Bir Sınıflandırma Modelinin Oluşturulması ve Değerlendirilmesi
Building and Validating a Classification Model

Çağrı ÇAYCI

***Özetçe* —Bu belge, bir veri madenciliği işleminin problem tanımından başlayarak model değerlendirilmesine kadar olan tüm adımların uygulamalı bir şekilde gösterilmesi amacıyla hazırlanmıştır. Bu adımlar sırasıyla; problem tanımı, veri toplama, veri temizleme, veri birleştirme, model seçimi ve model değerlendirme işlemlerinden oluşmaktadır. Veri madenciliği süreci boyunca, her bir adımın titizlikle uygulanması, sonuçların doğruluğunu ve etkinliğini artırmak için kritiktir.**

***Anahtar Kelimeler*—veri madenciliği, veri toplama, veri temizleme, veri birleştirme, model seçimi, model değerlendirme.**

***Abstract*—This document is prepared with the aim of demonstrating all the steps of a data mining process from problem definition to model evaluation in an applied manner. These steps include problem definition, data collection, data cleaning, data integration, model selection, and model evaluation processes. Throughout the data mining process, meticulous implementation of each step is critical to enhance the accuracy and effectiveness of the results.**

***Keywords*—data mining, data collection, data cleaning, data integration, model selection, model evaluation.**

I. Giriş

In today's world, there's a lot of information online. It's easy to find what you need, but all that data isn't much help unless you can make sense of it. That's where data mining comes in. It's like a tool that helps sort through all the information to find the important stuff. By using special methods, data mining helps uncover useful patterns and insights, turning raw data into helpful knowledge. This helps businesses, researchers, and others make better decisions and come up with new ideas based on what they find in the vast sea of online information.This paper demonstrates the process of extracting information and shows how it can be put to use.

II. Data Mining Steps

A. Problem Definition

In the realm of user decision prediction without historical behavioral data, the objective is to develop a predictive model that anticipates a user's decision solely based on their demographic and contextual features. Leveraging a dataset containing user attributes such as age, gender, location, income level, education, and contextual information such as time of day, device type, and any other relevant contextual features, the aim is to build a model capable of accurately predicting the user's decision.

B. Data Collection

The data used in this data mining process was sourced from the UCI Machine Learning Repository. This dataset contains 25 features and 12685 instances. It was gathered through a survey conducted on Amazon Mechanical Turk. The survey presents participants with different driving scenarios, including factors like destination, current time, weather conditions, presence of passengers, and more. All features are categorical. Following each scenario, participants are asked whether they would accept a coupon if they were the driver in that specific situation. Consequently, the dataset consists of two classes.

0	1	2	3
No Urgent Place	No Urgent Place	No Urgent Place	No Urgent Place
Alone	Friend(s)	Friend(s)	Friend(s)
Sunny	Sunny	Sunny	Sunny
55	80	80	80
2PM	10AM	10AM	2PM
Restaurant(<20)	Coffee House	Carry out & Take away	Coffee House
1d	2h	2h	2h
Female	Female	Female	Female
21	21	21	21
Unmarried partner	Unmarried partner	Unmarried partner	Unmarried partner
1	1	1	1
Some college - no degree	Some college - no degree	Some college - no degree	Some college - no degree
Unemployed	Unemployed	Unemployed	Unemployed
$37500 - 49999	$37500 - 49999	$37500 - 49999	$37500 - 49999
NaN	NaN	NaN	NaN
never	never	never	never
never	never	never	never
NaN	NaN	NaN	NaN
4~8	4~8	4~8	4~8
1~3	1~3	1~3	1~3
1	1	1	1
0	0	1	1
0	0	0	0
0	0	0	0
1	1	1	1
1	0	1	0

Şekil 1: Some data examples. (It is rotated to demonstrate)

C. Data Cleaning

Data cleaning is indeed a crucial step in the data mining process. Missing or duplicated values can significantly impact the accuracy of results. Properly addressing and handling these issues ensures that the data used for analysis is reliable and representative, leading to more accurate and trustworthy outcomes in the data mining process.

1) Handling Identical Data: Handling identical data in a dataset can indeed vary depending on the specific characteristics of the dataset. In the case of this project, where the dataset contains only 77 identical entries, the proportion of identical data relative to the total dataset size is relatively small. Given this context, the most straightforward and efficient approach would be to remove these identical data.

Removing identical data helps in ensuring the integrity and accuracy of the dataset by eliminating redundant information. Since the number of duplicates is small compared to the overall dataset size, removing them is unlikely to significantly impact the analysis or results. Moreover, it simplifies subsequent data processing steps and reduces the risk of bias or distortion in the analysis.

Therefore, in this scenario, removing the 77 identical data is a pragmatic and effective strategy for handling identical data in the dataset.

```
4192, 4236, 4280, 4324, 4409, 4475, 4498, 4586, 4739, 4761, 4805,
4827, 4849, 4871, 4959, 4980, 5002, 5058, 5102, 5124, 5190, 5454,
5475, 5497, 5606, 5628, 5672, 5694, 5738, 5782, 5848, 5936, 6001,
6023, 6067, 6089, 6255, 6321, 6343, 6448, 6470, 6492, 6557, 7838,
7841, 7843, 7844, 7845, 7846, 7847, 7848, 7849, 7853, 7854, 7855,
7856, 7857, 7858, 7859, 8495, 8496, 8497, 8499, 8501, 8503, 8506,
8507, 8508, 8509, 8511, 8512, 8513, 8515, 8516
```

Şekil 2: Indices of identical data.

2) Handling Missing Values: There various approaches exist for handling missing values in a dataset, each with its own advantages and considerations. These approaches include:

Removal: Simply removing rows or columns with missing values can be effective if the missing values are few and do not significantly impact the overall dataset.

Imputation: Imputing missing values involves replacing them with estimated or calculated values. This can include replacing missing numerical values with the mean, median, or mode of the respective feature, or using more advanced techniques such as regression imputation or k-nearest neighbors imputation.

In this project, there are some missing values in dataset and they are handled using two distinct approaches: replacing missing values with the mode and employing k-nearest neighbors (KNN) imputation. Both replacing with the mode and k-nearest neighbor algorithms are implemented. The primary rationale behind utilizing the mode is its effectiveness in replacing missing values, particularly in categorical features. Given that the features in the dataset are categorical in nature, replacing missing values with the mode is considered a prudent choice. The mode represents the most frequently occurring value in a feature, making it a sensible strategy for filling in missing values, as it preserves the categorical nature of the data while minimizing the potential impact on the overall distribution of the feature. This approach ensures that the imputed values align with the existing categories in the dataset, thereby maintaining the integrity of the categorical variables.

```
destination              0
passanger                0
weather                  0
temperature              0
time                     0
coupon                   0
expiration               0
gender                   0
age                      0
maritalStatus            0
has_children             0
education                0
occupation               0
income                   0
car                  12576
Bar                    107
CoffeeHouse            217
CarryAway              151
RestaurantLessThan20   130
Restaurant20To50       189
toCoupon_GEQ5min         0
toCoupon_GEQ15min        0
toCoupon_GEQ25min        0
direction_same           0
direction_opp            0
Y                        0
```

Şekil 3: Categories and missing number of value

D. Data Transformation

Data transformation is converting and standardizing the data into a common format or structure. In this project, data transformation is applied through normalization, a process of converting and standardizing data into a common format or structure. Since the dataset comprises categorical features, they are transformed into numerical features to facilitate suitable data mining techniques. The technique is used for this purpose is one-hot encoding.

	destination	passanger	weather	temperature	time	coupon	expiration	gender	age	maritalStatus	...
0	0	0	0	0	0	0	0	0	0	0	...
1	0	1	0	1	1	1	1	0	0	0	...
2	0	1	0	1	1	2	1	0	0	0	...
3	0	1	0	1	0	1	1	0	0	0	...
4	0	1	0	1	0	1	0	0	0	0	...

Şekil 4: One-hot encoding for normalization

E. Data Integration

Data integration is indeed a crucial step in many data analysis projects, involving the combination of data from multiple sources to create a unified and comprehensive dataset. However, in the context of this particular project, where the data is sourced from a single origin, the need for data integration is obviated.

Since the dataset originates from a single source, there is no requirement to integrate data from disparate sources.

Consequently, the data is already in a unified format, and there is no need for additional transformations or merging with other datasets.

F. Data Partition

Data partitioning is indeed a critical step in the data mining process, involving the division of a dataset into subsets for different purposes such as model training and testing. This partitioning is essential for accurately assessing the performance of predictive models.

In this project, the dataset is partitioned into two subsets using a ratio of 0.8. This means that 80% of the dataset is allocated for training purposes, while the remaining 20

By allocating the majority of the dataset (80%) for training, the models can learn from a sufficient amount of data to capture underlying patterns and relationships effectively. This larger training set helps ensure that the models are adequately trained and have a comprehensive understanding of the data.

The testing subset, comprising 20% of the dataset, is then used to evaluate the performance of the trained models. By withholding a portion of the data during training and using it for testing, the models' performance can be assessed on unseen data, providing a more realistic estimate of how well they generalize to new, unseen instances.

Overall, partitioning the dataset into training and testing subsets in a ratio of 0.8:0.2 ensures that the models are trained on a sufficient amount of data while also allowing for robust evaluation of their performance on unseen data.

G. Model Building

In this project, selecting the appropriate model is indeed crucial for effective data mining. In this particular case, two classification algorithms, Naive Bayes Classification and Decision Tree Classification, are utilized, along with the implementation of Hierarchical Clustering.

1) Naive Bayes Classification: Naive Bayes Classification is a probabilistic algorithm based on Bayes' theorem and Chain Rule, commonly used for classification tasks. It assumes that the features are conditionally independent, making it particularly effective for categorical data and relatively simple to implement. Naive Bayes is well-suited for classification tasks with discrete features, making it a suitable choice for this project given the categorical nature of the dataset. Indeed, Naive Bayes classification often yields logical and interpretable results, which is one of its key advantages. Therefore, Naive Bayes Classification is implemented for this project.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{1}$$

Şekil 5: Bayes' theorem

$$P(\text{class}|\text{features}) = \prod_{i=1}^{n} P(\text{feature}_i|\text{class}) \tag{2}$$

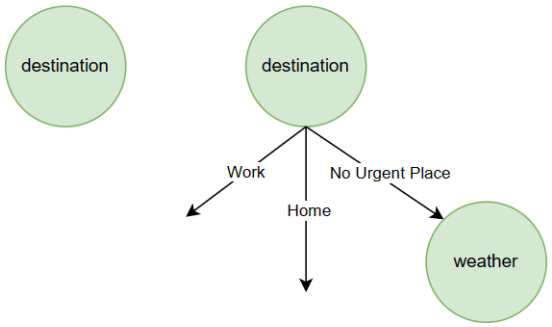
Şekil 6: Chain Rule

2) Decision Tree Classification: The logic of the algorithm revolves around recursively partitioning the feature space based on the features that provide the best split, typically measured by criteria such as the Gini Index or information gain.

The Gini Index, in the context of decision trees, is a measure of impurity or uncertainty within a dataset. It represents the probability of incorrectly classifying a randomly chosen element if it were randomly labeled according to the distribution of classes in the set. Essentially, the Gini Index quantifies the level of statistical dispersion or impurity within a given set of data.

During each iteration of building a decision tree, the algorithm evaluates different features and their potential splits based on the Gini Index. The feature that leads to the greatest reduction in impurity, as measured by the decrease in the Gini Index, is selected as the splitting criterion. This process is repeated recursively for each subset of data until a stopping criterion is met, such as reaching a maximum depth or minimum number of samples in a leaf node.

By selecting the feature that provides the best Gini Index at each iteration, decision trees can efficiently partition the feature space into regions that are increasingly homogeneous with respect to the target variable, leading to effective classification.



Şekil 7: Representative figure of building process of DT

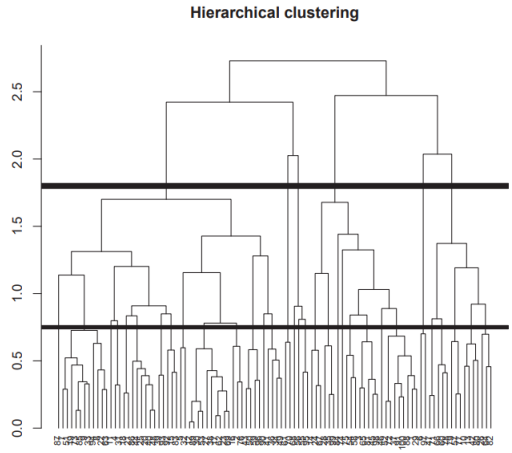
3) Hierarchical Clustering: Although Hierarchical Clustering was implemented in the project, it couldn't be successfully trained and tested. Hierarchical Clustering is a method used to group similar data points into clusters and iteratively merge the closest pair of clusters based on a defined distance metric until there are 2 clusters left.

In this implementation, the distance between two clusters is calculated as the shortest distance between any two points belonging to the two clusters. Specifically, the distance between two points is determined by the number of uncommon categories they possess.

The reason why Hierarchical Clustering could not be trained and tested in the project is due to its computational complexity, particularly its time complexity of O(n$\hat{3}$). This

complexity arises from the need to calculate the distance between all pairs of data points, which becomes increasingly burdensome as the dataset size grows.

Given the large size of the dataset, training the Hierarchical Clustering model would require significant computational resources and time. In some cases, the training process might even become infeasible or impractical due to the prolonged duration it takes to complete.



Şekil 8: Hierarchical Clustering

H. Model Evaluation

In this project, each model is trained and tested with two different datasets, each filled using a different imputation method. One dataset is filled using the K-Nearest Neighbor (KNN) imputation technique, while the other is filled using the mode (most frequent value) imputation method.

By testing each model with both KNN-imputed and mode-imputed datasets, the project aims to evaluate the performance of the models under different imputation strategies. This allows for a comprehensive assessment of how each model responds to variations in the imputation technique and helps in determining the most suitable model-dataset combinations for the given project requirements.

Model	Accuracy	Precision	Recall	F1 Score
Naive Bayes Mode Imbalanced	0.526566	0.545287	0.852624	0.665171
Decision Tree Mode Imbalanced	0.618557	0.648443	0.673616	0.66079
Naive Bayes KNN Imbalanced	0.526566	0.545287	0.852624	0.665171
Decision Tree KNN Imbalanced	0.610626	0.639973	0.672178	0.65568
Naive Bayes Mode Equal	0.837827	0.837827	1	0.911758
Decision Tree Mode Equal	0.312847	0.874016	0.218128	0.338802
Naive Bayes KNN Equal	0.837827	0.837827	1	0.911758
Decision Tree KNN Equal	0.320777	0.873134	0.221486	0.353341

Şekil 9: Test results

To explain what tests scenarios represents: Model Mode Imbalanced: Using Naive Bayes Classification on handled missing values by mode on imbalanced dataset. Model KNN Imbalanced: Using Naive Bayes Classification on handled missing values by using KNN on imbalanced dataset. Model Mode Equal: Using Naive Bayes Classification on handled missing values by mode on uniformly distributed dataset. Model KNN Equal: Using Naive Bayes Classification on handled missing values by using KNN on uniformly distributed dataset.

$$
\begin{aligned}
precision &= \frac{TP}{TP+FP} \\
recall &= \frac{TP}{TP+FN} \\
F1 &= \frac{2 \times precision \times recall}{precision + recall} \\
accuracy &= \frac{TP+TN}{TP+FN+TN+FP}
\end{aligned}
$$

Şekil 10: The formulas to measure models performances

1) Conclusion: The results obtained suggest that Decision Trees perform better in handling imbalanced datasets, as anticipated. Decision Trees are known for their ability to effectively handle imbalanced class distributions. This is because Decision Trees make splits in the data based on feature values without explicitly relying on the class distribution. As a result, they can still accurately predict minority classes, even in imbalanced datasets.

On the other hand, Naive Bayes Classification demonstrates better performance than Decision Trees in uniformly distributed datasets. This may be attributed to the fact that Naive Bayes Classification tends to perform well when there is a balanced distribution of classes in the dataset.

The reason for Naive Bayes outperforming Decision Trees in uniformly distributed datasets could also be due to an insufficient number of negative instances. Since Naive Bayes relies on class conditional independence assumptions, it may provide better performance when there are enough instances for each class to accurately estimate the class conditional probabilities.

Overall, these findings highlight the importance of considering the characteristics of the dataset, such as class distribution, when selecting the appropriate classification algorithm. While Decision Trees excel in handling imbalanced datasets, Naive Bayes may be more suitable for uniformly distributed datasets with balanced class proportions.

III. Appendix

The link of my video: https://youtu.be/FxGqolei3AM

Bilgilendirme

In this project, K-Nearest Neighbor, Naive Bayes Classification, and Hierarchical Clustering models are implemented. However, Decision Tree Classification is only used not implemented.

Kaynaklar

[1] UCI Machine Learning Repository. "in-vehicle coupon recommendation", 2020. [Online]. Available: https://doi.org/10.24432/C5GS4P

[2] Webb, Geoffrey I., Eamonn Keogh, and Risto Miikkulainen. "Naïve Bayes." Encyclopedia of machine learning 15.1 (2010): 713-714.

[3] Kingsford, C., & Salzberg, S. L. (2008). What are decision trees?. Nature biotechnology, 26(9), 1011–1013. https://doi.org/10.1038/nbt0908-1011

[4] Murtagh, Fionn, and Pedro Contreras. "Algorithms for hierarchical clustering: an overview." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2.1 (2012): 86-97.