**Python Assignment – 1**

**Group Name: D**

**Group Members:**

|  |  |  |
| --- | --- | --- |
| **First name** | **Last Name** | **Student number** |
| **Jash** | **Vaghasiya** | **C0884733** |
| **Nivedini** | **Kathagonda** | **C0872720** |
| **Keval** | **Parmar** | **C0882386** |
| **Monil** | **Rupawala** | **C0882370** |
| **Sai Divya Madhuri** | **Guntupalli** | **C0882360** |

**Submission date: *29/07/2023***

Contents

Abstract…………………………………………………………………………………………………….3

Introduction……………………………………………………………………………………………………………………………….3

Methodology 4

Results 18

Conclusion and Future work 19

References 20

**Abstract**

In this study, we thoroughly examine a large dataset containing drug evaluations, ratings, and other relevant factors to find fundamental patterns and derive valuable insights. The research begins with detailed data analysis to identify significant trends and outliers, followed by rigorous language preparation to turn the unstructured text of evaluations into a machine-readable format. We place a high value on feature engineering, which is where we generate and select critical features to strengthen our models. At the same time, we handle the categorical nature of some data items using feature encoding approaches, ensuring that they are appropriately processed and integrated.

To address the issue of data imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is used to provide a balanced representation across classes. With the revised data, we install a succession of complex machine learning models, exposing them to rigorous training and testing to determine their predictive prowess. Our efforts have resulted in a strong model capable of prediction and categorization. Beyond its current scope, this research lays the groundwork for future studies by outlining new lines of investigation and approaches in the large subject of medical data analysis.

**Introduction**

The healthcare sector is undergoing a paradigm shift in the age of digital transformation, with an increased reliance on online platforms for medical guidance and consultation. Patients worldwide may now share their experiences, shedding light on the effectiveness, side effects, and general contentment of their treatments. This democratization of medical input offers enormous promise. It can be a goldmine of information, serving as a touchstone for pharmaceutical research, personalized medicine, and the broader panorama of healthcare management. The main challenge is navigating this massive sea of unstructured data. We can turn this issue into an opportunity by leveraging modern analytical approaches and converting patient feedback into useful information.

Understanding patient reviews is quite important. On the one hand, these reviews can help other patients make informed health decisions based on their own experiences. On the other hand, they provide pharmaceutical corporations, researchers, and healthcare professionals with an in-depth look into how a drug performs outside of clinical trials in various real-world scenarios. This interaction between individual patients' personal experiences and the broader ramifications for the medical community makes a compelling justification for rigorous drug review analysis. Unraveling the patterns, attitudes, and numerous elements influencing a drug's rating might pave the path for more effective and patient-centered healthcare solutions.

Given the data's intricacy and breadth, a systematic strategy for dissecting it is required. This is the foundation of our research. We want to create a story that decodes current attitudes and provides the groundwork for predicting models by delving deep into a dataset rich with medicine reviews, ratings, conditions, and other relevant factors. Such models, supported by rigorous data processing and machine learning approaches, have the potential to transform how we receive patient input, bridging the gap between individual experiences and global healthcare trends.

**Methodology**

**Dataset Details:**

The project dataset contains several key components sourced from Kaggle:

uniqueID: This unique identification number was given to each entry in the dataset.

drugName: Indicates the name of the drug to which the review applies.

condition: Identifies the illness or medical condition for which the medication was prescribed or used as a treatment.

Review: refers to the written account of the user's or patient's interactions with the medication.

rating: A numeric value provided by the user, usually on a predetermined scale like 1 to 10, that denotes the degree of efficacy or satisfaction with the drug.

Date: Indicates the day the review was published or posted.

useful Count: Represents the tally of individuals who found the review to be informative or helpful.  
  
A screenshot of a computer

Description automatically generated

**Data cleaning and preprocessing:**

**Dropping Columns:**

The **uniqueID** is unique identification number so there is no use of that column.

**Missing Values:**

In the dataset, there was a missing value. Only the condition column has 899 NaN entries; all other columns provide useful data. NaN has been replaced with Unknown in the condition column.  
  
**Duplicate Values:**

There are no duplicate values in the dataset.

**Text pre-processing:**  
The provided set of functions serve the purpose of preparing textual data for analysis in a comprehensive manner:

The tokenize\_text work utilizes the word\_tokenize strategy to break down a given content into person words, changing a sentence or section into a list of discrete words. This encourages the consequent preparing steps.

Another, the lowercase\_text work guarantees consistency by converting all characters within the content to lowercase. This step addresses any case-sensitivity concerns, ensuring that words just "Like" the and "the" are treated as proportionate, disposing of potential errors.

The remove\_stopwords work targets common stopwords, which are regularly overlooked in content examination due to their visit event over different writings and restricted commitment to important experiences. Stopwords envelop terms such as "and," "or," and "not." By disposing of these, the center remains on words that carry more critical semantic weight.

Accentuation marks are tended to by the remove\_punctuation work, which utilizes a customary expression to strip them from the content. This step is particularly profitable when the point is to analyze words and their implications, instead of the auxiliary components of sentences. Accentuation commotion is in this way minimized.

The lemmatize\_text work is outlined to assist streamline the content information by utilizing the WordNet Lemmatizer. This device breaks down words into their base or root shapes. For occasion, varieties such as "running," "ran," and "runner" can all be diminished to the common base shape "run." This not as it were decreases the dimensionality of the information but too improves the capacity to get a handle on the elemental implications of words independent of their linguistic shapes.

At long last, the preprocess\_text work serves as a comprehensive wrapper that successively applies all the previously mentioned operations to a particular column inside a pandas DataFrame. Its central reason is to cleanse and standardize the content for consequent inquire about or utilization in machine learning models. While the first audit substance remains within the 'unprocessed\_review' column, the cleaned adaptation replaces the 'review' column, coming about in a refined dataset that's more conducive to important examination. This handle guarantees that the information is well-prepared, reliable, and prepared for progressed investigation or modeling endeavors.

A screenshot of a computer

Description automatically generated

**EDA and Visualization**

**Summery of Dataset**

A table with numbers and symbols

Description automatically generated

The **uniqueID** is unique identification number so there is no use of that column.

The **rating** column contains values ranging from 1 to 10. The average score is around 7, with a median of 8. This indicates that the majority of the reviews are good. The 25th percentile number is 5, indicating that 75% of the reviews are rated 5 or above.

The **usefulCount** column shows how many users considered a specific helpful review. An appraisal is valuable to 28 people on average. This column's value ranges from 0 (indicating that some studies were not useful to anyone) to 1291. The median is 16, meaning half of the reviews were helpful to 16 or fewer users. The 75th percentile number is 36, suggesting that over 36 individuals found the top 25% of reviews beneficial. The standard deviation is around 36.4, showing that the number of times evaluations were judged valuable varied widely.

**Distribution of Rating:**

**A graph with blue lines

Description automatically generated**

* Ratings ranging from 1 to 2 indicate **high discontent** among customers.
* Ratings ranging from 8 to 10 indicate **strong contentment** among customers.
* Ratings ranging from 4 to 7 indicate **mixed or average feelings**.

**Number of Reviews per Drug (Top 20)**

A graph of blue bars

Description automatically generated

* With 3657 and 3336 reviews, respectively, "Levonorgestrel" and "Etonogestrel" are the most reviewed medications, showing significant user experience or widespread usage of these drugs.
* The majority of the pharmaceuticals on the list are associated with hormone treatments, antidepressants, and mood stabilisers, indicating a high prevalence of feedback or experiences shared for these prescription categories.

**Number of Reviews per condition (Top 20):**

A graph with blue bars

Description automatically generated

#### **Distribution of Ratings**

The ratings in the dataset are skewed towards higher values, with a rating of 10 being the most common. This skewness could be attributed to people being more likely to leave reviews when they have an exceptionally positive experience with a drug.

#### **Number of Reviews per Drug (Top 20)**

The dataset shows that "Levonorgestrel" is the most reviewed drug, followed by "Etonogestrel" and "Ethinyl estradiol / norethindrone". This observation suggests that these drugs are either commonly prescribed or have a significant impact on users, prompting more reviews regarding their effectiveness or potential side effects.

#### **Number of Reviews per Condition (Top 20)**

The most common condition for which reviews are written is "Birth Control", followed by "Depression" and "Pain". This pattern may indicate that these conditions are prevalent among the population or that they significantly affect patients' lives, leading to more user feedback about their experiences with treatments.

**WordClouds**

**Review Wordcloud** **Positive Reviews WordCloud**

**A close up of words

Description automatically generatedA close up of words

Description automatically generated**

### **WordCloud of the reviews with rating equal to 1**

A close up of words

Description automatically generated

#### **Observation of all the Word Clouds:**

**Recurring Themes:**

* Words like "Side Effect", "birth control", "weight gain", and "mood swing" indicate that many users are discussing the side effects of the medications, particularly those related to birth control. This suggests that adverse effects, particularly weight changes and mood swings, are frequently discussed in the evaluations.

**Duration & Consistency:**

* The frequent use of phrases such as "day", "months", "first month", and "every day" suggests that users are discussing the duration of consumption, early experiences, or continuous daily use of the prescriptions.

**General Sentiment:**

* Words like "feel like", "problem", and "taking" convey a personal experience or attitude related with the medicine. The presence of these terms in all types of reviews implies that people are sharing their own journeys, both happy and negative, with the site.

##### **Overall:**

Given that these words appear in all categories of reviews (overall, good, negative, and one-star), it's clear that these are overarching themes and worries that people express when discussing their experiences with these prescriptions.

**Top 10 Drugs Used for Birth Control:**

**A graph of green and grey bars

Description automatically generated with medium confidence**

* **Popularity and Usage:** With 3314 mentions, "Etonogestrel" is the most mentioned birth control pill, indicating that it may be one of the most popular or commonly used alternatives. This is closely followed by "Ethinyl estradiol / norethindrone" (2337 mentions) and "Nexplanon" (2149 mentions).
* **Variety of Options:** The list includes both single-component medications (such as "Etonogestrel" and "Nexplanon") and combination drugs (such as "Ethinyl estradiol / norethindrone"). This suggests that consumers are discussing and employing a wide range of birth control techniques, both in terms of active components and brand names.
* **Notable Brands:** Specific products such as "Nexplanon," "Implanon," and "Mirena" appear on the list as well, indicating that certain named birth control techniques have a large user base or are regularly discussed

**Number of reviews per year:**

**A graph showing the growth of a number of years

Description automatically generated**

* The number of reviews has clearly increased throughout the years, beginning in 2008 and culminating around 2016.
* This could imply an increasing tendency of people expressing their drug experiences online, or a rise in the platform where these reviews are gathered.
* The modest decrease in 2017 could be attributed to inadequate data for that year or other factors influencing the number of reviews

**Top 10 Conditions:**

**A graph of different colored squares

Description automatically generated**

* Reviews for the most common ailments include "Birth Control," "Depression," "Pain," and "Anxiety," among others.
* "Birth Control" takes the lead by a wide margin, showing that a sizable number of users are discussing their experiences with birth control drugs.
* The use of terms such as "Depression" and "Anxiety" implies that mental health treatments are commonly addressed medication.

**Unigrams**

A screenshot of a graph

Description automatically generated

**BiGrams**

**A screenshot of a graph

Description automatically generated**

**Trigrams:**

A screenshot of a graph

Description automatically generated

* + We would expect to observe words associated with negative attitudes or adverse effects in unfavourable evaluations. Words such as "pain", "problem", "issue", "worst" and "bad" could be among the top unigrams.Words expressing happiness or favourable outcomes may predominate in positive reviews. Words like "good", "effective", "best", "happy", and "love" could be among the top unigrams
  + Bigrams and trigrams provide more contextual information. Negative reviews may include phrases like "side effects," "feel bad," or "waste of," whereas positive reviews may consist of words like "works well," "highly recommend," or "feel great."

**Word Count Plot:**

A graph showing a number of different colored bars

Description automatically generated with medium confidence

This would provide a comprehensive snapshot of the most frequently used words across all evaluations. Words like "general drug use," "common conditions," and "standard sentiments" (both positive and negative) are likely to feature here

**Feature Engineering:**

#### Description of DataFrame Feature Engineering Columns:

* **count\_word:** The total number of words in each records review column.
* **count\_unique\_word:** The total number of unique words in each records review column.
* **count\_letters:** The total number of characters (including spaces and punctuation) in each records review column.
* **count\_punctuations:** The total number of punctuation characters in each record's 'unprocessed\_review' column.
* **count\_words\_upper:** The number of uppercase words in the 'unprocessed\_review' field for each entry.
* **count\_words\_title:** The number of words in each record's 'unprocessed\_review' column in title case (i.e., the word's initial letter is uppercase).
* **count\_stopwords:** The number of frequently used (stop words) in each record's 'unprocessed\_review' field.

A screenshot of a computer

Description automatically generated

**Correlation Plot:**

A screenshot of a graph

Description automatically generated

**Observations:**

* The positive association between count\_word, count\_unique\_word, and count\_letters is expected. As the number of words in a review grows, so will the number of unique words and total letters.
* count\_word (as well as count\_unique\_word and count\_letters) exhibits a strong positive association with count\_stopwords, count\_words\_title, and count\_punctuations, implying that as reviews get longer, they naturally contain more of these components.
* The rating has a weak yet significant positive association with usefulCount. This shows that people may find higher-rated reviews more valuable, but other factors will likely influence a review's perceived usefulness.
* The rating also has a negative association with the year, meaning that the average rating may have deteriorated with time.

### **TF-IDF Vectorization**

* The written reviews are then converted into a numerical format using the Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer. The TfidfVectorizer is initialized with max\_features=1500, which implies it will only consider the top 1500 terms in the corpus, ordered by term frequency. This method aids in lowering the dataset's dimensionality by focusing on the most critical words while filtering out some of the noise or less relevant phrases. The resulting matrix is then turned into a data frame, review\_df, which is then concatenated with the original df (without the'review' column) to produce data\_final, a data frame enriched with the reviews' TF-IDF properties.
* By selecting only 1500 features, I am attempting to compromise between obtaining the essential information from the reviews and managing computational complexity and memory utilization, which is especially important when dealing with massive datasets.

**Modeling:**

**Logistic Regression:**

With training and test scores barely above 0.31, the Logistic Regression model performs poorly. The model is significantly biased towards predicting the '10' class, as evidenced by its recall of 1.00 for that class. At the same time, it fails to generate any significant predictions for the other courses (with precision, memory, and f1-scores mostly being 0). This indicates a considerable class imbalance problem or an insufficient model configuration.

**Decision Tree Classifier:**

As shown by a flawless training score of 1.0 compared to a significantly lower test score of 0.57, the Decision Tree model is overfitting the training data. Despite this, the model performs very well across classes, with precision, recall, and f1-score values typically ranging between 0.45 and 0.67, although there is potential for improvement, particularly in middle-tier courses.

**Random Forest Classifier:**

As seen by a flawless training score of 1.0 vs. a test score of 0.667, the Random Forest model is overfitting the training data. However, it performs significantly better than earlier models. With f1-scores of 0.71, the model is especially good at predicting classes '1' and '10'. Despite great precision (particularly classes 2–8), other classes have low recall, resulting in moderate f1-scores. This implies that, while the model's predictions are reliable when it predicts specific courses, it is frequently hesitant or misses numerous actual instances of these classes

**MultiNomial Naivebayes:**

With training and test scores both below 0.19, the Multinomial Naive Bayes model performs poorly. The model's predictions are distributed thinly over several classes, but it needs help to produce precise forecasts for most of them. The category '10' has the highest f1-score of 0.36, yet even this is subpar. Classes '6' and '8' have inferior precision and recall values, indicating that the model needs help distinguishing and correctly classifying these classes. This model lacks the predictive capabilities required for this dataset and could benefit from more tuning or a new modeling strategy

**K-Nearest Neighbour:**

The K-Nearest Neighbours (KNN) model overfits the training data, as demonstrated by a training score of 0.473 versus a test score of 0.2506. Except for class '10', which has a f1-score of 0.43, the model needs help to predict most types with high precision and recall. However, even this is not particularly impressive. Courses with low f1-scores include '2', '3', '4', '5', and '6', demonstrating the model's difficulties in identifying and correctly classifying these classes. The overall performance of the KNN model needs to be improved for this dataset, indicating the need for hyperparameter adjustment or alternate modeling methodologies.

**Confusion Matrix:**

A screenshot of a table

Description automatically generated

### **Confusion Matrix Observation:**

**1. Logistic Regression Confusion (logreg\_confusion):**

* Most predictions are focused on the last class (index 9). Other courses have a low number of correct predictions, suggesting poor performance. The model is skewed towards predicting the dominating class.

**2. Decision Tree (dt\_confusion):**

* The diagonal elements representing correct predictions are more evenly distributed than in Logistic Regression. This shows that the Decision Tree retains more distinctions across classes but favors guessing the last type.

**3. Random Forest Confusion (rf\_confusion):**

* The more significant amounts along the diagonal indicate that this model outperforms the Decision Tree. However, as with the Logistic Regression and Decision Tree models, there is a considerable bias toward the last class.

**4. Naive Bayes (nb\_confusion):**

* Like Logistic Regression, this model exhibits a bias towards the last class. It delivers a diverse set of predictions across classes, albeit the diagonal elements (right predictions) are lower than one may expect.

**5. K-Nearest Neighbours (knn\_confusion):**

* KNN predicts in a somewhat balanced manner across classes. Although the last class has a greater prediction rate, it only dominates the forecasts for other classes as much as in past models.

Overall, all models are biased towards the last class, implying that the dataset is imbalanced with a higher proportion of samples from the previous category. Random Forest and K-Nearest Neighbours handle this bias better than the other models, although there is still space for improvement. Techniques such as resampling, class weighting, or employing other evaluation metrics may help to improve model performance on such data.

**Improvements :**To deal with the dataset's imbalance, we can employ the Synthetic Minority Over-sampling Technique (SMOTE). SMOTE generates synthetic samples in the feature space. It selects k of the nearest neighbors for each sample in the minority class, selects one at random, and generates similar but randomly altered new examples.

* The Random Forest model's overall accuracy after employing SMOTE is 34.48%. This is not exceptionally high, indicating that the model needs help correctly classifying a large number of observations.
* SMOTE rectified the class imbalance, but the findings show that while the data imbalance may have been fixed, the classifier is still ineffective at discriminating between the distinct classes.
* Class 10 achieved the highest recall at 0.78, identifying 78% of instances accurately. However, its prevalent support might lead to biased predictions. Class 1 showed balanced precision and recall. Classes 2-9 had lower recall and precision, indicating frequent overlooking of real examples. F1-scores were generally low, reflecting trade-offs between accuracy, recall, and precision across classes.
* In conclusion, while SMOTE has corrected the class imbalance, the RandomForest classifier still performs poorly. To improve the model's predictive capability, further refinement in preprocessing, feature engineering, or experimenting with other algorithms may be required.
* We increased Random Forest classification performance by refining the preprocessing pipeline by using Word2Vec for text representation and PCA, followed by Recursive Feature Elimination (RFE) for feature selection.

**Observations after performing Random Forest Classification:**

* **Overfitting:** Overfitting occurs when the model performs very well on the training data (train score: 0.998), but much worse on the test data (test score: 0.677), indicating overfitting.
* **Imbalance in Class:** Some classes have more data than others, which affects precision and recall. Classes with more help tend to perform better.
* **Precision and Recall Scores Vary:** Precision and recall scores vary between classes. Certain classes are accurately predicted (for example, class 10) whereas others struggle (for example, classes 2, 3, 4).
* **F1-Score Balance:** F1-scores vary between courses, showing different trade-offs between precision and recall.
* **Average F1-Scores:** Classes with lower support influence the weighted average F1-score (0.67), influencing the overall performance rating.
* **Potential Improvement:** Model performance for classes with poorer precision and recall could be improved utilizing hyperparameter tuning or more advanced models.
* **Distinct Class 10 Prediction:** Class 10 is distinguished. Class 10 has a high precision and recall, indicating a significant separation from the other classes.

**Results**

We successfully did exploratory data analysis (EDA), data cleaning, preprocessing, and modelling on the provided real estate listings data. The EDA featured visualisations such as histograms, bar graphs, and word clouds to grasp the data's features better. The data was cleaned and prepared for modelling during the preprocessing step. We attempted text preprocessing, such as tokenization and lemmatization.

We performed smote analysis to rectify the class imbalance and increased Random Forest classification performance by refining the preprocessing pipeline by using Word2Vec for text representation and PCA, followed by Recursive Feature Elimination (RFE) for feature selection.

**Conclusion**

We went on a thorough data science journey in our Jupyter Notebook, beginning with exploratory data analysis (EDA) and visualisation. This first step offered a thorough grasp of the dataset's structure, distribution, and probable anomalies.

Because textual data made up a large component of the dataset, considerable preparation was required. Tokenization, lemmatization, stopword removal, and punctuation exclusion were all used to prepare the text data for further analysis.

Given the categorical character of some variables, label encoding was utilised to convert them to a numerical representation, assuring compatibility with machine learning methods.

TF-IDF vectorization was used to convert the preprocessed text into a format suitable for machine learning. This method efficiently converted the text into numerical vectors while simultaneously emphasising the relevance of less common, more informative phrases.

When we got into the modelling phase, we discovered a recurring issue: class imbalance. This constituted a risk because models in such situations frequently demonstrate a bias towards the majority class. The Synthetic Minority Over-sampling Technique (SMOTE) was used to address issue. SMOTE intended to equalise the number of instances across classes by generating synthetic examples in the feature space.

Following that, several categorization models were trained and tested. Although several models demonstrated promise, there was opportunity for development, particularly in dealing with the complexities of textual data. Following the deployment of SMOTE, the Random Forest classifier displayed considerable improvements in its performance measures, suggesting the impact of resolving data imbalance.

We increased Random Forest classification performance by refining the preprocessing pipeline by using Word2Vec for text representation and PCA, followed by Recursive Feature Elimination (RFE) for feature selection.

In conclusion, this notebook exemplifies the multidimensional nature of data science initiatives, particularly when working with unstructured data such as text. While great progress was made, the journey demonstrated the value of continual iteration and refining. To boost performance, further steps could include going deeper into hyperparameter tweaking, experimenting with sophisticated text representation approaches, or even investigating deep learning models.

**Future Work**

There are various potential future work directions. To boost performance, further steps could include going deeper into hyperparameter tweaking, experimenting with sophisticated text representation approaches, or even investigating deep learning models.

**References**

Towards Data Science. (n.d.). Text Preprocessing in Natural Language Processing using Python. <https://towardsdatascience.com/text-preprocessing-in-natural-language-processing-using-python-6113ff5decd8>

Brownlee, J. (n.d.). SMOTE Oversampling for Imbalanced Classification. Machine Learning Mastery. <https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>

Turing. (n.d.). Guide on Word Embeddings in NLP. Turing. <https://www.turing.com/kb/guide-on-word-embeddings-in-nlp>