12.2 Project Report

This example will use a dataset of customer reviews for a fictional e-commerce platform, combining structured data (user information, product details) with unstructured data (review text). Expanded from assignment 4.2 in week four. The pipeline covers data ingestion and storage using TinyDB to store simulated customer review data. Panadas is then used for data processing, cleaning, and feature engineering. Two visual creations are then used for EDA. Statsmodel is used for statistical analysis methodologies to perform a simple regression analysis. PySpark is then implemented for machine learning by calling Spark MLib to train the Random Forest Classifier. Afterward, a simple neural network is built using Tensorflow/Keras for predicting ratings. Text processing with NLTK and transformers such as BERT are used for natural language processing and sentiment analysis. For model deployment, I started with a basic Flask app structure. Unfortunately, I ran into difficulties and pivoted to Streamlit, where it does the following:

* Create a user interface for input customer information and review.
* A button, for when it’s pressed uses the deep learning model to predict the rating based on customer age, review word count, and purchase amount.
* Results are then displayed to the user.

Challenges faced in the project were incompatible libraries such as Tensflow/Keras version with Numpy. To circumnavigate, I used Google collab.

It is important to note that loading large models like BERT directly in a Streamlit app can be memory-intensive and slow. For a production environment, serving these models separately (using TensorFlow Serving) and having the Streamlit app call an API endpoint is best. To further enhance the application, more visualizations, explanations of the results, and additional features are needed.