Smart Road Pricing Strategies for Sustainable Tourism Management in Sierra Nevada National Park: Insights from a Dynamic Field Study

Trabajo Fin de Máster
Curso académico 2023/2024
Máster Universitario en Matemáticas

Trabajo realizado por: Alberto Jesús Durán López
Tutor: Blanca Delgado Márquez
Co-tutor: Miguel Luis Rodríguez González

Universidad de Granada
Escuela Internacional de Posgrado

Declaración de originalidad

D. **Alberto Jesús Durán López**

Declaro explícitamente que el trabajo presentado como Trabajo de Fin de Máster (TFM), correspondiente al curso académico 2023-2024, es original, entendido esto en el sentido de que no he utilizado para la elaboración del trabajo fuentes sin citarlas debidamente.

En Granada a 5 de Julio de 2024.

Fdo: **Alberto Jesús Durán López.**

CONTENTS

SUMMARY

This project focuses on the Sierra Nevada National Park, specifically in the Alpujarra towns of Pampaneira, Capileira, and Bubión. The objective is to develop a dynamic pricing algorithm for parking management in the Barranco de Poqueira area. The Barranco de Poqueira annually attracts a significant number of tourists and nature enthusiasts. While this influx benefits the local economy, it also poses potential threats to the park's ecological balance. For this reason, this project proposes an alternative approach that combines machine learning techniques to develop a dynamic pricing algorithm based on a dataset of tourist visitation data in the area while preserving social responsibility, economic sustainability, and environmental preservation components. Additionally, we incorporate a behavioral component, which is based on the likelihood of future visits if a parking toll is implemented. This data is obtained through questionnaires.

First, due to the limited instances and subjective responses of survey questions, we introduce an oversampling method called KM-SMOTE, which combines K-medoids clustering with the SMOTE (Synthetic Minority Over-sampling Technique) algorithm. This method is designed to enhance the performance of our data, which is affected by noise and high dimensionality. We analyze and clean the data and compare the algorithm to other clustering-based algorithms. With this model, we predict the likelihood of returning to the Barranco de Poqueira after the implementation of a toll in a parking lot. Second, we enrich our data with this new variable to create a dynamic pricing model. The objective is to separate the variables into four categories (environmental, social, economic, and behavioral) and assign weights to the different features using a predefined strategy. Thus, the weights assigned to each feature will be used to assign prices in a price assignment function. Finally, we create a regression model to understand the impact of each feature on the final price, allowing us to validate the pricing algorithm and adjust the weights of the variables if needed.

The goal of this project is to implement our contributions in the Alpujarra region within the Sierra Nevada National Park to dynamically assign toll prices to vehicles. This approach, accompanied by data collection over several months, allows for dynamic price assignments for new vehicles entering the area. This prepares stakeholders to manage and prevent overcrowding in the Barranco de Poqueira, ensuring sustainable tourism management and enhancing the visitor experience. The dynamic pricing algorithm aims to minimize high traffic in sensitive areas, promote the use of eco-friendly transportation, and distribute the economic benefits of tourism more evenly across the region.

CHAPTER 1

INTRODUCTION

1.1 Motivation

In recent years, machine learning has become increasingly important in various areas of society, such as education [1, 2], health [3, 4], economics [5, 6], and tourism [7, 8]. This paradigm allows the use of different algorithms to build useful models for data-driven decision-making. The data collected provides valuable information on behavioral patterns that can be important in predictive analytics [9]. This information has the potential to develop service innovation, management practices, and the competitiveness of all parties involved [10]. For example, deep learning models can be used to build dynamic toll pricing schemes for smart transportation systems based on road behavior [11].

However, in many cases, the data collection process can pose difficulties for scientists who need to train predictive models [12]. Two examples are: First, the proliferation of data collection sensors that accompanies the rise of the Internet of Things (IoT). These devices, while producing a great deal of valuable information in different fields, often contain noise due to error rates and device failures [13]. Second, questionnaires are conditioned by economic and temporal factors [14, 7], limiting the instances collected. The scarcity of instances contrasts with the large number of variables typically collected, resulting in higher-dimensional datasets that pose challenges during model training [15]. Additionally, when dealing with classification problems, class imbalance issues appear, which are compounded by the two previous problems. Several studies have explored solutions to these problems, such as synthetic data generation to increase sample sizes [16, 17] and oversampling algorithms [18, 19], which have shown considerable performance improvements in certain scenarios compared to generative algorithms [20, 21]. However, no research has delved into the application of a class balancing algorithm on noisy and high-dimensional datasets.

In this project, we focus on two different contributions. First, we propose KM-SMOTE, a novel data balancing model based on K-medoids combined with SMOTE algorithm. Second, we develop a smart pricing model based on traffic flow, which assigns a price to each vehicle according to its environmental footprint, socio-economic factors, and behavioral patterns.

Specifically, we test our approach in the area of smart villages. We collected data from License Plate Recognition (LPR) sensors installed on the access road to the area to monitor traffic. These data have been cross referenced with street questionnaires obtained in January, March, and July 2023, which asked about the likelihood of future visits if a parking fee is implemented, as parking is currently free of charge. We perform a study of the proposed model using various oversampling methods and distance metrics.

Our work is useful for scientists working with high-dimensional and noisy data, mainly derived from sensors and questionnaires. We provide a new balancing algorithm that they can apply with a high probability of improvement to their own data. In addition, our results allow us to build a smart pricing model based on traffic flow and classify visitors according to their likelihood to visit in the future if they are charged for parking in the area that is currently free.

1.2 Related work

The problem of unbalanced data is common in a variety of domains [22, 23, 24] and raises issues of model performance and generalization [25]. Various solutions have been proposed in the literature, ranging from synthetic data generation algorithms such as Generative Adversarial Networks (GAN) [21, 26] or Variational AutoEncoders (VAE) [27, 28] to the use of oversampling algorithms such as Adaptive Synthetic Sampling (ADASYN) [29, 30] and SMOTE [19, 31], often in combination with other techniques. However, it has been shown that the effectiveness of each method is highly dependent on the specific problem and the characteristics of the data collected. For example, GAN may show superior performance when dealing with datasets with very small minority classes compared to VAE [32]. In some cases, the combination of VAE and GAN [33] manages to optimize the individual shortcomings of each other [34]. The weakness of GAN compared to SMOTE is the lack of control over the quality of minority class samples [35], as GAN can introduce noise into the generated samples which affects the classification [20]. GAN tends to generate samples in the same direction, making it difficult for classifiers to create a clear decision boundary between major and minor classes, unlike SMOTE [21].

Particularly in the area of sensors or questionnaires data, noise arising from the distribution of data is a major problem [13, 14]. In addition to the challenges posed by the economic and temporal constraints associated with questionnaires [7], and the inherent problem of data imbalance, another critical factor arises. When data is collected through questionnaires, noise is introduced into the dataset. This noise arises because, within the same class, data can exhibit specific behaviors not because they are mislabeled, but due to individual differences in how participants respond. Although all responses are correctly classified, the differences in how questions are interpreted and answered generate a type of noise characteristic of that class, potentially affecting analysis and modeling in machine learning [36]. This is especially common in the healthcare field, where the response to a drug of two patients with similar characteristics may be totally different [37]. This aspect remains relatively unexplored in the literature, some work has investigated oversampling SMOTE methods for noisy datasets [38, 39], but none of those explain the noise in high-dimensional scenarios.

The literature review reveals examples of approaches such as K-means combined with SMOTE [19, 40], which employ Euclidean distance followed by the calculation of the mean to balance the data. Similarly, CURE-SMOTE [41] employs the Clustering Using REpresentatives (CURE) algorithm to first cluster the minority class instances, followed by SMOTE to generate synthetic samples within these clusters. In contrast, the use of K-medoids, which employs the median and is more robust to outliers and noise, is particularly well-suited to scenarios involving combined questionnaires and sensor data. The combination of K-medoids and SMOTE has not been explored so far in the literature in the field of oversampling of unbalanced noisy data. Another element that has also not been explored is the impact of different distance metrics on SMOTE data generation. K-medoids allows incorporating various distance metrics, which is advantageous in cases with high dimensionality, where Euclidean distance may not be optimal [42].

In the field of dynamic pricing, the literature has evolved significantly from early static pricing strategies to advanced, real-time algorithms. Initial approaches relied on fixed pricing models, where prices remained constant over time [43]. With the development of revenue management in the airline industry, dynamic pricing models began to adjust prices based on demand fluctuations, significantly improving profitability [44]. As computational capabilities and data availability have increased, modern dynamic pricing algorithms have integrated machine learning and big data analytics to optimize prices in real-time. For example, Uber uses dynamic pricing algorithms to adjust fees based on current demand and supply conditions [45]. Similarly, Amazon employs machine learning models to dynamically set prices for their products, taking into account factors such as competitor prices, customer behavior, and inventory levels [46]. These advancements demonstrate the effectiveness of dynamic pricing algorithms compared to fixed ones.

1.3 Objectives

The objective of this work is to analyze methods for dynamic pricing in the context of parking management within the Sierra Nevada National Park, specifically targeting the Barranco de Poqueira area. The specific objectives are as follows:

- Review previous works on clustering applied to traffic behavior with License Plate Recognition (LPR) data to understand the existing approaches and identify gaps that our project can address.
- Create an oversampling algorithm to deal with noisy and high-dimensional data based on K-medoids clustering with SMOTE (KM-SMOTE). Study oversampling algorithms, particularly focusing on clustering-based variants.
- Examine, with our smart villages use case, the effectiveness of the proposed KM-SMOTE algorithm in handling noisy and high-dimensional data derived from sensor and survey inputs, and its impact on predicting the likelihood of tourists returning if a parking toll is implemented.
- Evaluate the performance of the proposed algorithm through metrics such as F1-

Score, G-Mean, and Recall, and compare its efficacy against other clustering-based algorithms.

- Design a dynamic pricing algorithm that integrates environmental, socio-economic, and behavioral features to manage parking and reduce congestion in environmentally sensitive areas.
- Develop a regression model to understand the influence of various features on the final assigned price, validating and refining the dynamic pricing algorithm.

By achieving these objectives, this work aims to provide a robust solution for sustainable tourism management that balances environmental preservation, social responsibility, and economic sustainability. The implementation of this model in the Alpujarra region will help manage tourist inflow dynamically, ensuring a positive impact on the Sierra Nevada National Park.

Data Availability

All code developed for this study is available at `https://github.com/thealberteitor/SmartTourism_TFM`.

1.4 Thesis Outline

The remainder of the work is organized as follows. Chapter 2 describes the main mathematical fundamentals that we use in this work. Chapter 3 introduces the proposed oversampling algorithm. Chapter 4 presents and describes the dynamic pricing algorithm. Finally, Chapter 5 concludes with the findings from this project.

CHAPTER 2

FUNDAMENTALS

2.1 Machine Learning Methodologies

In machine learning, there are two key approaches focused on developing algorithms that enable computers to learn from data: supervised and unsupervised learning.

2.1.1 Supervised Learning

In supervised learning, the goal is to generate a function that maps input features to corresponding output labels. This is achieved by training a model on a labeled dataset, where each data point consists of input features and its associated correct label. There are 2 types classification and regresion.

- **Classification**: Consider a dataset with n samples and d features, represented as a matrix X of size $n \times d$, where each row represents a sample and each column a feature:

  $$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{pmatrix}.$$

  Each sample is associated with a label y_i, forming the label vector $\mathbf{y}$:

  $$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix},$$

  where $y_i \in \{c_1, c_2, \dots, c_k\}$, with $c_1, c_2, \dots, c_k$ representing the possible classes, and k being the number of classes in the dataset.

  The goal of a classification model is to generate a mapping function $f : \mathbb{R}^d \to \{c_1, c_2, \dots, c_k\}$ from the feature space to the class labels, where the output space consists of discrete class labels. For a new input $\mathbf{x}$, the function $f(\mathbf{x})$ predicts the corresponding class label.

2.1.1.1 Classification Metrics

To evaluate the performance of a classification model, various metrics can be employed:

- **Recall**: It measures the proportion of actual positives that are correctly identified by the model.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

- **F1-Score**: This metric is the harmonic mean of Precision and Recall. It is particularly useful in balancing the trade-off between Precision and Recall, providing a single measure of performance that considers both false positives and false negatives. It is especially important for evaluating imbalanced data to handle adjusted class proportions effectively. The F1-Score combines the sensitivity of the model to both false positives and false negatives, and is defined as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

where $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$ and $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$.

- **G-Mean**: This metric measures the balance between performance on both the positive and negative classes and is also useful in imbalanced datasets. The G-Mean is the geometric mean of sensitivity (Recall) and specificity, thus it provides insight into the accuracy of the classifier on both classes, and is defined as:

$$\text{G-Mean} = \sqrt{\text{Recall} \times \text{Specificity}},$$

where $\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}}$.

- **Regression**: It aims to establish a relationship between a dependent variable y and one or more independent variables $\mathbf{x}$, with the goal of finding a function that maps these input variables to a continuous output variable in $\mathbb{R}$. This is particularly useful for predicting continuous values. When the relationship between the dependent and independent variables is linear, it is addressed using linear regression. In contrast, nonlinear regression techniques are employed for nonlinear relationships.

The formula for a linear regression model, which includes multiple independent variables, is expressed as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \varepsilon,$$

where y represents the continuous output variable, β_0 is the intercept, $\beta_1, \dots, \beta_n$ are the coefficients of the features $x_1, \dots, x_n$, and ε is the error term, representing the difference between the observed and predicted values.

The coefficients $\beta_0, \beta_1, \ldots, \beta_n$ are estimated using the method of least squares, which minimizes the sum of the squared residuals (the differences between observed and predicted values):

$$\min_{\beta} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in}))^2.$$

2.1.1.2 Regression Metrics

To evaluate the performance of a regression model, various metrics can be employed:

- **Mean Squared Error (MSE)**: This metric measures the average squared difference between the actual and predicted values. It is useful for understanding the variance of the errors. Lower values of MSE indicate a better fit of the model to the data.
$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2,$$
where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of observations.
- **Mean Absolute Error (MAE)**: This metric measures the average absolute difference between the actual and predicted values. It provides a measure of model accuracy, without emphasizing large errors as MSE does. Lower MAE values indicate a better fit of the model.
$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|,$$
where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of observations.
- **R-squared (R^2) Coefficient**: This metric indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. It provides an indication of goodness-of-fit. An R^2 value of 1 indicates that the regression predictions perfectly fit the data.
$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2},$$
where y_i is the actual value, $\hat{y}_i$ is the predicted value, $\bar{y}$ is the mean of the actual values, and n is the number of observations.

2.1.2 Unsupervised Learning

In unsupervised learning, the goal is to find hidden patterns or intrinsic structures within input data. Clustering is one of the most common tasks in unsupervised learning. Consider a dataset $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where each $\mathbf{x}_i \in \mathbb{R}^d$. The objective of clustering is to partition the dataset into k clusters $\{C_1, C_2, \ldots, C_k\}$. The aim is to group objects in such a way that objects in the same cluster are more similar to each other than to those in other clusters. Two clustering algorithms are K-means and K-medoids.

- **K-means clustering** partitions data into k clusters by minimizing the within-cluster sum of squared Euclidean distances to cluster centroids, which are the arithmetic means of the data points in each cluster. This means that the algorithm is sensitive to outliers, as the mean can be easily skewed by extreme values. K-means minimizes:
$$\sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|x - \mu_i\|^2,$$
where $\mathbf{x}$ represents a data point, μ_i is the centroid of cluster i, and C_i denotes the set of data points in cluster i.

- **K-medoids clustering**, on the other hand, is more robust to outliers because it uses medoids, which are actual data points within each cluster that minimize the sum of distances (not squared distances) to other points in the cluster. This means that K-medoids minimizes:
$$\sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - m_i\|,$$
where k is the number of clusters, m_i represents the medoid of cluster i. The medoid can be considered a representative of the cluster, similar to how the median is a representative value in a dataset.

2.2 The Problem of High Dimensionality

The high dimensionality presents significant challenges in data analysis. In these high-dimensional spaces, data becomes sparse, which diminishes the effectiveness of traditional distance measures and, by extension, the concept of nearest neighbors. This problem is often known as *Curse of dimensionality* [42].

The choice of the distance metric plays an important role in the performance of algorithms that rely on distance calculations. The L_k norm, used to define such metrics, is expressed for any vector $x \in \mathbb{R}^d$ as:
$$\|x\|_k = \left(\sum_{i=1}^{d} |x_i|^k \right)^{1/k}.$$

The distance between two points $x, y \in \mathbb{R}^d$ using L_k norm is given by:
$$D(x, y) = \|x - y\|_k = \left(\sum_{i=1}^{d} |x_i - y_i|^k \right)^{1/k}.$$

Here, k determines the type of distance. For $k = 1$, it defines the Manhattan distance:
$$D(x, y) = \sum_{i=1}^{d} |x_i - y_i|.$$

For $k = 2$, the measure is known as the Euclidean distance:

$$D(x, y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}.$$

For $k \geq 3$, the distance is known as the Minkowski distance, which generalizes the Manhattan and Euclidean distances. It is particularly noted that the Manhattan distance may be more effective in high-dimensional settings due to its robustness against outliers and its ability to maintain sparsity [47]. Distances for $k \geq 3$ are known to magnify the impact of large differences, potentially difficulting the challenges of high dimensionality. That, combined with imbalanced class distributions introduces further complexity. Conventional machine learning models trained on such datasets may not adequately recognize critical minority class samples, leading to significant consequences in areas such as fraud detection, disaster management, or disease diagnosis. The predominance of the majority class in distance-based metrics often limit classifier performance. Increasing the volume of training data can sometimes mitigate this issue, as high-dimensional spaces require larger sample sizes to capture the underlying structure of the data and develop robust models [15]. However, this is not always possible.

2.3 Label noise

Noise in datasets manifests as irregularities or perturbations that complicate data analysis, introducing uncertainty and complicating the modeling process. A dataset quality is often measured by examining its attributes and class labels. Consequently, noise can be broadly categorized into two types: attribute noise and label noise [48]. Attribute noise pertains to inaccuracies in the data attributes that are used as independent variables in classification algorithms. This type of noise can stem from various sources such as data entry errors, measurement inaccuracies, the presence of outliers, or missing data. Such errors in the attributes can lead to misclassifications, impacting the dependent variables and adding additional complexity to machine learning algorithms [49]. Label noise is a significant issue in machine learning, negatively impacting performance metrics due to the potential decrease in model accuracy. Mislabeled instances can arise from subjectivity or communication errors during data labeling [50]. Typically, a label should correspond to the true class of a sample; however, it may be subjected to a noise process before being presented to the learning algorithm. It is important to clarify that mislabeled instances are not necessarily outliers or anomalies, which are themselves subjective concepts and distinct from noise in labels. Label noise is commonly classified into two primary types [48]:

- **Random Label Noise:** Each label has a fixed probability ρ of being incorrect, regardless of its true class. This can be expressed as:

  $$P(\tilde{Y} = j \mid Y = i) = \rho, \quad \forall\, i \neq j,$$

  where $\tilde{Y}$ is the observed noisy label and Y is the actual label of the data point.

- **Class-dependent Label Noise:** The probability of a label being incorrect depends on the actual class of the data point. This type of noise can be described using a noise transition matrix R where each entry R_{ij} represents the probability of the true class i being observed as class j:

  $$P(\tilde{Y} = j \mid Y = i) = R_{ij}.$$

  This model allows for different probabilities of label corruption depending on the actual class, with i and j representing the indices of the true and observed classes, respectively.

CHAPTER 3 KM-SMOTE ALGORITHM APPROACH

3.1 KM-SMOTE

We propose KM-SMOTE, a new algorithm that combines the K-medoids clustering technique with SMOTE to generate synthetic data samples. As shown on the left-hand side of Figure fig. 3.1, the algorithm starts with the original dataset. The initial step involves identifying the minority class in binary classifications or all classes except the majority class in multiclass settings. Subsequently, K-medoids clustering is applied to each minority class, identifying the specified number of medoids based on the number of clusters chosen during the algorithm's execution, resulting in the step shown in the middle of Figure fig. 3.1. K-medoids aims to minimize the total dissimilarity between the points in a cluster and the cluster's medoid. This objective function can be expressed as follows:

$$S = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, m_i) \tag{3.1}$$

where $d(x, m_i)$ measures the distance between a point x and the medoid m_i of the cluster C_i, and k is the number of clusters. The number of clusters used in K-medoids depends on the distribution of the classes within the data. If samples follow a homogeneous distribution, one cluster per class should be sufficient. Conversely, if the samples in the data do not follow a convex distribution, further clusters per class may be necessary. Finally, as seen on the right-hand side of Figure fig. 3.1, SMOTE is applied within each cluster to generate new samples by linearly interpolating between the selected sample and its k-nearest neighbors. The pseudocode detailing the steps of the KM-SMOTE algorithm is provided in Algorithm 1.

The motivation for using K-medoids over other clustering algorithms is based on two main reasons. First, it is more robust to outliers, as it operates with medoids, which are real central points that are not affected by anomalous points [51]. Second, it allows the use of alternative distance metrics to the squared L_2 norm, which is useful in situations of high dimensionality, as discussed in Section 2.2. The choice of the distance metric is important as it influences the formation of clusters and the generation of synthetic samples. An appropriate distance metric ensures that medoids accurately represent the

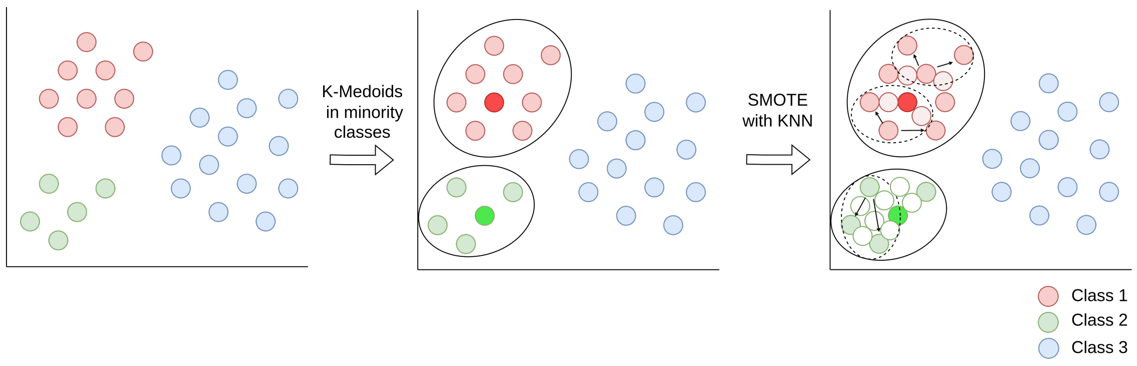

Figure 3.1: KM-SMOTE method working scheme.

main characteristics of the data, enhancing the effectiveness of the subsequent SMOTE [42].

Algorithm 1: KM-SMOTE Algorithm

```
Data: Features $X$, Labels $y$, Distance Metric $distance$, Number of Clusters $n\_clusters$
Result: $X\_resampled$, $y\_resampled$
Resampling Process:
Initialize $X\_resampled \leftarrow X$, $y\_resampled \leftarrow y$
classes $\leftarrow$ All classes but majority class
foreach class in classes do
    $K - medoids \leftarrow$ K-medoids($n\_clusters$, $distance$)
    Train $K - medoids$ on $X[class]$
    $clusters \leftarrow$ Assign data points in $X[class]$ to clusters
    foreach $cluster$ in $clusters$ do
        N $\leftarrow \frac{\text{samples in the majority class} - \text{samples in current class}}{\text{number of clusters}}$
        Initialize and train $NearestNeighbors$ with $distance$ on cluster's samples
        foreach $i \leftarrow 1$ to N do
            $s_1 \leftarrow$ Select random sample from $cluster$
            $neighbors \leftarrow$ Get nearest neighbors of $s_1$ using $NearestNeighbors$
            $s_2 \leftarrow$ Select random sample from $neighbors$
            $synthetic\_sample \leftarrow s_1 +$ rand(0,1) $\cdot (s_2 - s_1)$
            $X\_resampled \leftarrow X\_resampled \cup synthetic\_sample$
            $y\_resampled \leftarrow y\_resampled \cup class$
return $X\_resampled$, $y\_resampled$
```

3.2 Experiments

3.2.1 Methodology

We follow the pipeline shown in Figure 3.2. Data is collected from various sources: License Plate Recognition (LPR), the General Directorate of Traffic (DGT), the National Institute of Statistics (INE), and survey questionnaires. Initially, these data are analyzed and preprocessed, which includes cleaning, handling missing values, and normalization. This allows us to build a dataset focused on our smart village area. Once this step is completed, we extract relevant features to build a model. We then conduct an evaluation process that involves using different classification models, oversampling methods, and distance metrics on the models. We use the models K-Nearest Neighbors (KNN), Logistic Regression (LR), Support Vector Machine (SVM), and Multilayer Perceptron (MLP). The oversampling methods we use are SMOTE, K-means-SMOTE, and CURE-SMOTE. The distances we use are Euclidean (associated with the L_2 norm) and Manhattan (associated with the L_1 norm). We use these results to build a model in which the new predicted variable, "likelihood", is used to enrich the dataset for the dynamic pricing algorithm.

In the second phase, with that predicted likelihood, we build a dynamic pricing algorithm, in which the previous model will contribute with that behavioral component to the price assignment. Finally, to measure the impact of the different features in that dynamic pricing algorithm, we build a regression model.

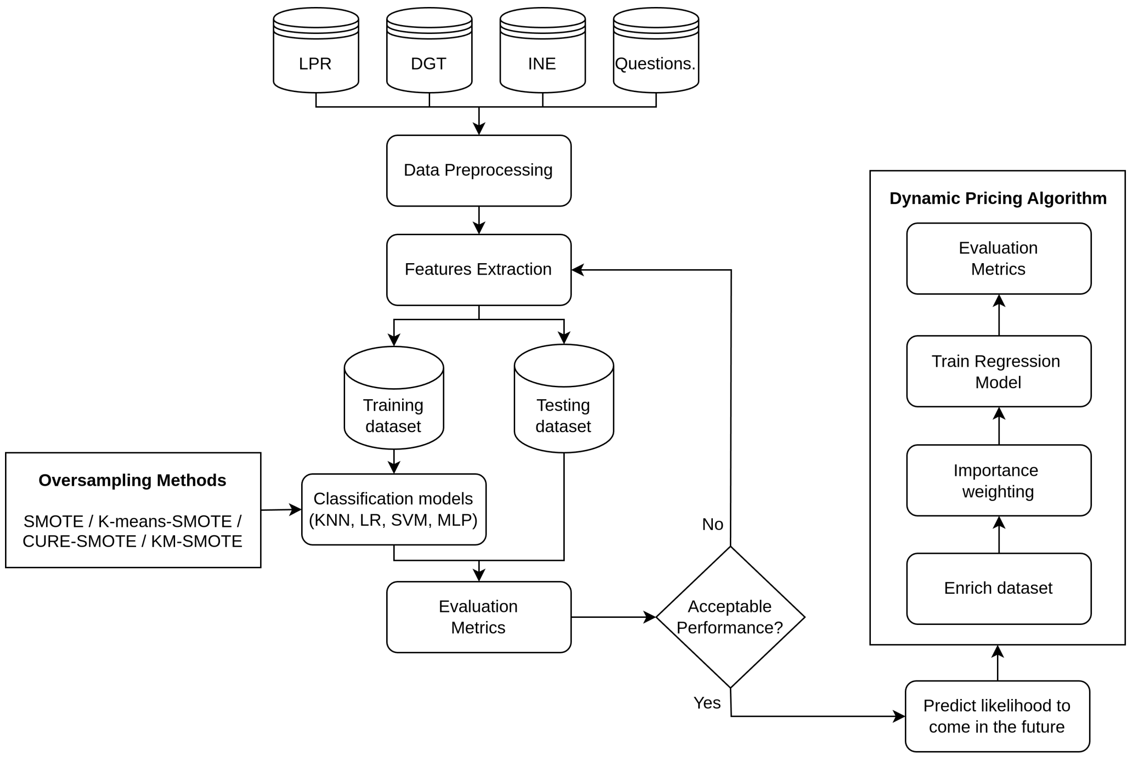

Figure 3.2: Pipeline schema

3.2.2 Data overview

To test the different algorithms in our smart village area, we build a dataset from various sources including LPR, DGT, INE, and questionnaires, where instances are joined with the vehicle's license plate number. The dataset is available in [52] and it comprises:

1. **Visitation data** collected from a sensor network consisting of four Hikvision LPR IP cameras equipped with Deep Learning-based License Plate Recognition (LPR) technology.

2. **Geographic data** sourced from the Spanish Directorate-General for Traffic (DGT[1]), providing variables related to vehicle origin.

3. **Demographic data** sourced from the National Statistics Institute (Spanish: Instituto Nacional de Estadística, INE[2]), offering further insights into vehicle origins.

4. **Questionnaire** responses gathered in January, March, and July 2023, excluding local residents to maintain proportional representation similar to the LPR data. Questionnaires, which were conducted in a parking lot, included questions such as license plate number, residential postcode, total visits to the area, overnight stays, etc. One question assessed respondents' inclination to revisit the area if a parking toll is implemented, with response options: "visit more", "visit less", or "no affect". This variable serves as the prediction class for this dataset.

All this information has been preprocessed and analyzed, including normalization of variables, imputation of missing values, and elimination of outliers, so that the information is ready for the proposed algorithm. A summary of the main characteristics of the dataset can be seen in the table 3.1.

Data Category	Variable	Data Type	Description
Vehicle Behavior (LPR Cameras)	visit_time	Time	Total duration of stay.
	distance	Float	Total distance covered in kilometers within the area.
	nights	Integer	Count of nights stayed.
	visits_dif_weeks	Integer	Number of different weeks with at least one visit.
	visits_dif_months	Integer	Number of different months with at least one visit.
	fidelity	Float	Visits after maintaining a fidelity of at least five days.
	cumulative_entries	Integer	Total count of entries.
Holiday Context	num_holiday	Integer	Total number of holidays.
	num_workday	Integer	Total number of workdays.
	num_high_season	Integer	Number of days spent during high season.
	num_low_season	Integer	Number of days spent during low season.
	entry_in_holiday	Boolean	Entry during holidays.
	entry_in_high_season	Boolean	Entry during high season.
Demographic and Economic Data (INE)	avg_gross_income	Float	Average gross income of the origin area of the vehicle.
	population	Integer	Population size of the orithin of the vehicle city/town.
	is_resident	Boolean	Tourist is resident from Pampaneira, Capileira or Bubión.
Geographic Data (DGT)	km_to_dest	Float	Distance in kilometers between the origin of the vehicle and destination.
	num_seats	Integer	Number of seats in the vehicle.
	environmental_distinctive	String	Blue - ECO - C Green - B Yellow (scraped).
	co2_emissions	Float	CO2 emissions in g/km (imputed).
Questionnaires	likelihood	Integer	Predicted class label indicating the likelihood to repeat the visit.

Table 3.1: Selected Dataset Variables

[1]https://sede.dgt.gob.es/es/vehiculos/
[2]https://www.ine.es/index.htm

3.2.3 Model Training

We use a 5-fold cross-validation method to validate our models. This method divides the dataset into five equal parts. Each model is trained and tested five times, with each part used once as the test set and four times as part of the training set. This method helps us evaluate the model's performance and stability across different conditions. We divide the data in three sets, training, validation and test, with a 70-10-20 proportion, respectively.

We avoid using accuracy as a metric in those imbalanced datasets because it tends to favor the majority class and might give a high score while overlooking the minority class. Instead, we use Recall, F1-Score, G-Mean to evaluate our classifiers across the dataset [47, 53]. These metrics ensure that our models are effective across all categories, not just the dominant class.

3.2.4 Results

We use four different classifiers to evaluate the proposed algorithm and compared it across various algorithms: KNN, MLP, LR, and SVM. The Table table 3.2 shows the results of the differents metrics with the diffents algorithms we used: the base case (None), SMOTE, K-means-SMOTE, CURE-SMOTE and KM-SMOTE with L_1 and L_2 norm.

Table 3.2: Performance comparison of oversampling techniques across classifiers

Method	**KNN**			**MLP**			**LR**			**SVM**		
	Recall	F1-Score	G-Mean	Recall	F1-Score	G-Mean	Recall	F1-Score	G-Mean	Recall	F1-Score	G-Mean
None	0.6408	0.6315	0.6972	0.6311	0.6326	0.6815	0.7212	0.6103	0.6728	0.5701	0.4497	0.5712
SMOTE	0.6117	0.6121	0.6830	0.7087	0.7046	0.7374	0.7401	0.6041	0.6842	0.5437	0.4236	0.5780
k-means-SMOTE	0.6311	0.6315	0.6901	0.6893	0.6895	0.7346	0.6214	0.6142	0.6748	0.5534	0.4447	0.5840
CURE-SMOTE	0.6408	0.6413	0.6981	0.6311	0.6324	0.6822	0.6214	0.6103	0.6728	0.5437	0.4236	0.5737
KM-SMOTE+L_1	**0.6712**	**0.6592**	**0.7133**	**0.7206**	**0.7064**	**0.7586**	**0.7622**	**0.6436**	**0.7139**	0.6112	**0.6001**	**0.6761**
KM-SMOTE+L_2	0.6563	0.6413	0.6981	0.7114	0.6908	0.7492	0.7412	0.6319	0.6987	**0.6128**	0.5962	**0.6761**

3.2.5 Discussion

There is noticeable variability in the results of the different classification algorithms used. KM-SMOTE achieves the best results in all metrics across all methods. As observed in Table 3.2 and Figures 3.3, 3.4, and 3.5, KM-SMOTE in its both versions on SVM shows a significant improvement in the results compared to the other methods. In KNN, MLP, and LR, KM-SMOTE also improves performance, but the improvement is not as pronounced. The other clustering-based algorithms, K-means-SMOTE and CURE-SMOTE, do not perform as well. In fact, in the recall score shown in Figure 3.3, the base case, SMOTE, without any previous clustering even yields better results.

KM-SMOTE with the L_1 norm, which utilizes the Manhattan distance, generally yields the highest scores across all classifiers. This significantly improves performance compared to the baseline and SMOTE, effectively balancing sensitivity and specificity across different classes, and managing noise and class imbalance more efficiently.

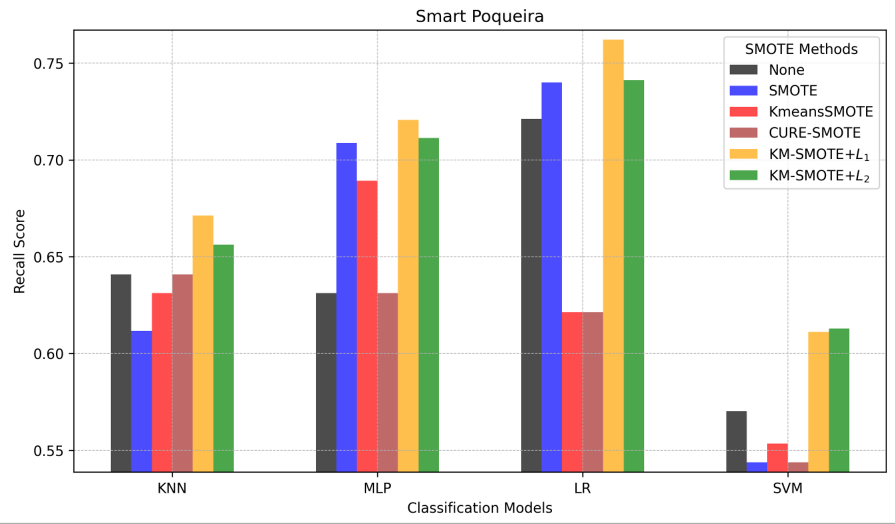

Figure 3.3: Recall comparison

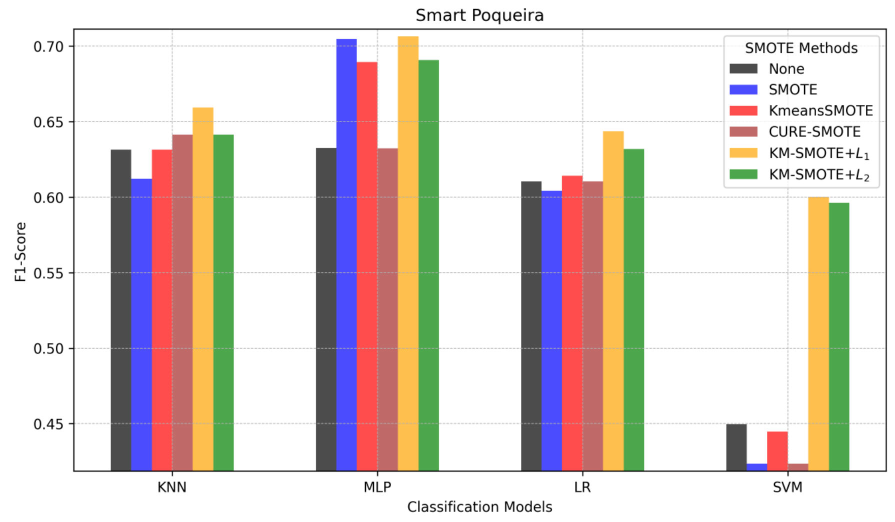

Figure 3.4: F1-Score comparison across datasets

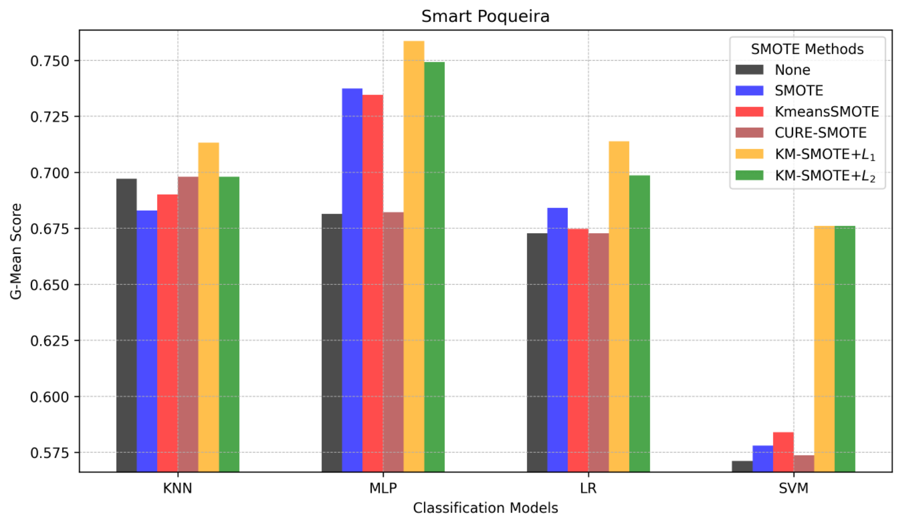

Figure 3.5: G-Mean comparison across datasets

CHAPTER 4

DYNAMIC PRICING ALGORITHM

4.1 Pricing Strategies

We propose two pricing strategies to utilize on our data, which consists of various visit instances. Each instance in the dataset is represented by a vector of features. Table 4.1 shows the data representation.

	Feature 1	**Feature 2**	**Feature 3**	...	**Feature m**
Visit 1	Value 1	Value 2	Value 3	...	Value m
Visit 2	Value 1	Value 2	Value 3	...	Value m
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Visit n	Value 1	Value 2	Value 3	...	Value m

Table 4.1: Data Representation

To determine the price of a tourist visit, the algorithm uses a rule-based method, where each feature contributes partially to the final price. The significance of variables varies, so the weights assigned to the features are not the same. To address this, we sort the features according to two different criteria.

1. **Strategy 1**: We assign each variable an importance, and prices are determined accordingly. We order the variables in a descending sequence using a common ratio r, with $r \in (0, 1)$. This implies that if the first variable has an importance I, the subsequent variable will have an importance of $r \cdot I$, and so on until the least important variable will have an assigned value of $r^{m-1} \cdot I$, where m is the total number of features.

Feature	Importance
X_1	I
X_2	$r \cdot I$
$\vdots$	$\vdots$
X_m	$r^{m-1} \cdot I$

Table 4.2: Strategy 1

This is done by employing a geometric progression of weights. The weight for the k-th feature is $w_k = r^{k-1}$, indicating the relative importance between consecutive features. We normalize the weights to ensure $\sum_{k=1}^{m} w_k = 1$. The normalization is as follows:

$$\text{Normalized Importance}_k = \frac{r^{k-1} \cdot I}{I \sum_{j=0}^{m-1} r^j} = \frac{r^{k-1}}{\sum_{j=0}^{m-1} r^j}.$$

This ensures that the sum of the normalized importances is 1, making them suitable for use as weights in the pricing algorithm.

2. **Strategy 2**: We consider the existence of different categories within the set of all features, so the variables do not have a uniform descending order across the entire set. Instead, we assign different weights within each category, ordering the features within each category using Strategy 1. This strategy ensures that features are prioritized appropriately within their respective categories. We present this strategy as follows:

Category	Feature	Importance
Category 1	Feature$_{1,1}$	$I_{1,1}$
	$\vdots$	$\vdots$
	Feature$_{1,t_1}$	$r_1^{t_1-1} \cdot I_{1,1}$
Category 2	Feature$_{2,1}$	$I_{2,1}$
	$\vdots$	$\vdots$
	Feature$_{2,t_2}$	$r_2^{t_2-1} \cdot I_{2,1}$
	$\vdots$	
Category T	Feature$_{T,1}$	$I_{T,1}$
	$\vdots$	$\vdots$
	Feature$_{T,t_T}$	$r_T^{t_T-1} \cdot I_{T,1}$

Table 4.3: Strategy 2

If there are T categories, denoted as $c_1, \ldots, c_T$, with associated weights $w_1, \ldots, w_T$, it holds that $w_1 + w_2 + \ldots + w_T = 1$. Additionally, within a category c_i, its assigned price w_i satisfies:

$$w_i = \sum_{j=0}^{t_i-1} r_i^j \cdot I_i, \quad i = 1, \ldots, T.$$

Thus, it also holds that:

$$\sum_{i=1}^{T} w_i = \sum_{i=1}^{T} \sum_{j=0}^{t_i-1} r_i^j \cdot I_i = 1.$$

4.1.1 Price Assignation

The pricing function $P : \mathbf{X}_i \to \mathbb{R}$ assigns a price to each instance i based on the weighted sum of its features. Here, $\mathbf{X}_i$ represents the feature vector for the i-th instance, and the function integrates weighted factors from different categories $c_1, \ldots, c_T$. The formula for the pricing function is:

$$P(\mathbf{X}_i) = \alpha + (\beta - \alpha) \cdot \sum_{k=1}^{m} w_k \cdot f(x_{ik}, x_{\min k}, x_{\max k}),$$

where:

- α and β are the predefined lower and upper bounds of the price, respectively.
- w_k, $k = 1, \ldots, m$, are the weights derived from Strategy 1 or Strategy 2, with $\sum_{k=1}^{m} w_k = 1$.
- $f(x_{ik}, x_{\min k}, x_{\max k})$ is the Min-Max normalization function. To scale each feature x_{ik} to the range $[0, 1]$, we apply a modified version of the Min-Max normalization function:

$$f(x_{ik}, x_{\min k}, x_{\max k}) = \begin{cases} \frac{x_{ik} - x_{\min k}}{x_{\max k} - x_{\min k}}, & \text{if lower value of the feature is better,} \\ 1 - \frac{x_{ik} - x_{\min k}}{x_{\max k} - x_{\min k}}, & \text{if higher value of the feature is better.} \end{cases}$$

- x_{ik} represents the value of the k-th feature for instance i, $x_{\min k}$ is the minimum value, and $x_{\max k}$ is the maximum value of the k-th feature across all instances.

In the price assignment process, we use the normalized and encoded feature values to compute the weighted average. The computed weighted average is then scaled between the predefined bounds α and β to produce the final price. This systematic approach ensures that each feature's contribution is properly accounted for, resulting in a fair and accurate pricing mechanism based on the tourist's characteristics.

4.2 Experiments

We consider the same dataset from Section 3.2.2, which is derived from the study by [52]. In this section, we conclude with the final part of the pipeline shown in Figure 3.2. Here, we use the likelihood output feature obtained after applying different classification comparisons to our data as input. This behavioral variable contributes to the dynamic pricing algorithm for price assignment, along with the features listed in Table 4.4.

4.2.1 Data Preparation

We create a new dataset containing only environmental, socio-economic, and behavioral variables to assign prices to each vehicle. The objective is that vehicles scoring better in these variables will have a lower price, while those scoring worse will have a higher price. Table 4.4 shows the variables we use in this study, along with the category to which they belong. The table also indicates whether a higher value of the feature is better.

Category	Variable	Type	Better High
Environmental	co2_emissions	Numeric	False
	distance	Numeric	False
	num_seats	Numeric	True
Socio-economic	entry_in_holiday	Categorical	False
	entry_in_high_season	Categorical	False
	is_resident	Categorical	True
	num_nights	Numeric	True
	visit_hours	Numeric	True
Behavioral	cumulative_entries	Numeric	True
	likelihood_to_come	Categorical	True

Table 4.4: Description of variables with Better High column

For this analysis, we separate the variables into categorical and numeric types. We encode categorical variables as binary values (0 or 1) based on their specific conditions. For instance, "is_resident" is 1 if the individual is a resident of Alpujarra and 0 otherwise. For numeric variables, we determine the lower and upper bounds using the Interquartile Range (IQR). The IQR is defined as $IQR = Q3 - Q1$. Using the IQR, we calculate the minimum and maximum values ($x_{\min}$ and $x_{\max}$) for each variable based on their respective first quartile ($Q1$) and third quartile ($Q3$) from historical data. The formula is as follows:

$$[x_{\min}, x_{\max}] = [Q1 - 1.5 \cdot IQR, Q3 + 1.5 \cdot IQR].$$

The "Better High" column from Table 4.4 indicates whether higher values are preferable. If this parameter is false, the normalized value is inverted to reflect that lower values are better [54]. This inversion is applied to both numerical and categorical variables to ensure they are appropriately scaled and encoded. This method ensures that values near the maximum incur the maximum cost, and values near the minimum incur the minimum cost for each variable, thus accurately reflecting their influence on the final price calculation.

4.2.2 Regression Model

We create a regression model to understand the impact of each feature on the final price. By analyzing the coefficients obtained from the regression model, we can determine how much each variable contribute to the final assigned price. This analysis allows us to assess the relative importance of different features with respect to the original assigned weights. Additionally, it helps to verify the effectiveness of our pricing algorithm by comparing the assigned weights in the price assignation with the learned coefficients from the regression model.

Given a new vehicle, we can test the predictive performance of our model using new data. Historical data was used to train the model, enabling us to identify any discrepancies or biases in the initial price assignment. The performance of the model is evaluated using the metrics MSE, MAE and R^2.

4.2.3 Results

We used Strategy 2 with the categories from Table 4.4. The weights for the environmental, socio-economic, and behavioral categories were assigned equally $w_1 = w_2 = w_3 = \frac{1}{3}$. Within each category, we applied a common ratio of $r = 0.9$. Additionally, we set the bounds $[\alpha, \beta] = [0.5, 1]$ to ensure that the prices fall within this range.

Table 4.5 shows the weights and the regression coefficients after applying Strategy 2. Table 4.6 presents the results of the evaluation metrics. Figures 4.1 and 4.2 display the plots of the regression model with 100 and 500 samples, respectively.

Category	Feature	[Min, Max]	Weights	Regression coefficients
Environmental	co2_emissions	[104.5, 140.5]	0.1230	0.0146
	distance	[0, 20.25]	0.1107	0.0148
	num_seats	[2, 5]	0.0996	-1.4242e-06
Socio-economic	entry_in_holiday	N/A	0.0814	0.0418
	entry_in_high_season	N/A	0.0733	0.0393
	is_resident	N/A	0.0659	-0.0478
	num_nights	[0, 7]	0.0593	-0.0192
	visit_hours	[0, 166.95]	0.0534	-0.0006
Behavioral	likelihood_to_come	N/A	0.1754	-0.0240
	cumulative_entries	[0, 61]	0.1579	-0.0207

Table 4.5: Results: Weights and regression coefficients

Metric	MSE	MAE	R2
Values	0.00064	0.0182	0.7933

Table 4.6: Model Performance Metrics

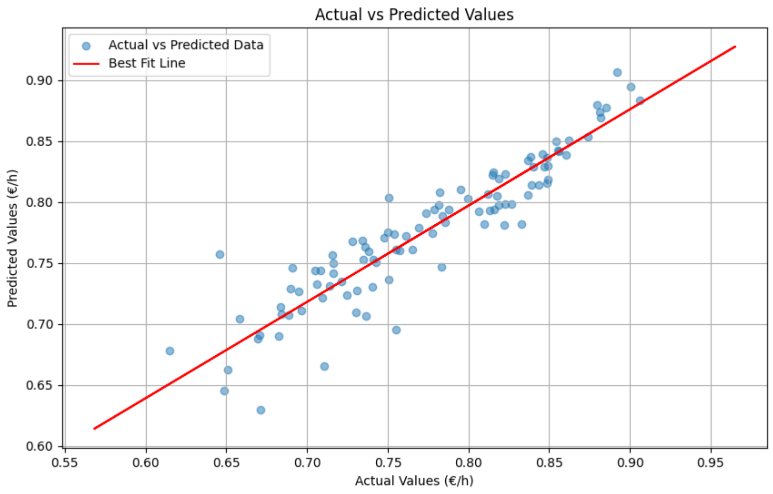

Figure 4.1: Regression model plot with 100 samples

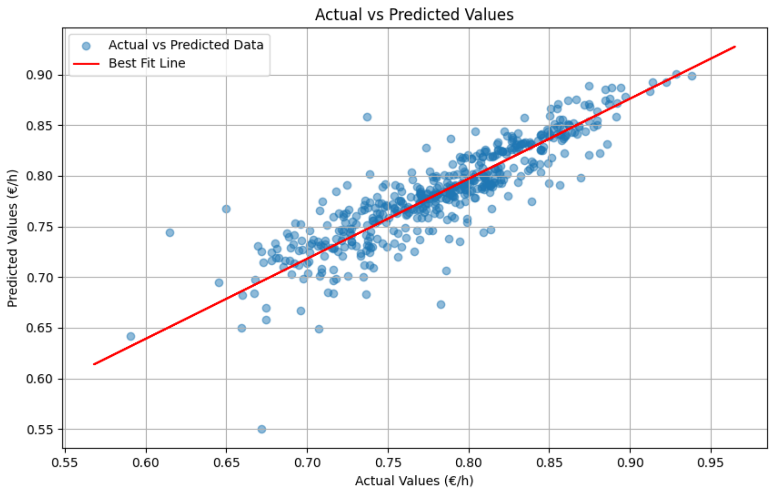

Figure 4.2: Regression model plot with 500 samples

4.2.4 Discussion

The weights from Table 4.5 align with the selected strategy, where the weights from each category c_i sum to w_i, and the sum of all weights $w_1, \ldots, w_T$ equals 1. In this case, the features with the highest weights are the behavioral features likelihood_to_come and cumulative_entries, which makes sense because there are only two features in their category.

According to the regression coefficients, positive values indicate that those features increase the price by that amount. Similarly, negative values indicate a decrease in the price. For example, for co2_emissions, if a vehicle produces 104.5 g/km or lower (minimum bound), it will contribute 0 to the final price. Conversely, for 140.5 g/km or higher (maximum bound), it will contribute 0.0146 to the final price. Intermediate values will be assigned a contribution between 0 and 0.0146.

The MSE value of 0.00064 indicates a small average squared difference between the observed actual outcomes and the outcomes predicted by the model, suggesting high accuracy. The MAE value of 0.0182 further supports this, showing that the average magnitude of the errors in our predictions is low. The R^2 value of 0.7933 suggests that approximately 79.33% of the variability in the price can be explained by the model, which is a good indication of the model's explanatory power. These results show that the pricing model is effective and reliable, with a strong predictive performance.

CHAPTER 5 CONCLUSIONS

This project is divided into two main parts. First, we introduced a novel oversampling method, KM-SMOTE, that combines K-medoids clustering with the SMOTE algorithm. It is specifically designed to enhance the performance of models on unbalanced datasets affected by noise and high dimensionality. We validated this method through a practical use case focused on smart villages, where we applied this algorithm to noisy sensor and survey data.

The proposed algorithm showed improvement in all metrics compared to other clustering-based algorithms and the original SMOTE. This improvement is due to the presence of outliers and noisy samples in this dataset, where the use of medoids contributes additional benefits. The class balancing proved effective in all cases, as indicated by the F1-Score, G-Mean, and Recall metrics. We also demonstrated that the problem of high dimensionality is reduced by using the Manhattan distance metric instead of the Euclidean metric. The results highlight the potential of this proposed algorithm, enabling data scientists and researchers to achieve more reliable and interpretable results, especially in scenarios with data imperfections.

Second, using the predictive power of our model, we predicted the willingness to return to the area after the implementation of a toll in a parking lot. Using that behavioral feature along with other environmental and socio-economic factors, we created a dynamic pricing algorithm to categorize vehicles based on these three dimensions: social, economic, and environmental, with a behavioral component. Each vehicle contributes according to these features. We used historical data to adjust variables in the model and created a dynamic price model using two strategies. Additionally, we developed a regression model to study how coefficients affect price weights, providing deeper insights into the pricing mechanism.

This contribution can be implemented in the Alpujarra region, assigning toll prices to vehicles. Accompanied by continuous data collection over several months, this dynamic pricing algorithm will allow dynamic price assignments to new vehicles entering the area. It will also prepare stakeholders to manage and prevent overcrowding in the Barranco de Poqueira, which was the main objective of this project.

5.1 Limitations

This study presents some limitations that open up opportunities for future research. Firstly, the amount of data collected is limited to specific periods and locations within the Sierra Nevada National Park, which may not be representative of all possible seasonal conditions or traffic variations. Additionally, the use of survey data may introduce biases due to subjective responses that the KM-SMOTE algorithm cannot fully learn. It may be beneficial to include other non-clustering-based and novel oversampling methods based on neural networks and compare them to assess KM-SMOTE's performance. Another limitation is that the data is collected from cameras, which sometimes suffer from outages, resulting in data loss. Finally, incorporating external factors beyond the specific set of features (environmental, socio-economic, and behavioral), such as public transportation policies or changes in road infrastructure, could provide a more comprehensive analysis. Although the KM-SMOTE algorithm has been shown to improve performance on our imbalanced and noisy dataset, future research should explore the integration of these additional factors to create a more comprehensive and adaptable pricing model.

5.2 Future Work

The practical implementation of this model in the Alpujarra region will require continuous monitoring and adjustments based on real-time data to adapt to changing visitor dynamics and traffic patterns. Future work could focus on developing a real-time management platform that allows park managers to quickly and effectively adjust dynamic pricing policies, ensuring a sustainable balance between environmental preservation and tourism.

Additionally, future studies could investigate tourist patterns based on the routes they take within the national park. This could provide insights into the most visited areas and help in better managing foot traffic and preserving sensitive areas. Predicting tourism peaks using Google Trends is another avenue for future research, as it could help in anticipating high visitor periods and implementing proactive measures to manage the influx effectively.

BIBLIOGRAPHY

[1] Siti Dianah Abdul Bujang, Ali Selamat, Roliana Ibrahim, Ondrej Krejcar, Enrique Herrera-Viedma, Hamido Fujita, and Nor Azura Md Ghani. Multiclass prediction model for student grade prediction using machine learning. *IEEE Access*, 9:95608–95621, 2021.

[2] Ahmad Aburayya, S Salloum, K Alderbashi, Fanar Shwedeh, Yara Shaalan, Raghad Alfaisal, S Malaka, and Khaled Shaalan. Sem-machine learning-based model for perusing the adoption of metaverse in higher education in uae. *International Journal of Data and Network Science*, 7(2):667–676, 2023.

[3] Tarakashar Das, Sabrina Mobassirin, Syed Md Minhaz Hossain, Aka Das, Anik Sen, Khaleque Md Aashiq Kamal, and Kaushik Deb. Patient questionnaires based parkinson's disease classification using artificial neural network. *Annals of Data Science*, pages 1–44, 2023.

[4] Francisco M Garcia-Moreno, Maria Bermudez-Edo, María José Rodríguez-Fórtiz, and José Luis Garrido. A cnn-lstm deep learning classifier for motor imagery eeg detection using a low-invasive and low-cost bci headband. In *2020 16th international conference on intelligent environments (IE)*, pages 84–91. IEEE, 2020.

[5] Deepak Kumar, Pradeepta Kumar Sarangi, and Rajit Verma. A systematic review of stock market prediction using machine learning and statistical techniques. *Materials Today: Proceedings*, 49:3187–3191, 2022.

[6] Mark Eshwar Lokanan and Kush Sharma. Fraud prediction using machine learning: The case of investment advisors in canada. *Machine Learning with Applications*, 8:100269, 2022.

[7] Fotis Kitsios, Eleftheria Mitsopoulou, Eleni Moustaka, and Maria Kamariotou. User-generated content behavior and digital tourism services: A sem-neural network model for information trust in social networking sites. *International Journal of Information Management Data Insights*, 2(1):100056, 2022.

[8] Daniel Bolaños-Martinez, Maria Bermudez-Edo, and Jose Luis Garrido. Clustering pipeline for vehicle behavior in smart villages. *Information Fusion*, 104:102164, 2024.

[9] Abhinav Jain, Hima Patel, Lokesh Nagalapatti, Nitin Gupta, Sameep Mehta, Shanmukha Guttula, Shashank Mujumdar, Shazia Afzal, Ruhi Sharma Mittal, and Vitobha Munigala. Overview and importance of data quality for machine learning tasks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3561–3562, 2020.

[10] Dimitrios Buhalis. Technology in tourism-from information communication technologies to etourism and smart tourism towards ambient intelligence tourism: a perspective article. *Tourism Review*, 75(1):267–272, 2020.

[11] Arpit Shukla, Pronaya Bhattacharya, Sudeep Tanwar, Neeraj Kumar, and Mohsen Guizani. Dwara: A deep learning-based dynamic toll pricing scheme for intelligent transportation systems. *IEEE Transactions on Vehicular Technology*, 69(11):12510–12520, 2020.

[12] Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal*, 32(4):791–813, 2023.

[13] Taha Mansouri, Mohammad Reza Sadeghi Moghadam, Fatemeh Monshizadeh, and Ahad Zareravasan. Iot data quality issues and potential solutions: a literature review. *The Computer Journal*, 66(3):615–625, 2023.

[14] Yuan Gao. Forecast model of perceived demand of museum tourists based on neural network integration. *Neural Computing and Applications*, 33:625–635, 2021.

[15] Ulisses Braga-Neto. *Fundamentals of pattern recognition and machine learning*. Springer, 2020.

[16] Vikash Kumar and Ditipriya Sinha. Synthetic attack data generation model applying generative adversarial network for intrusion detection. *Computers & Security*, 125:103054, 2023.

[17] Zhiqiang Wan, Yazhou Zhang, and Haibo He. Variational autoencoder based synthetic data generation for imbalanced learning. In *2017 IEEE symposium series on computational intelligence (SSCI)*, pages 1–7. IEEE, 2017.

[18] Dina Elreedy, Amir F Atiya, and Firuz Kamalov. A theoretical distribution analysis of synthetic minority oversampling technique (smote) for imbalanced learning. *Machine Learning*, pages 1–21, 2023.

[19] XW Liang, AP Jiang, Tao Li, YY Xue, and GT Wang. Lr-smote—an improved unbalanced data set oversampling based on k-means and svm. *Knowledge-Based Systems*, 196:105845, 2020.

[20] Yuan Bao and Sibo Yang. Two novel smote methods for solving imbalanced classification problems. *IEEE Access*, 11:5816–5823, 2023.

[21] Md Manjurul Ahsan, Shivakumar Raman, and Zahed Siddique. Bsgan: A novel oversampling technique for imbalanced pattern recognitions. *arXiv preprint arXiv:2305.09777*, 2023.

[22] Xuchun Wang, Hao Ren, Jiahui Ren, Wenzhu Song, Yuchao Qiao, Zeping Ren, Ying Zhao, Liqin Linghu, Yu Cui, Zhiyang Zhao, et al. Machine learning-enabled risk prediction of chronic obstructive pulmonary disease with unbalanced data. *Computer Methods and Programs in Biomedicine*, 230:107340, 2023.

[23] Palak Gupta, Anmol Varshney, Mohammad Rafeek Khan, Rafeeq Ahmed, Mohammed Shuaib, and Shadab Alam. Unbalanced credit card fraud detection data: a machine learning-oriented comparative study of balancing techniques. *Procedia Computer Science*, 218:2575–2584, 2023.

[24] Anthony Anggrawan, Hairani Hairani, and Christofer Satria. Improving svm classification performance on unbalanced student graduation time data using smote. *International Journal of Information and Education Technology*, 13(2):289–295, 2023.

[25] Le Wang, Meng Han, Xiaojuan Li, Ni Zhang, and Haodong Cheng. Review of classification methods on unbalanced data sets. *IEEE Access*, 9:64606–64628, 2021.

[26] Funa Zhou, Shuai Yang, Hamido Fujita, Danmin Chen, and Chenglin Wen. Deep learning fault diagnosis method based on global optimization gan for unbalanced data. *Knowledge-Based Systems*, 187:104837, 2020.

[27] Po-Jen Chuang and Pang-Yu Huang. B-vae: a new dataset balancing approach using batched variational autoencoders to enhance network intrusion detection. *The Journal of Supercomputing*, 79(12):13262–13286, 2023.

[28] Huixin Tian, Chunzhi Tian, Kun Li, and Weinan Jia. Unbalanced regression sample generation algorithm based on confrontation. *Information Sciences*, 642:119157, 2023.

[29] Ayyoub EL HARIRI, Mohammed MOUITI, Omar HABIBI, and Mohamed LAZAAR. Improving deep learning performance using sampling techniques for iot imbalanced data. *Procedia Computer Science*, 224:180–187, 2023.

[30] Shan Guan, Haiqi Yang, and Tongyu Wu. Transformer fault diagnosis method based on tlr-adasyn balanced dataset. *Scientific Reports*, 13(1):23010, 2023.

[31] Damien Dablain, Bartosz Krawczyk, and Nitesh V Chawla. Deepsmote: Fusing deep learning and smote for imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[32] A Kiran and S Saravana Kumar. A comparative analysis of gan and vae based synthetic data generators for high dimensional, imbalanced tabular data. In *2023 2nd International Conference for Innovation in Technology (INOCON)*, pages 1–6. IEEE, 2023.

[33] Shengli Wang, Mulin Xieshi, Zhangpeng Zhou, Xiang Zhang, Xujie Liu, Zeyi Tang, Yuxing Dai, Xuexin Xu, and Pingyuan Lin. Two-channel vae-gan based image-to-video translation. In *International Conference on Intelligent Computing*, pages 430–443. Springer, 2022.

[34] Jichao Bao, Liangping Li, and Arden Davis. Variational autoencoder or generative adversarial networks? a comparison of two deep learning methods for flow and transport data assimilation. *Mathematical Geosciences*, 54(6):1017–1042, 2022.

[35] Anuraganand Sharma, Prabhat Kumar Singh, and Rohitash Chandra. Smotified-gan for class imbalanced pattern classification problems. *Ieee Access*, 10:30655–30665, 2022.

[36] Pamela Grimm. Social desirability bias. *Wiley international encyclopedia of marketing*, 2010.

[37] Chin-Wei Wang, Yuning Hao, Riccardo Di Gianfilippo, James Sugai, Jiaqian Li, Wang Gong, Kenneth S Kornman, Hom-Lay Wang, Nobuhiko Kamada, Yuying Xie, et al. Machine learning-assisted immune profiling stratifies peri-implantitis patients with unique microbial colonization and clinical outcomes. *Theranostics*, 11(14):6703, 2021.

[38] Nur Ulfa Maulidevi, Kridanto Surendro, et al. Smote-lof for noise identification in imbalanced data classification. *Journal of King Saud University-Computer and Information Sciences*, 34(6):3413–3423, 2022.

[39] Tanapol Kosolwattana, Chenang Liu, Renjie Hu, Shizhong Han, Hua Chen, and Ying Lin. A self-inspected adaptive smote algorithm (sasmote) for highly imbalanced data classification in healthcare. *BioData Mining*, 16(1):15, 2023.

[40] Xinyi Zhu, Hongbing Zhang, Quan Ren, Dailu Zhang, Fanxing Zeng, Xinjie Zhu, and Lingyuan Zhang. An automatic identification method of imbalanced lithology based on deep forest and k-means smote. *Geoenergy Science and Engineering*, 224:211595, 2023.

[41] Li Ma and Suohai Fan. Cure-smote algorithm and hybrid algorithm for feature selection and parameter optimization based on random forests. *BMC bioinformatics*, 18:1–18, 2017.

[42] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *Database Theory—ICDT 2001: 8th International Conference London, UK, January 4–6, 2001 Proceedings 8*, pages 420–434. Springer, 2001.

[43] Marcella Scrimitore. Quantity competition vs. price competition under optimal subsidy in a mixed oligopoly. *Economic Modelling*, 42:166–176, 2014.

[44] Peter P Belobaba. Or practice—application of a probabilistic decision model to airline seat inventory control. *Operations Research*, 37(2):183–197, 1989.

[45] Gerard P Cachon, Kaitlin M Daniels, and Ruben Lobel. The role of surge pricing on a service platform with self-scheduling capacity. *Manufacturing & Service Operations Management*, 19(3):368–384, 2017.

[46] Le Chen, Alan Mislove, and Christo Wilson. An empirical analysis of algorithmic pricing on amazon marketplace. In *Proceedings of the 25th international conference on World Wide Web*, pages 1339–1349, 2016.

[47] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[48] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

[49] Shivani Gupta and Atul Gupta. Dealing with noise problem in machine learning data-sets: A systematic review. *Procedia Computer Science*, 161:466–474, 2019.

[50] Dana Angluin and Philip Laird. Learning from noisy examples. *Machine learning*, 2:343–370, 1988.

[51] Noor Kamal Kaur, Usvir Kaur, and Dheerendra Singh. K-medoid clustering algorithm-a review. *Int. J. Comput. Appl. Technol*, 1(1):42–45, 2014.

[52] Alberto Durán-López, Daniel Bolaños-Martinez, Maria Bermudez-Edo, Blanca L. Delgado-Márquez, and Juan Alberto Aragon-Correa. Smart poqueira: Predicting rural parking lot feasibility with sensor-questionnaire integration, may 2024.

[53] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[54] Ian H Witten, Eibe Frank, Mark A Hall, Christopher J Pal, and Mining Data. Practical machine learning tools and techniques. In *Data mining*, volume 2, pages 403–413. Elsevier Amsterdam, The Netherlands, 2005.