Compliant Classification – Marriott Hotels

Introduction:

Many Hotels try to improve their customer experience by analyzing feedback from various sources, such as online reviews and social media comments. With a high volume of feedback flowing in, it became challenging for the brand to analyze and categorize complaints manually. This project aims to develop a compliant classification system that could automatically classify customer complaints and highlight areas needing attention to enhance service quality and customer satisfaction.

Objective:

The objective was to develop a Machine learning model to classify customer complaints. This involved creating a custom dataset, implementing NLP techniques, and training machine learning models to accurately categorize complaints and provide actionable feedback for Marriott's service teams.

Dataset:

The dataset was gathered via web scraping using **Selenium WebDriver** configured with ChromeDriver, installed via webdriver_manager.

The dataset includes customer feedback with the following key columns:

- **Review No**: Unique identifier for each review.
- **Customer Name**: Name or pseudonym of the customer who submitted the review.
- **Date**: Date when the review was posted.
- **Stars**: Star rating assigned by the customer, indicating overall satisfaction.
- **Review**: Text of the customer's feedback, detailing their experience or specific complaints.
- **Services Rated**: Specific services or aspects (e.g., cleanliness, staff, amenities) the customer evaluated.
- **Verified Purchase**: Flag indicating if the review is from a verified transaction, enhancing the review's credibility.

Methodology:

1.Data Preprocessing:

1. **Loading Data**
    - The dataset is loaded using Pandas, and any missing values in each column are identified with df.isnull().sum() to understand where data cleaning may be needed.
2. **Text Cleaning**
    - **Lowercasing**: All text in the *Review* column is converted to lowercase to maintain uniformity.
    - **Removing Special Characters**: Non-alphabetic characters (e.g., punctuation, numbers) are removed using regular expressions, leaving only alphabetical text.
3. **Stop Words Removal and Lemmatization**

   - **Stop Words Removal**: Commonly used words that do not add much meaning (e.g., "is," "the," "and") are removed from the text to focus on more significant terms.
   - **Lemmatization**: Words are reduced to their base form using a lemmatizer, which helps group similar words (e.g., "running" becomes "run") to reduce dimensionality in text analysis.

4. **Processed Review Column**
   - A new column, *Processed_Review*, is created in the dataset, containing reviews after stop word removal and lemmatization, making it ready for further analysis, such as feature extraction and model training.

2.Topic Modeling:

1. **Tokenization**
   - Each review in the *Processed_Review* column is split into tokens (words), which creates a list of tokenized reviews suitable for topic modeling.
2. **Dictionary and Corpus Creation**
   - A dictionary is created from the tokenized reviews, mapping each unique word to an integer ID.
   - A *corpus* is generated, representing each review as a Bag-of-Words (BoW), where each entry contains word IDs and their frequencies in the review. This structure is required by the LDA model.
3. **LDA Model Training**
   - An LDA model with 5 topics was trained on the corpus. The number of topics can be adjusted to explore various levels of detail in the topics identified.
   - The passes=10 parameter specifies the number of iterations for model convergence, and random_state=42 ensures reproducibility.
4. **Topic Interpretation**
   - Each topic generated by the LDA model includes the top 5 words associated with it. This helps in interpreting the topics, as the frequent terms within each topic give insights into themes discussed in customer reviews, such as issues with "service," "amenities," or "staff."
5. **NMF Modeling**
   - TF-IDF features are created from the processed reviews using TfidfVectorizer, limiting to the top 1000 features.
   - An NMF model is trained with a different number of topics (num_topics_nmf), which can also be adjusted.
   - The document-topic matrix (W) and the topic-word matrix (H) are obtained.
   - The top 5 words for each NMF topic are printed.

3. Multi-Label Classification and Feature Engineering:

1. Label Classification:

- A function classify_labels is defined to classify reviews into multiple labels based on the presence of specific keywords.
- The labels include:
  - **Product-related**: Assigned if keywords like 'product', 'price', or 'item' are found.
  - **Service-related**: Assigned if keywords like 'service', 'staff', or 'customer' are found.
  - **Payment-related**: Assigned if keywords like 'payment', 'bill', or 'charge' are found.
  - **Technical issues**: Assigned if keywords like 'app', 'website', or 'technical' are found.
- The function is applied to the Processed_Review column, and the resulting labels are stored in a new column named Labels.

2. Label Binarization:

- The MultiLabelBinarizer from scikit-learn is used to convert the list of labels into a binary format suitable for multi-label classification.
- The transformed labels are stored in the variable y.

3.Feature Engineering:

- TF-IDF features are generated from the processed reviews using TfidfVectorizer, limiting to the top 500 features.
- The resulting feature matrix is stored in the variable X, which will be used for training classification models.

4. Evaluation Summary:

1. **Logistic Regression**:
   - **Best Parameters**: Identified through GridSearchCV.
   - **Classification Report**: Provided metrics including precision, recall, and F1-score for each label.
   - **Key Metrics**:
     - Precision: Indicates the proportion of true positive predictions among all positive predictions.
     - Recall: Reflects the ability of the model to capture all relevant instances.
     - F1-score: The harmonic mean of precision and recall, offering a balance between the two metrics.
2. **Random Forest**:

- **Best Parameters**: Optimized using cross-validation.
- **Classification Report**: Included similar metrics as Logistic Regression.
- **Key Metrics**:
  - Assessments typically show Random Forest's ability to handle class imbalance and provide robust predictions.

3. **Support Vector Classifier (SVC)**:
   - **Best Parameters**: Found through grid search across various configurations.
   - **Classification Report**: Evaluated performance for each label.
   - **Key Metrics**:
     - Often demonstrates strong performance in high-dimensional spaces, which is beneficial for text data.
4. **Multinomial Naive Bayes**:
   - **Best Parameters**: Determined via GridSearchCV.
   - **Classification Report**: Metrics indicating model performance across labels.
   - **Key Metrics**:
     - This model typically performs well with text classification due to its probabilistic nature and assumption of feature independence.

5.Conclusion:

1. Engagement as a Reliable Metric: The project successfully utilized customer feedback metrics, such as star ratings and review content, as indicators of customer satisfaction. Reviews that aligned with higher engagement levels, as assessed through sentiment analysis and classification thresholds, were deemed more significant for improving service quality.

2. Threshold-based Classification: Implementing a threshold mechanism based on model predictions provided an effective way to categorize customer complaints. This approach helped in distinguishing between various types of feedback, such as product, service, payment, and technical issues, allowing for more focused and actionable insights.

3. Performance of Machine Learning Models: The use of various classifiers, including Logistic Regression, Random Forest, and SVC, demonstrated effective classification of customer complaints. The models were fine-tuned through hyperparameter optimization, enhancing their predictive accuracy in identifying key issues raised by customers.

4. Scalability and Adaptability: The classification system is scalable and can be adapted for other sectors within the hospitality industry or beyond. By modifying the features and thresholds, the system can be used to analyze a broader range of customer feedback, enhancing its applicability across different contexts.

Overall, the project establishes a robust framework for classifying customer complaints at Marriott Hotels, leveraging machine learning to provide valuable insights into customer experiences. By categorizing feedback effectively, the classification system empowers the organization to address concerns proactively and improve service offerings. Future iterations could explore advanced techniques, such as deep learning, to further enhance classification accuracy and expand the system's capabilities.

Done By,
Valluri Vinuthna & Kudumula Varalakshmi Himaja