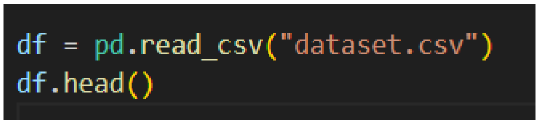
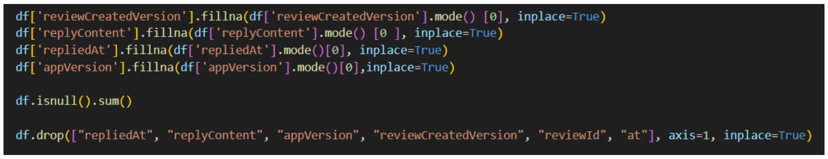
Data Collection and Preprocessing Phase

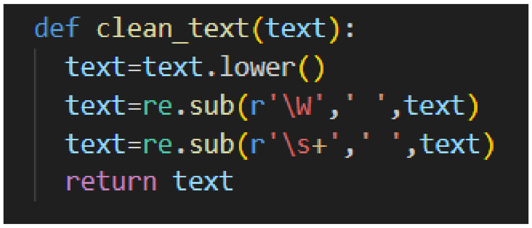
Date	18 June 2025
Team ID	SWTID1750052396
Project Title	Analysis of medium app review from Google Play Store
Maximum Marks	6 Marks

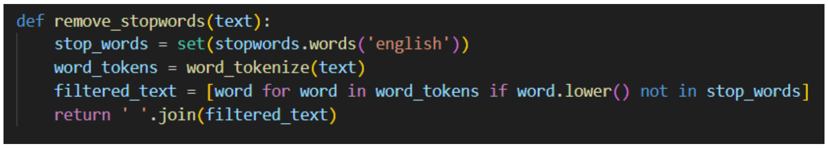
Preprocessing Template

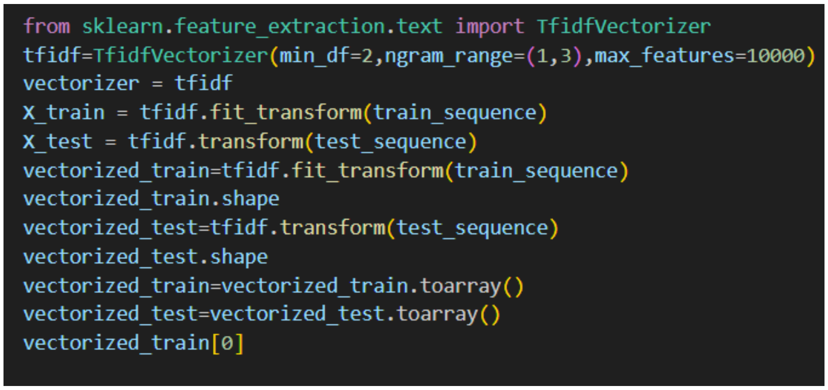
The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	**Description**
Data Overview	The dataset contains user reviews and associated sentiment labels (sentiment). Other unused metadata columns were dropped.
Loading Data	Used pandas.read_csv() to load the CSV file into a DataFrame.
Handling Missing Data	Filled missing values in reviewCreatedVersion, replyContent, repliedAt, and appVersion using mode. Dropped unused columns like reviewId, at.
Text Cleaning	Texts were lowercased, punctuation removed using regex, and multiple spaces collapsed into one.
Stopword Removal	Eliminates common words (like "the", "is", "and") that don't add meaningful information, helping improve model focus and performance.
Vectorization	Converts text into numerical form (using TF-IDF) so machine learning models can process and learn from it.

Data Preprocessing Code Screenshots	
Loading Data	 df = pd.read_csv("dataset.csv") df.head()
Handling Missing Data	 df['reviewCreatedVersion'].fillna(df['reviewCreatedVersion'].mode() [0], inplace=True) df['replyContent'].fillna(df['replyContent'].mode() [0], inplace=True) df['repliedAt'].fillna(df['repliedAt'].mode()[0], inplace=True) df['appVersion'].fillna(df['appVersion'].mode()[0],inplace=True) df.isnull().sum() df.drop(["repliedAt", "replyContent", "appVersion", "reviewCreatedVersion", "reviewId", "at"], axis=1, inplace=True)

Text Cleaning	 def clean_text(text):
	    text=text.lower()
	    text=re.sub(r'\W',' ',text)
	    text=re.sub(r'\s+',' ',text)
	    return text

Stopword Removal	 def remove_stopwords(text):
	    stop_words = set(stopwords.words('english'))
	    word_tokens = word_tokenize(text)
	    filtered_text = [word for word in word_tokens if word.lower() not in stop_words]
	    return ' '.join(filtered_text)

Vectorization	 from sklearn.feature_extraction.text import TfidfVectorizer
	tfidf=TfidfVectorizer(min_df=2,ngram_range=(1,3),max_features=10000)
	vectorizer = tfidf
	X_train = tfidf.fit_transform(train_sequence)
	X_test = tfidf.transform(test_sequence)
	vectorized_train=tfidf.fit_transform(train_sequence)
	vectorized_train.shape
	vectorized_test=tfidf.transform(test_sequence)
	vectorized_test.shape
	vectorized_train=vectorized_train.toarray()
	vectorized_test=vectorized_test.toarray()
	vectorized_train[0]