UNIVERSITY OF
PATRAS
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

BIG DATA MANAGEMENT

2st Academic Year Assignment 2023-2024, due 26/11/2023

Collaborators
Georgakis Giorgos, 1052995
Benekos Konstantinos, 1067754
Servos Dimitrios, 1067563
Petroulis Veliassarios, 1074579

Contents

1 Task 2 **3**
1.1 Paper 1 . . . 3
1.2 Paper 2 . . . 4
1.3 Paper 3 . . . 5
1.4 Paper 4 . . . 6
1.5 Paper 5 . . . 9

2 Task 3 **11**
2.1 I) . . . 11

3 Task 4 **13**
3.1 I) . . . 13
3.2 II . . . 15
3.3 III . . . 15

4 References **16**
4.1 data . . . 16
4.2 Theoretical basis . . . 16
4.3 Statistical tools . . . 16

1 Task 2

1.1 Paper 1

The aim of this study is to assess the impact of information mediation systems on the hospitality industry by analyzing user-generated content, particularly online reviews, to understand travelers' satisfaction with hotels. It basically aims at defying What defines a good or bad hotel? And what are the key concerns, features and supplementary services that are more likely to get a hospitality business a higher rating.

The data used in the study is 47,172 hotel reviews from the 9th Yelp Dataset Challenge, concentrating on Las Vegas, from March 2005 to January 2017. By using a binary classification of 1-star and 5-star ratings, the analysis focused on extreme satisfaction and dissatisfaction scenarios. Language consistency was ensured through the textcat package in R.

The study introduces a classification strategy applicable to businesses beyond hotels, such as movie theaters, by processing large amounts of review data through supervised machine learning. First a Maximum Relevance Minimum Redundancy (MRMR) approach is utilized to identify the most relevant data by selecting features that are highly pertinent to the classification task while minimizing redundancy, discarding terms that were strongly associated with other terms but weakly associated with star-ratings (indicating low predictiveness). The decision of selecting which features to keep and which to discard is based on the rates of change of their Matthews correlation coefficient (MCC), which considers true and false positives and negatives in binary classifications. Subsequently, the researchers employ the Naive Bayes classification algorithm to categorize and analyze the identified features, aiming to discern patterns in the data and enhance the precision of the classification process. Multiple Naive Bayes models were created by iteratively increasing the number of sorted features. The researchers tested each model using 10-fold cross-validation, estimating MCC values to assess classification performance and the model with the maximum MCC value was selected. Features were chosen based on the rates of change of MCC, excluding features with negative rates of MCC change. The researchers assessed the performance of their approach using various metrics such as accuracy, precision, recall, F1 score, Kappa, and area under the curve (AUC).

The research done, offers innovative insights by combining knowledge extraction from text reviews, applying supervised learning, and addressing the underexplored joint analysis of non-structured reviews and classification algorithms in the context of hospitality experiences. Doing this it succeeds in tackling the challenge of handling a massive number of reviews and provides a managerial contribution by summarizing hospitality service reviews effectively, hence enhancing market transparency. The author emphasizes the potential influence of online reviews on guests' decision-making processes and underscores the managerial contribution of the classification strategy in summarizing and utilizing vast amounts of review data.

The findings reveal the effectiveness of the applied methodology, giving emphasis to the significance of extracting knowledge from text reviews and their impact on non-economic satisfaction, an area that has been relatively underexplored in existing literature. Since the study identified key elements influencing guest's perception it provides valuable insight

for hotels to allocate their resources to better meet the clients' priorities, achieving better customer satisfaction, and subsequently providing an efficient way to attract guests, thereby enhancing long-term online relationships and genuine customer loyalty. Future researchers are encouraged to continue this analytical quest by evaluating non-parametric aspects of customer satisfaction/loyalty, like the additional effects of affective trust and the overall psychological sentiments influencing attitudes for continued relationships. Furthermore, it is also underlined that another significant feature worth of delving into, in future research, are the gender-based differences in how individuals structure their hotel experiences. Overall, the study provides a practical framework for summarizing and extracting insights from hospitality service reviews to enhance decision-making in the industry.

1.2 Paper 2

In today's era, there is a large number of websites where people can access and share their opinions about any entity. Since opinions are numerous and cannot be examined individually by a person, there is a need for a suitable reputation measure that is reliable and can be taken into consideration by individuals. This article aims at a new approach to improve this reputation index.

The above article proposes a new approach to the problem by combining Naïve Bayes and LSVM classifiers to separate positive from negative comments. It suggests using both classification algorithms mentioned earlier to categorize reviews as positive or negative before grouping them into various sets based on semantic relations. The primary classifier used is Naïve Bayes due to its higher accuracy, as found in previous works by Shawe-Taylor and Sun (2011) and Pang, Lee, and Vaithyanathan (2002). Subsequently, SVM is used to match the results in cases where Naïve Bayes predicts a negative review but the rating is above 5 stars, and vice versa. To overcome the weakness of the LSA model in extracting opinions with the same perspective, the article introduces a classification step before the grouping phase. This step ensures that positive opinions are grouped together and vice versa, addressing the limitation observed by Yan, Jing, and Pedrycz (2017) in the LSA model's ability to perform effective separation when opinions in each set hold a similar or same perspective. Finally, the article proposes calculating the Weighted Arithmetic Mean separately for positive and negative groups, as the number of positive and negative reviews may not always be the same.

As mentioned earlier, we utilized the Naïve Bayes and Linear Support Vector Machine (LSVM) classifiers, both trained on a dataset comprising movie reviews (polarity dataset v2.0)2, which includes 1000 positive and 1000 negative processed reviews. A dataset was created to test the validity of our models, consisting of 10 datasets for 10 different movies, each containing 100 reviews [comment + rating + manually annotated sentiment polarity], randomly selected. Care was taken to ensure that the datasets are representative based on the IMDb users' weighted average vote.

Based on the sample, it appears that the accuracy achieved with the combined grouping of Naïve Bayes and SVM surpasses that of using each individually. It should be noted that by accuracy, we mean which reviews were correctly categorized as positive or negative. The creation of reputation involves clustering positive and negative reviews separately. Addi-

tionally, by using the weighted average, we take into account that positive and negative reviews are not always equal. It seems to provide results closer to reality compared to what Yan, Jing, and Pedrycz (2017) constructed. Additionally, a sub-analysis was conducted within the scope of the analysis. To determine the suitable value of t_0 to achieve the highest accuracy in relation to the real world. t_0 signifies the degree of similarity that reviews must have to be grouped into the same set, for example, positive or negative comments in this case. They found that the results remain stable when $t_0 \leq 0.6$ and the best results, those closer to reality, are achieved when $t_0 = 0.95$. We conclude that it is crucial to choose the appropriate value for t_0 to ensure the reliability of our results.

In this paper, we propose an approach for reputation generation using the combination of two classifiers, Naive Bayes and Linear Support Vector Machine (LSVM). These classifiers are used to categorize reviews as either positive or negative with minimal error using text analysis. Additionally, we suggest computing the reputation value using the Weighted Arithmetic Mean, improving upon the work presented by Yan, Jing, and Pedrycz (2017) by enhancing the accuracy of reputation values for various movies. It is evident that a classifier depends on both language and the domain of use, making it somewhat limited, regardless of how effective it may be for the given sample to categorize using text in the specific case. Fortunately, there are methods available that can address this issue to some extent.

It was observed that many reviews were irrelevant to the target entity. As a result, an intermediate step was proposed to remove such reviews, avoiding unnecessary operations and improving accuracy.

1.3 Paper 3

The article titled "The Determinants of Bitcoin's Price: Utilization of GARCH and Machine Learning Approaches" discusses the factors influencing the value of Bitcoin by examining various models to predict fluctuations. Initially, Bitcoin is a digital currency created in 2009 by an individual using the pseudonym Satoshi Nakamoto. It is a form of digital money that utilizes cryptography to control the creation of new units, manage transactions, and verify money transfers. One of the predictive models used is GARCH, an advanced autoregressive conditional heteroskedasticity model, employed to explain how Bitcoins, considered as the dependent variable, are affected.

Subsequently, Support Vector Machine (SVM) is used for supervision and prediction, categorizing data into two classes and seeking the optimal margin boundary. The third model is a decision tree used for data mining and classification, identifying significant factors used as decision indicators. Data for these three models were collected from various sources such as bitcoin.org, investing.com, FRED, and the World Gold Council website, covering time periods from July 2010 to December 2018, with a sample of 1955 observations.

GARCH uses unit root testing to process non-stationary data, while SVM employs independent variables with time lag. Rejecting the null hypothesis of a unit root indicates that the processed data is stationary, allowing for continued empirical analysis. This study utilizes the ARCH-LM test to check the sequence of the series and the Akaike Information Criterion (AIC) to evaluate each model and select the best one. The SVM model then demonstrates high prediction accuracy for the Bitcoin price, approximately 80.61%. The

article concludes that the development of Bitcoin combines research on influencing factors, applying three models for a better understanding of parameters affecting market value and Bitcoin transaction volume. It suggests that SVM has the highest accuracy for Bitcoin price, as it can minimize the generalization error boundary, unaffected by overfitting issues, and can achieve high generalization performance.

1.4 Paper 4

The article delves into employing a Naive Bayes model for predicting recessions. It outlines that as the economy reaches a stall phase, there's an increased likelihood of a subsequent recession. Comparisons are drawn between the Naive Bayes approach and other methodologies like logistic regression and Markov-switching models, highlighting the superior performance of Naive Bayes in recession forecasting. The discussion extends to leveraging real-time data and integrating diverse explanatory variables within the Naive Bayes model. Ultimately, the article concludes that Naive Bayes achieves its asymptotic error rate quicker than logistic regression, culminating in more precise recession predictions despite limited data availability.

The aim of the work discussed in the article is to demonstrate the use of a Naive Bayes model as a tool for forecasting recessions. The researchers explore the application of Bayesian classification techniques to predict whether the economy will enter a recession in the future. They discuss the importance of using real-time data and how it can improve the accuracy of detecting the onset of recessions. The researchers also compare the Naive Bayes approach with other methods, such as logistic regression and Markov-switching models, which are commonly used for business cycle forecasting. They highlight the advantages of the Naive Bayes model, including its ability to handle a large amount of data with a rich lag structure and capture the persistence of business cycle phases through Markov-switching transition probabilities. Additionally, the researchers discuss the limitations and assumptions of the Naive Bayes model, such as the assumption of variable independence, which may not always hold in real-world applications. They acknowledge that violating this assumption can undermine the accuracy of the model. However, they argue that the Naive Bayes approach still outperforms other models, such as logistic regression, in the context of predicting recessions. Overall, the aim of the work is to present the Naive Bayes model as a valuable tool for recession forecasting and to demonstrate its superiority over other methods in terms of accuracy and predictive ability.

The adopted approach of using Bayesian Classification for recession forecasting does present something innovative. The traditional approach to forecasting recessions has often relied on binary response models or Markov-switching models. However, the Naive Bayes (NB) model introduced in this study offers a different and potentially more accurate approach to recession forecasting. One innovative aspect of the NB model is its ability to incorporate a larger amount of data with a rich lag structure. By incorporating multiple variables and their conditional probabilities, the NB model can capture the complexity and interdependence of economic indicators. This is in contrast to other models that may only focus on a limited set of data, such as GDP. Additionally, the NB model treats National Bureau of Economic Research (NBER) business cycle turning points as data, rather than hidden states to be inferred by the model. This allows for a more explicit consideration of

the NBER-defined recessions and expansions in the forecasting process. Furthermore, the NB model's asymptotic properties provide some intuition as to why it outperforms other models in a business cycle forecasting context. The NB model reaches its asymptotic error rate much faster than logistic regression (LR), resulting in more accurate recession forecasts with limited data. This is particularly relevant in the context of recessions, which are relatively infrequent events. Overall, the adoption of the NB model for recession forecasting presents an innovative approach that incorporates a larger amount of data, treats NBER turning points as data points, and offers increased accuracy in forecasting recessions.

The machine learning models/algorithms that have been used in this context are Naive Bayes (NB), logistic regression (LR), and Markov-switching models. Naive Bayes is used as a recession forecasting tool because it can incorporate a large amount of data with a rich lag structure and has been shown to outperform logistic regression in a business cycle forecasting context. NB also treats National Bureau of Economic Research (NBER) business cycle turning points as data, rather than hidden states to be inferred by the model. Logistic regression is also used for recession forecasting, but it is shown to have slower convergence to the error rate compared to Naive Bayes. However, logistic regression can still be useful when certain distributional assumptions hold, but these assumptions do not hold for business cycle data. Markov-switching models are used for nowcasting and classification of business cycle states. These models are useful for identifying turning points similar to NBER dates and can assign a probability that the economy is in any number of predefined states. However, they are limited in their ability to forecast a recession several periods into the future.

The datasets used in the context information include macroeconomic data and financial indicators. The specific variables within these datasets are not explicitly mentioned, but some examples of variables that are commonly used for recession forecasting include nonfarm payroll growth, the Institute of Supply Managements (ISM) Manufacturing Index, the change in the S&P 500, and the term spread (the difference between yields on the 10-year and 2-year Treasury securities). Additionally, the context information mentions the use of real-time data and revised data for evaluating forecasting performance, particularly in relation to nonfarm payroll data. Therefore, the datasets used likely contain a range of economic and financial variables that are relevant for predicting recessions and assessing the state of the business cycle.

In the context of recession forecasting, the evaluation of models or algorithms is crucial to determine their accuracy and effectiveness. One common evaluation metric used is the F-measure under a zero-one loss, which measures the model's precision and recall. Precision refers to the ratio of true positives (correctly predicted recessions) to the total number of positive predictions, while recall refers to the ratio of true positives to the total number of actual recessions. The Naïve Bayes (NB) model, which is used in the context of this text, is evaluated based on its ability to correctly predict recessions. The NB algorithm is a probabilistic model that makes use of the Bayes' theorem to estimate the probability of a recession given certain input variables. The accuracy of the NB model is assessed by comparing its predictions with the actual occurrences of recessions. The NB model is compared to other models, such as logistic regression (LR) and Markov-switching models, to determine its relative performance. In the case of business cycle data, the assumptions underlying the equivalence between NB and LR do not hold, making the NB model more accurate and effective in predicting recessions. To evaluate the performance of these models,

various criteria are used. One criterion is the ability to correctly identify past recessions without relying on the National Bureau of Economic Research (NBER) turning points, which are considered as the reference for recession dates. Another criterion is the ability to nowcast or classify the current state of the business cycle accurately. Markov-switching models and binary response frameworks are commonly used for such evaluations. Furthermore, the evaluation of these models takes into account the timeliness of the predictions. Models that are able to provide early signals or warnings of an impending recession are considered more valuable. This is because the lag between the onset of a recession and the signal from the model is typically several months, and early predictions can allow for timely policy responses and mitigating actions. Overall, the evaluation of models or algorithms used in recession forecasting involves comparing their predictions with the actual occurrences of recessions based on various criteria, such as precision, recall, timeliness, and ability to correctly identify previous recessions. The NB model, in particular, has been shown to outperform other models in terms of these criteria, making it a valuable tool for recession forecasting.

The main findings of this paper are that the Naive Bayes (NB) model is a useful tool for forecasting recessions and that it outperforms logistic regression (LR) in a business cycle forecasting context. The paper demonstrates the use of NB as a recession forecasting tool and also focuses on nowcasting and classification of business cycle states. The paper explains that when the economy enters the stall phase, it is more likely to enter a recession in the subsequent periods. Therefore, predicting future real GDP growth and recessions is valuable information. The authors use Bayes theorem for k classes to predict recessions. They assume conditional independence of the variables and decompose the joint probability term as the product of the conditional probabilities. However, the assumption of variable independence is often violated in the business cycle context, which undermines the utility of NB. Nonetheless, the paper argues that NB is considerably better than LR in terms of recession forecasting accuracy under a range of criteria. The paper also discusses the connection between the NB approach and Markov-switching models and logistic regression. The NB framework can easily incorporate a large amount of data with a rich lag structure and captures the persistence of business cycle phases by using Markov-switching transition probabilities. In contrast, LR treats the National Bureau of Economic Research (NBER) business cycle turning points as hidden states to be inferred by the model. The paper emphasizes the importance of using real-time data within a Markov-switching model for recession forecasting. In terms of forecasting performance, the paper uses the F-measure under a zero-one loss to evaluate the accuracy of NB and LR. It finds that NB reaches its asymptotic error rate much faster than LR, resulting in more accurate recession forecasts with limited data. The paper also discusses the limitations of the NB approach, such as the assumption of variable independence and the need for accurate real-time data. Overall, the main findings of the paper suggest that the NB model is a valuable tool for recession forecasting and that it outperforms LR in a business cycle forecasting context. The paper highlights the importance of using real-time data and capturing the persistence of business cycle phases for accurate recession predictions.

This research concludes that a Naive Bayes (NB) model can be an effective tool for predicting recessions. The study compares NB with Logistic Regression (LR) and shows that NB outperforms LR in terms of recession forecasting accuracy. The NB model is based on

Bayesian classification, where the probability of a recession given certain data is calculated using Bayes' theorem. The study also highlights the importance of using real-time data in Markov-switching models for detecting the onset of recessions. The research provides insights into the strengths and limitations of different approaches to recession forecasting. While Markov-switching models have been successful in identifying turning points similar to the NBER dates, they may not be as effective in forecasting recessions several periods into the future. On the other hand, NB shows promise in accurately predicting recessions, especially when considering a rich lag structure and incorporating the persistence of business cycle phases using Markov-switching transition probabilities. The study emphasizes the importance of conditional independence assumptions in NB, as violation of these assumptions can undermine the utility of the model. However, for business cycle data, NB performs considerably better compared to LR under a range of criteria. NB also reaches its asymptotic error rate faster than LR, resulting in more accurate recession forecasts with limited data. Overall, the research suggests that NB can be a valuable tool for recession forecasting, particularly when considering real-time data and incorporating the dynamics of business cycle phases. Using the NB model can improve the accuracy of recession predictions, providing valuable insights for policymakers and decision-makers in preparing for economic downturns.

1.5 Paper 5

The study addresses the issue of fake reviews on the internet and introduces the Naive Bayes algorithm as an effective method for their detection. The Naive Bayes algorithm, rooted in the probability theorem of Bayes, proves efficient for classifying reviews as genuine or fake, since it is simple quickly trained and effective in analyzing text data which is the main format of reviews. For it to be implemented on an unknown data set it needs to be trained on a dataset of known real and fake reviews, learning the characteristics distinguishing between the two. Once trained, the algorithm can assess new reviews based on these learned characteristics.

The author does acknowledge, that Naive Bayes may not detect all fake reviews, and challenges such as data cleaning, normalization, missing data, and outliers must be addressed in pre-processing.

Also, the study, supplementary, uses the Support Vector Machine (SVM) algorithm for classification and regression analysis due to its accuracy and efficiency, especially in handling noisy and high-dimensional data.

The classifier model, in the study, is trained on 20 thousand reviews from Kaggle, focusing on star ratings and utilizing features like length, language, and emotive expression. Data cleaning involves removing unnecessary words, stop words, and punctuation, followed by vectorization to convert data into vectors. The model is then split into training and test data, with testing involving 2000 data points. Balancing the dataset is also emphasized for improved model accuracy in detecting fraudulent reviews.

The article then goes on to describe how it the dataset is "cleaned", for the data of interest (in this case the fake vs the real reviews) to be mined. First the phase of data extraction, can be carried out at the phrase, sentence, or document level using both supervised

(Utilizes the Naive Bayes (NB) algorithm) and unsupervised approaches. As the analysis does not consider the information from the current reviews, the proposed system employs an unsupervised method. Next, stop words are removed from the sample with the purpose of simplifying the analysis and enhancing the efficiency of subsequent processing steps. The following step is stemming, a technique for separating words into basic forms by removing affixes, making retrieval more accurate. Again, by reducing the size of the index, the process of retrieval and analysis becomes more accurate. Subsequentially the paper continuous with post tagging, a method which uses Natural Language Processing (NLP) to label words in a sentence based on their parts of speech with the goal of enhancing the understanding of sentence structure and contributes to subsequent analysis. Finally, the negative and positive datasets are compared to the Part-of-Speech (POS) tagging output, this generates positive and negative values, and identifies intermediate values if positive and negative values are the same.

The study focuses on detecting fake reviews, presenting an approach that considers both the characteristics of reviews and the behavioral features of reviewers. The results indicate that the Naive Bayes (NB) classifier outperforms other classifiers in detecting fake reviews, and considering reviewers' behavior enhances the f-score (a metric used in binary classification tasks, such as identifying fake reviews in this context. It combines precision and recall into a single value to provide a balanced evaluation of a model's performance). The innovation of the study lies in that it introduces a method for fake review detection that incorporates both review characteristics and reviewer behavior using NB classifier and the consideration of behavioral features. The study suggests for further research, including exploring additional behavioral attributes.

2 Task 3

2.1 I)

In this part of the assignment we present the results of the analysis of our text by showcasing, and explaining in depth

II)

In this section of the assignment, we will delve into the intricacies of sentiment analysis utilizing a Naive Bayes classifier. To enhance our comprehension of its application, we will explore a theoretical scenario in-depth. The objective is to provide a practical example that illustrates the implementation of sentiment analysis, offering a detailed insight into how the Naive Bayes classifier is employed and its significance in real-world scenarios. This theoretical scenario aims to serve as a practical illustration, shedding light on the nuanced aspects of its implementation and elucidating its utility in analyzing sentiment in various contexts.

We will start by first analyzing the theoretical framework behind the statistical tools used in our code to better establish the baseline of our analysis. The first step is the data preprocessing, since the data both in above exercise as well as in the example, which will be presented, is in text form, this step is crucial. We start by employing string manipulation techniques to clean and standardize the text data, this involves removing special characters and converting the text to lowercase. These steps ensure uniformity and simplicity in our text making the subsequent analysis easier. Next, we continue by removing stop words (common words that often don't contribute much to the sentiment) this is done using a library which holds all the most common stop words in English. We thus eliminate all common attributes(words) found in both our text and this library. The final step in preprocessing is stemming, which includes reducing words to their root form, aiding in the simplification of the text data, and capturing the essence of words.

Next, is the creation of the document-term matrix (DTM), this matrix represents the frequency of terms in each document (review). This transformation allows for the quantitative representation of textual data, enabling statistical analysis. Basically, it is matrix whose columns each represent one single word and it takes values based on how frequent each of these words can be observed in the given text. This is crucial since it gives us way to quantify, in numbers, how often or if certain words are used. This leads us to our next step, in which we perform term frequency filtering: identifying terms that occur at least five time. This process helps us to identify terms that are relatively frequent and hence are probably meaningful/informative to our analysis (the number 5 was chosen by rule of thumb, it would be adjusted depending on what the analysis goal is, on how much you want to compress your data/how thorough you want your analysis to be and most importantly on the text data itself). Finally, we eliminate sparse data, meaning that terms, which occur infrequently in documents, are removed to address computational efficiency and memory utilization. This step involves setting a threshold for sparsity, ensuring that only terms with a certain level of prevalence are retained for analysis (in our case that threshold is set at 99.745%-meaning that terms that occur in less than 0.255% (1 - 0.99745) of the documents will be removed). It is important to note the significance of this step, since in the DTM,

many entries might be zero because not every term appears in every document. Sparse matrices are often encountered in text data due to the vast vocabulary and the fact that each document typically contains only a subset of the terms.

Moving on, we continue by building a data frame out of the DTM. The conversion of the term-document matrix to a dense format and its combination with sentiment labels in a data frame facilitate the subsequent modeling process. This structured data format is essential for training and evaluating the performance of machine learning models. Next, we split the data set contained in the resulting data frame, into a test (20%) and a training (80%) set, using random sampling. This approach ensures a representative subset of data for training the model while maintaining a separate portion for evaluating its performance.

Now that are data is preprocessed and stored in way that facilitates sentiment analysis, we employ the Naïve Bayes classifier. This classifier is based on Bayes' theorem and assumes independence between features. It calculates the probability of a review belonging to a particular sentiment class, given the observed words in the review. In other word based on how frequent certain words appear in a text it is classifies the review as positive or negative (or for that matter any other type of classification we want). We "train" the classifier on our training set and test in on the testing set, and then we compute the confusion matrix to conclude the accuracy of our classifier. This matrix compares the predicted sentiment labels with the actual labels in the testing set. Accuracy is then computed as the ratio of correctly predicted instances to the total instances (True Positive, True Negative, False Positive, False Negative) providing a statistical measure of the model's performance in predicting sentiment. Finally, we obtain the accuracy of the classifier constructed using Naive Bayes. This accuracy is calculated as the ratio of the elements on the main diagonal of the confusion matrix to all the elements in the matrix. In our case, it resulted in 0.7636, meaning that approximately 76% of the predictions based on their features were indeed in the category assigned to them.

Now for our example-scenario, "Scenario: Evaluating the Impact of User Comments on the Price of a Specific Product in a Supermarket". We start by collecting our data, for this user comments from social media would be collected, specifically those discussing a particular product in a supermarket, the data would include the text of the comments and any relevant labels, such as the price of the product. Next, we would start with text preprocessing, as mentioned above, this process would include including converting to lowercase, removing special characters, and applying stemming. The remaining text would be analyzed and a term-document matrix representing the frequency of terms in each comment would be created. The TDM would be compressed using term frequency filtering and by eliminating sparse terms (this is common practice since DTM's tend to be pretty sparce (lots of zeros)). After converting the DTM into a data frame we would split of a portion n of the data to use for training the Naïve Bayes to classify the reviews into positive and negative (or as many classes as deemed necessary). After testing the accuracy of the classifier, we would continue by correlating the resulting sentiments with the price of the particular product. We can once again use the Naïve Bayes classifier to predict, based on the sentiments how the price of the product would respond(in other words, we could predict, based on how negative/positive a review is, how the price would respond by classifying it into rising, falling etc. It is important to note that those classes can take more specific parameter's like for example rising>5%etc.). It is important to address that having more classes will

make our predictions on price more exact but will come at a computational cost as well as a possible loss in accuracy of the classifier itself. A more realistic and likely feasible approach, after perfecting the classification process based on Naive Bayes, involves dividing our data according to the sentiment of the reviews into two categories: 0 for negative reviews and 1 for positive reviews. Subsequently, we could create a categorical variable representing these sentiments. Next, we could perform a regression analysis with the dependent variable being the product price and the independent variable being the categorical variable we constructed from the Naive Bayes process. This would allow us to investigate if there is any statistically significant relationship between the two variables. We would use p-values for the coefficients and the F-statistic to assess the overall significance of our model, thereby ensuring that our estimations are unbiased.

3 Task 4

3.1 I)

In this part of the assignment we build and put to use a decision tree classifier, using the "Mushroom data set", that classifies mushrooms species into poisonous or edible based on the given attributes. A decision tree is a supervised machine learning algorithm used for both classification and regression tasks.It is considered a supervised learning method because it requires a labeled data set for training. In supervised learning, the algorithm learns a mapping from input features to corresponding output labels by being provided with a set of input-output pairs during training. A decision tree consists of a **root node:** The topmost node of the tree, representing the entire dataset.**Internal Nodes:**Nodes that represent a decision based on the value of a certain feature. These nodes lead to child nodes.**Leaf Nodes:**Terminal nodes that represent the final decision or outcome. Each leaf node is associated with a class label (in classification) or a numerical value (in regression). **Branches:**The edges or branches connecting nodes represent the decision rules based on the feature values.

The process of building a decision tree consists of first using a classifier which at each node selects the best feature to split the data based on criteria such as in our model as Gini impurity, entropy gain, or variance reduction(in case of regression). The dataset is then split into subsets based on the chosen feature and a threshold. This process is repeated recursively. Overall we are looking for maximum possible homogeneity inside each node. Finally When a stopping criterion is met (e.g., maximum depth reached or minimum samples in a node), a leaf node is created, representing the final decision.

The Gini index is an impurity measure used in decision tree algorithms, it measures the probability of incorrectly classifying a randomly chosen element in the dataset if it were randomly labeled according to the distribution of labels, in the set and it is often employed to determine the optimal attribute for splitting nodes in a decision tree. The decision tree algorithm selects the attribute that results in the lowest Gini index, indicating the greatest reduction in impurity after the split. The mathematical representation of the Gini index

when used for splitting is as follows:

$$\text{Gini Index}(S, A) = \sum_{v \in A} \frac{|S|}{|S_v|} \cdot \text{Gini}(S_v)$$

(1)

- **S:** is the current data set
- **A:** is the set of possible values of the attribute being considered.
- S_v**:** is the subset of S for which the attribute A has value v

Entropy on the other hand is again a measure of impurity or disorder in a set(one could say it meassures chaos). The mathematical representation is as follows:

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \cdot \log_2(p_i)$$

(2)

- **c:**is the number of classes (unique labels) in set S
- p_i**:** is the proportion of instances in class i in set S

Overall a low entropy indicates that the set is relatively pure, with most instances belonging to a single class whilst a high entropy suggests that the set is impure, with instances spread across multiple classes.

Entropy is a key concept in decision tree algorithms. The way it is used is before splitting we calculate the entropy of the current set S based on the distribution of class labels then after splitting on a particular attribute(in our case in the second part of this task on "habitat") we calculate for each possible value of the chosen attribute the entropy of the subset created by the split and then compute the weighted sum of these entropies based on the size of each subset relative to the total size of the set. Next information gain is measured, information gain is the reduction in entropy achieved by making a particular split. It is calculated as the difference between the entropy before and after the split. The decision tree classifier selects the attribute with the highest information gain for splitting. We overall choose the attribute associated with the maximum information gain as the splitting attribute of each node. And the repeat the proccess until our stoping criterion is met, where in turn we create a leaf node and assign it the label of the majority class in the subset.

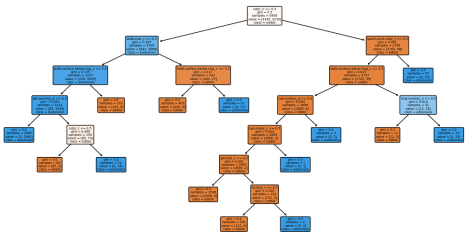

Figure 1: Decision tree-created using Python

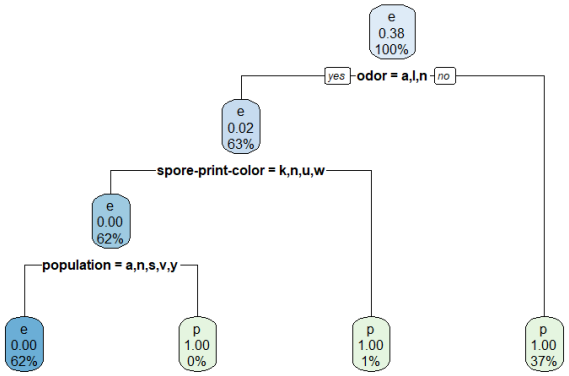

Figure 2: decision tree-created using R

The above presented decision trees 1 and 2 were created using Python and R respectively, the difference in structure observed is due to different programming languages using different implementations of decision tree algorithms. While the core logic and principles remain the same, there might be variations in how certain details are handled or optimized, leading to slightly different trees. The accuracy for both is 100%, while this is possible in certain data sets it overall could be a sign of overfitting, a decision tree that perfectly fits the training data may not generalize well to new, unseen data.

3.2 II

The entropy gain for habitat is computed at **0.33315**, this can be interpreted us that splitting the data based on habitat results in a reduction in entropy by 0.33315 units. Higher entropy gain values indicate a more significant reduction in disorder, and therefore, a more informative split.

3.3 III

In this section we use a Naive Bayes classifier to try and predict whether a mushrooms is poisonous or not, to test its efficiency the confusion matrix of the classifier is computed1. The accuracy of the classifier is computed at 0.945, which is notably a bit lower than the decision tree classifier. This could be due to a plethera of reason such as feature importance in the decision tree, the decision tree boundaries and the characteristics of the misclassified instances, overall it calls for further analysis.

T.N.=775	F.P.=88
F.N.=1	T.P.=761

Table 1: confusion matrix Naive Bayes

4 References

4.1 data

- IMBDataset
- Mushroom dataset

4.2 Theoretical basis

- A Refresher on Regression Analysis
- Introduction to the tm Package Text Mining in R
- Lecture 4: Classification

4.3 Statistical tools

- R
- Python