**CHAPTER 2: LITERATURE REVIEW**

**2.1 Introduction to Hotel Reservation Cancellation**

Providing a warm and welcoming environment for visitors, guests, or strangers is known as hospitality. The hospitality sector is wide within the service sector and includes hotels, restaurants, bars, and other accommodations, eating, events, entertainment, travel, and tourism-related businesses as mentioned in (Shirisha et al., 2023). The cancellation of a reservation plays a significant role in the decision-making process for associated demands in the hotel management system. It influences the hotel's services and financial status and makes appropriate output predictions. As mentioned in (Shirisha et al., 2023), hotel reservation cancellations have been increasing since 2013 due to unusual reasons of the customers like illness, change of plans, better prices at other hotels, lack of facilities, etc., which are the increasing the losses to the hotel. To minimize this kind of loss, hotels have made strict policies and rules about cancellations. The fact mentioned in (Satu, Ahammed, and Abedin, 2020) say that these regulations can lower service quality, and fewer refunds/non-refund policies decrease income and reservations as well, so they aren't always effective. By providing perks, discounts, or admission to an amusement park for such reservations, hotels may lessen the effect of potential cancellations and cut down on the need for aggressive overbooking as discussed in (Shirisha et al., 2023).

Also due to the online reservation systems (ORS), many people are traveling for vacations as tourism is growing at around 4% per year as mentioned in (He et al., 2018). Due to advancements in tech and the rise in tourism across the world, ORS has been widely used in the hospitality sector. While using an ORS increases bookings for hotels, it also makes it simpler for consumers to cancel due to changes in plans. Therefore, hotels must establish explicit cancellation procedures, especially given the prevalence of ORS. Customers worry about two things when they book a hotel: the cancellation deadline and the amount they will receive in refunds if they prepay and decide to cancel. But as discussed, cancellation and refund policies will incur some sort of loss to the hotel. To get rid of this hotel reservation cancellation problem, predicting whether a hotel reservation will be cancelled or not using machine learning is of necessary requirement for many hotels in order to reduce the loss. This project is about predicting the hotel reservation cancellation provided the customer details.

**2.2 What is One-Hot Encoding**

One-hot encoding is a technique used to represent categorical data as numerical values in a machine-learning model as mentioned in (Brownlee, 2017). This is done by creating a new binary feature for each possible category and assigning a value of 1 to the feature of each sample that corresponds to its original category. For example, in this project for the “market\_segment\_type” categorical feature has five categories, then one-hot encoding would create 5 new binary features, one for each category. The feature for the category that the sample belongs to would be assigned a value of 1, and the features for the other two categories would be assigned a value of 0.

A screenshot of a computer

Description automatically generated

Fig 1: One-hot encoding example for market type segment category

Also, column names will be created for each column with respective categories like "**Column Name + Category Name**". For Example: the “market\_segment\_type” column has 5 categories namely Aviation, Complementary, Corporate, Offline, and Online. Below is how it will create the column names.

"market\_segment\_type" + "Aviation" = market\_segment\_type\_Aviation

"market\_segment\_type" + "Complementary" = market\_segment\_type\_Complementary

"market\_segment\_type" + "Corporate" = market\_segment\_type\_Corporate

"market\_segment\_type" + "Offline" = market\_segment\_type\_Offline

"market\_segment\_type" + "Online" = market\_segment\_type\_Online

**2.3 Data Imbalance**

Most frequently, class labels are distributed unevenly in real-world datasets. The class imbalance problem arises when certain classes have significantly more instances than others. The accuracy of categorization can be greatly impacted by this imbalance, especially when forecasting instances of minority classes. Addressing class imbalance is a novel problem inside the machine learning framework. Class imbalance is a widespread problem across many domains. In response to this issue, several research studies have focused on using sampling strategies to improve classification performance.

**2.3.1 How Data imbalance will impact the performance of a model**

It might be difficult in computing to deal with unbalanced data. When one class dominates another in a dataset, that class is said to be the majority class. In contrast, a class is said to be in the minority when it is represented in the dataset by fewer instances than the other class. This kind of dataset is called an imbalanced dataset. Unbalanced data is a serious problem. (Spelmen and Porkodi, 2018) said that consider a database where 90% of the data falls under the majority category and just 10% falls under the minority category. 90% accuracy is attained using a classification method that simply identifies everything as belonging to the majority class as mentioned by (Spelmen and Porkodi, 2018). However, because it entirely ignores the minority class, this doesn't accurately reflect performance. The minority class is occasionally even dismissed and considered as noise. We must balance unbalanced data in order to improve classification accuracy and provide accurate predictions.

**2.3.2 Solutions for Data Imbalance**

To deal with the data imbalance, there are data-level methods like over-sampling and under-sampling which are discussed below.

**2.3.2.1 Over Sampling**

As mentioned in (Spelmen and Porkodi, 2018) by replicating more copies of the minority class, oversampling helps balance classes, although it doesn’t bring any new information and can lead to overfitting.

Traditional oversampling has the important drawback of increasing minority class instances, but frequently in a relatively similar region of the feature space. (Chawla et al., 2002) developed SMOTE, a method that generates "synthetic" instances rather than just replicating the minority class, to overcome this issue. SMOTE is popular and frequently performs better than random or straightforward oversampling.

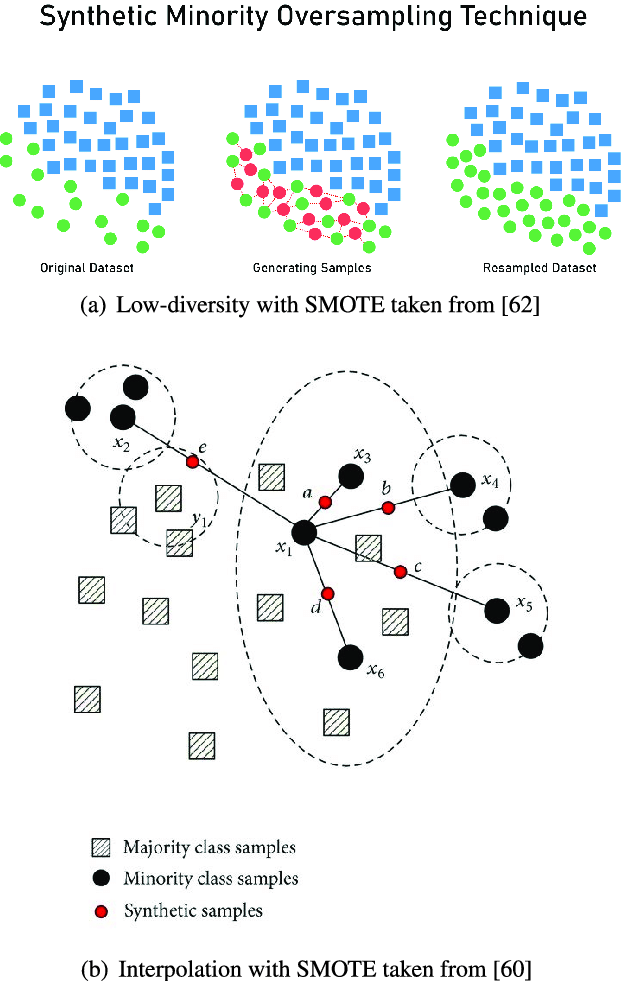


Fig 2: Data generation using SMOTE for the minority class.

[Source: (Sharma, Singh, and Chandra, 2022)]

**2.3.2.2 Under Sampling**

According to (Hasanin and Khoshgoftaar, 2018), Under-sampling the majority class removes the instances from the majority and the latter adds instances to the minority class. Random under-sampling (RUS) is based on randomly removing the majority class, but other methods selectively undersample the majority class while keeping the original population of the minority class. Tough undersampling can lead to loss of the original data from the majority class but it is preferred if people want to experiment only on the original data instead of the synthetic data generated. In this project random undersampling technique is also used to experiment on the algorithms.

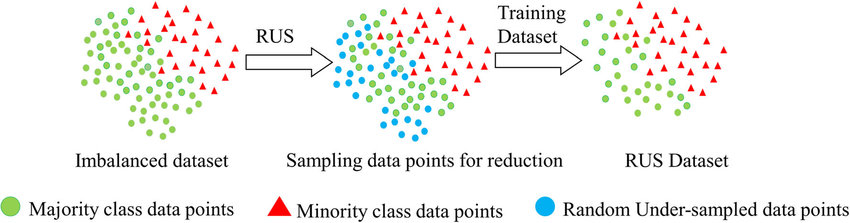


Fig 3: Data reduction using RUS from the majority class.

[Source: (Aldraimli et al., 2020)]

**2.4 Feature Scaling**

Feature Scaling is a method for ensuring that all the various characteristics in a dataset work well together. (Bhandari, 2020) mentioned that features might occasionally have inconsistent sizes or units, which can screw up a model. This is resolved via scaling, which uniformly scales all characteristics. Standardization, normalization, and min-max scaling are examples of scaling methods that modify the values while maintaining their relationships. As a result, models operate more effectively, converge more quickly, and avoid having one characteristic dominate others just because it has more data points according to (Bhandari, 2020). It makes data simpler for machines to understand.

**2.4.1 What is Standardization**

Data standardization is a preprocessing technique that modifies data such that its mean (average) is 0 and its standard deviation is 1 as mentioned in (Bhandari, 2020). All aspects may be immediately compared by making sure that each characteristic has a comparable scale. It is especially useful when working with variables that have a wide range of units. By first removing the mean and then dividing it by the standard deviation, each data point is subjected to standardization. Because of this, higher-value attributes don't have an outsized impact on the modeling process, which enhances the effectiveness of machine learning models. Below is the formula for standardization:

Standardization equation

Where µ is the mean of the column values and σ represents the standard deviation of the column values.

**2.4.2 Benefits of Standardization**

Below are the benefits of the standardizing the data:

* It ensures the equal contribution of all features for better performance of the models.
* During the training of machine learning algorithms, standardization helps the model to converge faster in the process of finding solutions.
* Standardisation strengthens models against outliers and lessens the distortion of the data distribution by reducing the impact of extreme values on the mean and standard deviation.

**2.5 Hyperparameter tuning with Grid Search Cross Validation**

Machine learning algorithms are essential in the decision-making process that is data-driven. To improve the model's internal settings, or hyperparameters, to improve model performance. GridSearchCV is one of the Python tools that can help in the process of determining the ideal hyperparameter values.

**2.5.1 What is Grid Search Cross Validation**

The ultimate goal of fine-tuning is to find the best hyperparameter values for accurate model predictions. It takes a lot of time to manually search via trial and error. Techniques like Random Search and Grid Search were devised to overcome this. Different hyperparameter combinations are thoroughly evaluated through Grid Search. Grid Search evaluates each hyperparameter combination's performance and chooses the optimal values. However, especially when there are several hyperparameters, it might be computationally costly.

As mentioned in (Shah, 2021), In GridSearchCV, cross-validation aids in model training, improving Grid Search. Instead of just splitting the data into train and test sets, cross-validation further splits the training data into "k" divisions, with one for validation and the remaining for training in each iteration. In general, model performance is determined for each iteration using K-fold Cross-Validation, and the results are averaged. This process takes a lot of time when using grid search and it requires a lot of computing, but it is necessary to find the best hyperparameters.

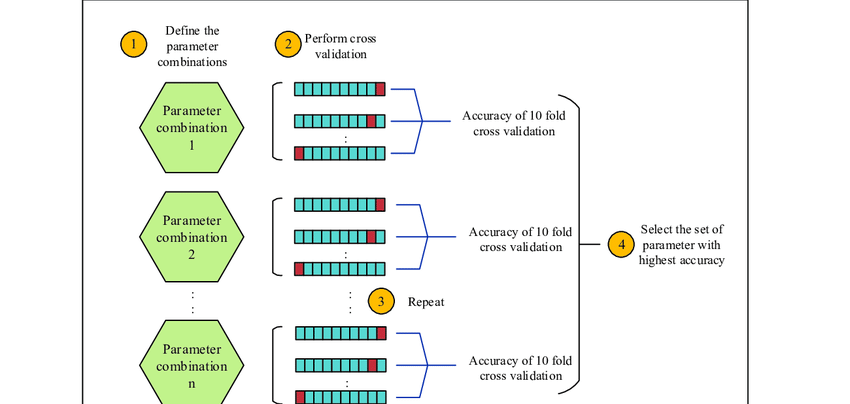


Fig 4: Grid Search Cross Validation Operation

[Source: (Nurul Liyana Hairuddin, Lizawati Mi Yusuf and Mohd Shahizan Othman, 2020)]

**2.5.2 Advantages and Disadvantages of Grid Search CV**

**Advantages:**

* A lot of time will be saved using GridSearchCV, from doing manual trial and error to find the best hyper-parameters (s, 2021).
* GridSearchCV is flexible and it has its own parameters that can be changed to meet our goals.

**Disadvantages:**

* If the computational resources are poor and there are a lot of hyper-parameters, it will cost time as mentioned in (s, 2021).
* The primary goal of GridSearchCV is to decrease bias error, however, it may neglect variance error, thus creating high variance problems (s, 2021).

**2.6 Overview of Machine Learning Classification**

Machine learning is defined as the subset of artificial intelligence that gives the ability to the computer to learn and improve from the experience without being explicitly programmed. Machine learning is divided into 3 categories namely, supervised learning which deals with classification and regression problems with labels. Meanwhile, unsupervised learning deals with the data without labels and the semi-supervised learning idea is to generalize better on large unlabelled data using a small amount of labeled data. In this project, we are dealing with a supervised classification task.

**2.6.1 Logistic Regression**

Modeling the probability that a certain class or event will occur using the supervised machine learning approach is known as logistic regression. It becomes relevant when there is linear separability in the data and there are two separate categories for the desired result. Essentially, binary classification problems, where the objective is to predict an output variable that belongs to one of two discrete groups, are the main applications of logistic regression as mentioned in (Bonthu, 2021). Logistic regression is mostly employed to answer Yes or No, Cancelled or Not Cancelled, Win or Lose, etc. The output is a probability value between 0 and 1.

Multinomial logistic regression is also available which comes into the picture if the output variable consists of three or more classes without natural ordering as mentioned in (Bonthu, 2021).

The equation for the logistic regression is

=

Similarly, with ‘n’ predictors the logistic regression equation is as below.

Where *Xn* is the nth observation of X and [w1, w2, w3, ….. , wn] are the weights and b is the bias term.

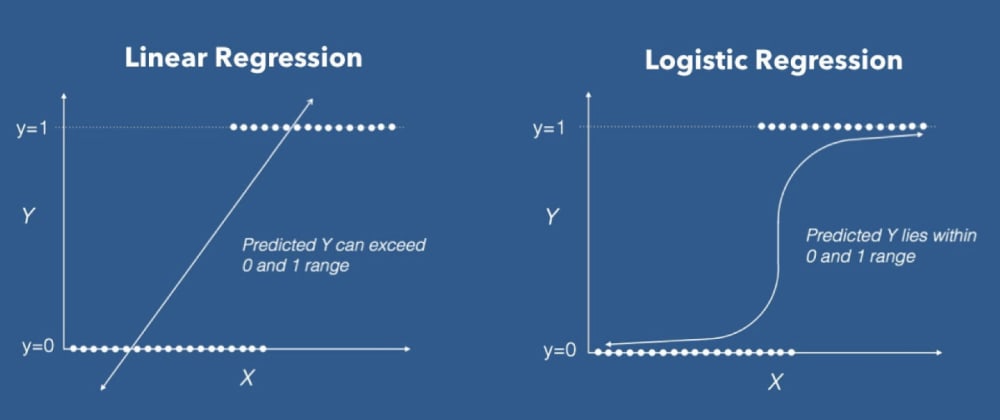


Fig 5: Logistic Regression [Source: (Bonthu, 2021)]

**2.6.2 K-Nearest Neighbours**

K-Nearest Neighbours (KNN) is a supervised learning non-parametric classification algorithm that is famous for its simplicity and efficiency as mentioned in (Taunk et al., 2019). Data is classified by KNN by looking at its near neighbors from the training dataset within a predefined area. This approach is favored because of its simple implementation and low processing demands. It uses Euclidean distance to calculate closeness for continuous data. As mentioned by (Taunk et al., 2019) KNN determines the K closest neighbors for a new input and classifies it according to the dominant class among these neighbors. Although this classifier is simple, the choice of the 'K' value has a big impact on how unlabeled data is classified.

Below are the steps for KNN working according to the points mentioned in (Taunk et al., 2019):

{ (x(1), y(1)) , (x(2), y(2)), …… , (x(m), y(m)) }

Step 1: Initially, training set given has to be stored.

Step 2: Now for each new unlabeled data, Euclidean distance has to be calculated for all the training data using the formula:

“Find the k-nearest neighbours with the specified K-value for the new datapoint and assign the class containing the maximum number of nearest neighbours” as mentioned in (Tanuk et al., 2019).

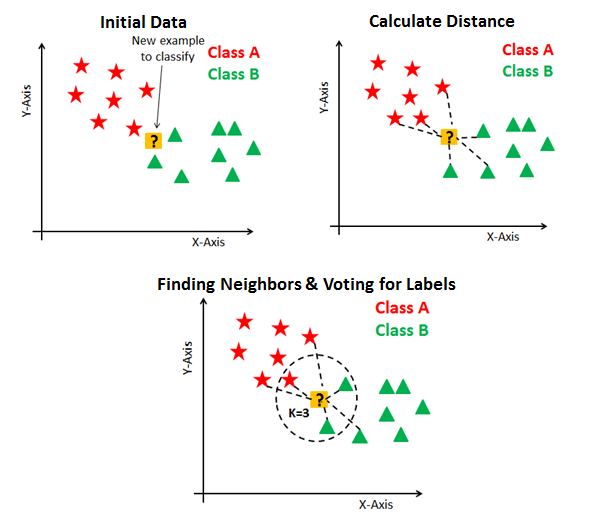


Fig 6: In-depth intuition of KNN Algorithm Working

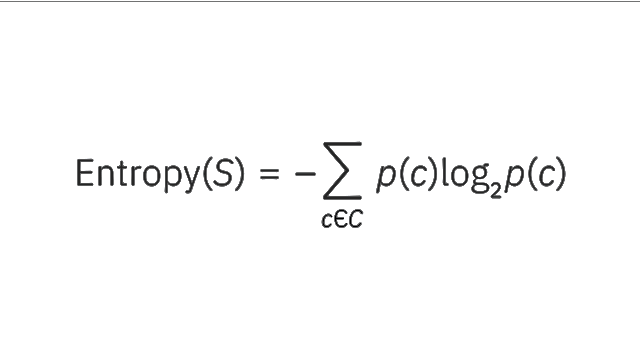
[Source: (S, 2020)]

**2.6.3 Decision Trees**

Similar to flowcharts, decision trees are graphic representations that may be used to categorize and sort data. A decision tree is a very important algorithm in supervised learning. There are two types: classification trees for categories and regression trees for continuous data. These trees operate by recursively separating data in a "top-down" manner, breaking data down incrementally. They continue separating until the data is comparable enough to satisfy user needs.

The thing to note here is excessive splitting might result in overfitting, where the data is too much categorized. This can be solved by pruning the tree. Though there are various ways to find the best feature at each node, the two methods namely information gain and entropy are the popular splitting techniques for decision tree models as mentioned in (IBM, 2022).

Knowing entropy is useful for understanding information gain. Entropy is a metric for assessing how disorganized a dataset is, with values ranging from well-sorted (0) to complete confusion (1). In order to create a successful decision tree, we try to select features that reduce entropy, or how chaotic things get. We select the traits with the lowest entropy in order to construct the best decision tree.

****

* S stands for the collection of data from which entropy is calculated.
* The classes in set S are represented by c.
* The percentage of data points in a collection that are members of class C is shown by the symbol p(c).

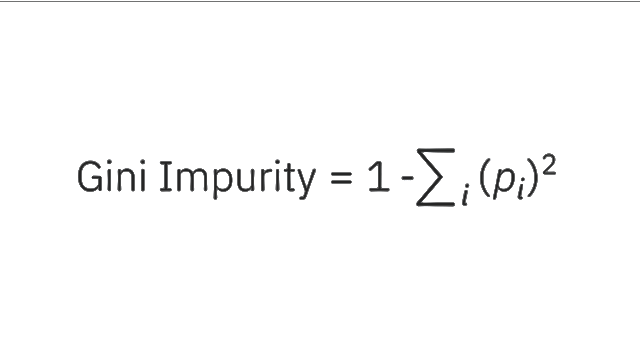
Entropy before and after a split on a particular property are different, and this difference is represented by information gain. Because it performs the best at categorising the training data in accordance with its goal classification, the attribute with the maximum information gain will result in the best split.

A screenshot of a computer

Description automatically generated

* ‘a’ stands for a specific class label
* Entropy(S) represents the entropy of dataset S.
* “|Sv|/ |S| represents the proportion of the values in Sv to the number of values in dataset, S” (IBM, 2022).
* “Entropy(Sv) is the entropy of dataset, Sv” (IBM, 2022).

Gini impurity is the likelihood that a randomly chosen data point in the dataset would be wrongly classified if it were labeled using the dataset's class distribution. Similar to entropy, the impurity of a set, S, if it is pure (belonging to one class), is zero. This is shown by the formula:



**2.6.4 Random Forest**

Random Forest is a very powerful supervised machine learning algorithm and is most frequently used for both classification and regression tasks. Random Forest operation is very simple since we know the operation of the decision tree in the above section.

As mentioned in (E R, 2021), “Random Forest builds decision trees on various data points and takes the majority vote in case of classification and in case of regression, it will take average. The random forest method produces a consolidated and more accurate result by integrating the outputs of various trees.

The Process of the Random Forest Algorithm

**Step 1:** Each decision tree in the Random forest model is built using a subset of attributes and a subset of data points. In simple terms, m features and n randomly chosen records are obtained from a data collection of k records.

**Step 2:** For each sample, a unique decision tree is built.

**Step 3:** An output will be produced by each decision tree.

**Step 4:** For classification and regression, the final result is evaluated using the majority vote or averaging.

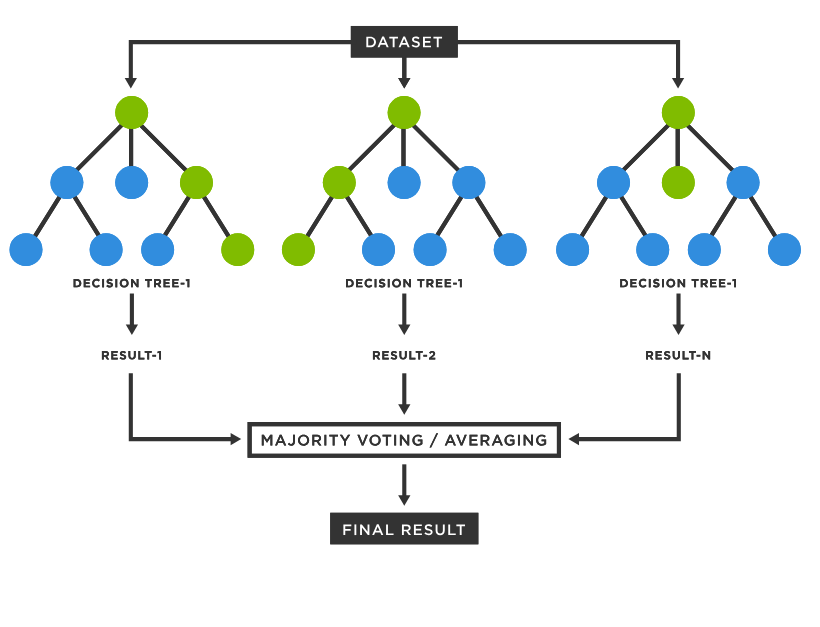


Fig 7: Random Forest Representation

[Source: (TIBCO, 2023)]

**2.6.4.1 Random Forest Feature Importance**

Feature importance is the key concept in machine learning as it tells, which features are most predictive of the target variable (most important) and tells the least important features to safely ignore them. “The feature importance is measured using **average impurity decrease** which is computed from all the decision trees” as mentioned in (Data Analytics, 2020).

**2.7 Research Gap**

The dataset for this research is imbalanced. As everyone knows, oversampling and undersampling techniques will help to get rid of imbalance issues. There are many papers dealing with sampling techniques individually on the machine learning algorithms. Hence the research gap identified is to know whether over-sampled data or under-sampled data gives the best performance on machine learning algorithms on predicting the hotel reservation cancellation. So, the machine learning algorithms will be applied on over-sampled and under-sampled data and the results will be compared to know which algorithm has given the best performance in predicting whether the customer will cancel the hotel reservation or not based on the details provided.