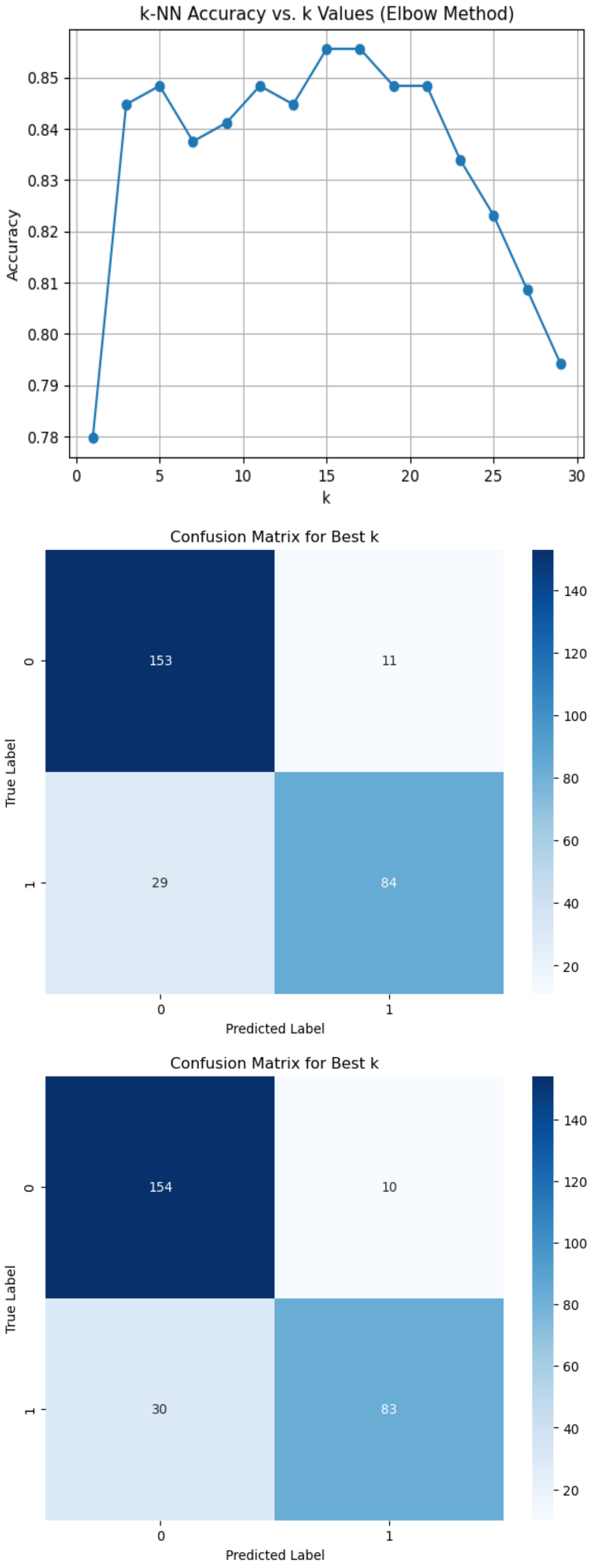
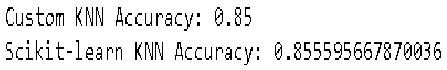


Figure 5: Results of KNN classification

Figure 6 depicts the results of the comparison between both methods.



```
Custom KNN Accuracy: 0.85
Scikit-learn KNN Accuracy: 0.855595667870036
```

Figure 6: Results of KNN comparison

Discussion

Dataset

As depicted in Figure 1, the dataset comprises both categorical and numerical attributes, forming a diverse information set. Notably, missing values were observed in the ”Cabin” and ”Embarked” attributes. To assess the significance of attributes in relation to the target variable, a chi-square test was conducted, with the results illustrated in Figure 3.
The results of the chi-square test indicate that attributes such as ”Sex,” ”Fare,” and ”Ticket” exhibit substantial associations with the target variable. ”Cabin” also displays a notable association, though its strength may be slightly less pronounced compared to ”Sex,” ”Fare,” and ”Ticket.” Conversely, attributes including ”Name,” ”Parch,” and ”Embarked” do not demonstrate a substantial association with the target variable, leading to the decision to exclude them from the dataset.
Notably, ”Embarked,” although seemingly significant by its definition and the task at hand, presented a unique challenge due to its high percentage of missing data, approximately 80%. Imputing missing values might introduce bias, rendering it unsuitable for analysis and leading to its omission from the dataset.
Imputation for the ”Cabin” attribute, a categorical variable, was carried out using mode imputation to ensure data completeness.
Subsequently, the dataset was balanced using the Synthetic Minority Over-sampling Technique (SMOTE), resulting in a class distribution ratio of 58.8% and 41.2%. This balanced distribution, although not a precise 50/50 ratio, minimizes the introduction of excessive augmented data points while achieving a significant enhancement in dataset balance, as depicted in Figure 2.
Finally, the preprocessed attributes underwent min-max normalization and one-hot encoding for further analysis.

KNN

Results from both the customized and the sklearn K-nearest neighbors (KNN) algorithms showcased a striking similarity, differing only in decimal points in accuracy. It’s noteworthy that the customized algorithm exhibited a longer computation time. This discrepancy in execution time is attributed to its design prioritizing functionality over efficiency.
The optimal value for the hyperparameter ”k” was determined using the elbow method, as illustrated in Figure 5, and was found to be 15. This choice for ”k” aligns with a balance between clustering complexity and quality.
Figure 6 presents the accuracy results obtained from both the customized and sklearn algorithms. These accuracies closely correspond to the expected values reported in the literature when utilizing KNN for this specific task.

The similarities in results between the two algorithms affirm the robustness and reliability of the KNN approach in our data analysis.

Conclusion

In this study, the application of the K-nearest neighbors (KNN) algorithm to the Titanic dataset was explored. The dataset presented several challenges, including missing data, categorical attributes, and variations in numerical scales. Through meticulous data preprocessing, including the identification of relevant attributes, imputation, balancing, normalization, and encoding, the dataset was prepared for meaningful analysis.

The KNN implementation, both the customized version and the sklearn library algorithm, yielded similar results in terms of accuracy, which was a final accuracy of 85%. The high accuracy values obtained from both algorithms align with expectations found in the literature, reinforcing the efficacy of KNN in this context.

The analysis of the Titanic dataset demonstrates the practical utility of the K-nearest neighbors algorithm in the area of data analysis.

References

[1] David Aha, Denis Kibler, and Marc K. Albert, "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37-66, 1991.

[2] Maurice G. Kendall and Alan Stuart, "The advanced theory of statistics: Volume 2, Inference and relationship," Charles Griffin & Company, 1979.

[3] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321-357, 2002.

[4] M. A. Aceves Fernández, "Inteligencia Artificial para Programadores con Prisa," 2023.

[5] C. C. Aggarwal and C. K. Reddy, *Data clustering: Algorithms and applications.* Chapman and Hall/CRC, 2014. (Book)