* 🔹 **Title of the Project**
* **Sentiment analysis**
* ​
* 🔹 **Objective**
* The goal of this project is to build a sentiment analysis model capable of determining the sentiment (positive, negative, or neutral) expressed in a given text. The system will leverage both traditional machine learning techniques and deep learning models to provide accurate sentiment predictions. The project will focus on data exploration, preprocessing, model selection, and optimization, culminating in a robust sentiment analysis solution with real-time capabilities.
* ​
* 🔹 **Dataset Used**
* **Sentiment Analysis Dataset (Amazon Reviews & Twitter Tweets)**
* This dataset contains:
* **Reviews dataset**: Text data from Amazon product reviews and Twitter tweets.
* **Columns**:
* **Text**: The content of the review or tweet.
* **Source**: Indicates whether the text is from an Amazon review or a Twitter tweet.
* ​
* 🔹 **Model Chosen**
* T **Two main approaches are implemented:**
* **Naive Bayes Classifier**  
  A probabilistic model used for text classification. It applies Bayes' Theorem with strong (naive) independence assumptions to predict the sentiment (positive, negative, or neutral) of a given review or tweet.
* **Linear Regression Model**  
  A regression-based model that predicts sentiment scores or continuous sentiment values (if applicable) based on features extracted from text data. While typically used for numeric predictions, it can be adapted for sentiment analysis by treating sentiment labels as continuous values.
* Libraries used:
* Pandas, NumPy, Scikit-learn, Matplotlib,Streamlit
* ​
* ​
* 🔹 **Challenges & Learnings**
* **Challenges Faced:**
* **Handling sparse data:** Many users only provide sentiment on a few pieces of text, making it challenging to analyze a wide range of opinions.
* **Cleaning and merging datasets:** Datasets often contain inconsistent formats, missing values, or noisy data, which require significant preprocessing.
* **Ensuring accurate sentiment predictions:** Balancing between false positives and false negatives while making sure the model captures nuanced sentiments in various contexts.
* **Dealing with imbalanced classes:** Some sentiments (e.g., positive vs. negative) may be underrepresented, affecting the accuracy of the analysis.
* **Handling sarcasm:** Sarcastic comments can completely invert the meaning of a sentence, making it difficult for the model to accurately predict sentiment. Detecting sarcasm requires advanced techniques and contextual understanding.
* **Learnings:**
* **Better understanding of sentiment analysis algorithms:** Gained more insight into natural language processing (NLP) models like Naive Bayes, SVM, and deep learning for sentiment classification.
* **Importance of data preprocessing:** Realized how crucial steps like tokenization, stop word removal, stemming, and lemmatization are to improve model performance.
* **Handling real-world text data:** Dealt with noisy, unstructured data (e.g., slang, emojis) and learned how to clean and preprocess text for effective analysis.
* **Feature extraction and vectorization:** Learned how techniques like TF-IDF and word embeddings (e.g., Word2Vec, GloVe) can significantly enhance sentiment prediction accuracy.
* **Building an end-to-end sentiment analysis pipeline:** Gained experience in creating a full workflow, from data collection to preprocessing, model training, evaluation, and deployment.