Video Classification and Analysis for Travel Content

Team Members

- Manikanta Kodandapani Naidu (k11@iu.edu)
- Jayanth Budigini (jbudigin@iu.edu)
- Charan Chowdary Pothumarthi (chpothu@iu.edu)

Abstract

This project developed a system for classifying and analyzing travel-related video content using machine learning techniques. The system categorizes videos and provides insights into travel trends, popular destinations, and viewer engagement. Using a dataset of travel video descriptions, we trained and validated models to enhance the understanding of viewer preferences, aiming to help stakeholders in the travel industry better cater to audience interests.

Keywords:Video classification, Travel content, Machine learning, Viewer engagement, Data analysis

Introduction

The rise of travel blogging and vlogging has led to an exponential increase in travel-related video content on platforms like YouTube. This project capitalized on this trend by developing a comprehensive system for classifying and analyzing travel-related video content. The system utilizes machine learning techniques to categorize videos based on their description content and provide valuable insights into trends, popular destinations, and viewer engagement.By leveraging a provided dataset of travel videos to train and validate the models, the project seeks to achieve the following primary goals:

- Classify travel videos into specific categories (e.g., destination, type of travel).
- Analyze the content to identify trends and popular travel destinations.

Data Collection and Preprocessing

Dataset Overview

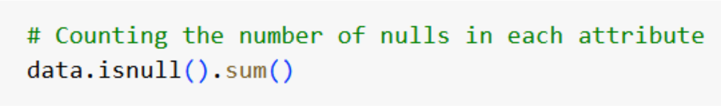
The project used a dataset of travel-related videos, stored in a CSV file named Advertisement_videos_dataset.csv. The dataset contained information about 10,333 videos, including their titles, descriptions, and categories.

Data Cleaning and Preprocessing

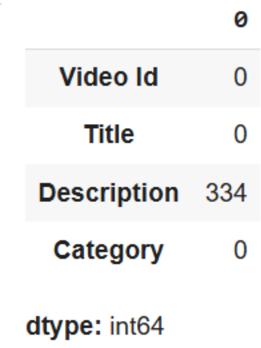
The data preprocessing steps included:

1. Removing duplicate entries

2. Handling missing values
3. Combining title and description into a single text field
4. Cleaning the text data by removing special characters, converting to lowercase, and removing stopwords
5. Applying lemmatization to reduce words to their base form

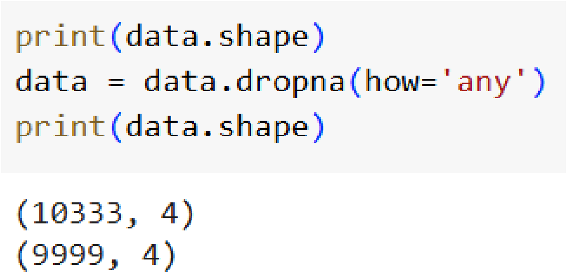


```
# Counting the number of nulls in each attribute
data.isnull().sum()
```



	0
Video Id	0
Title	0
Description	334
Category	0

dtype: int64



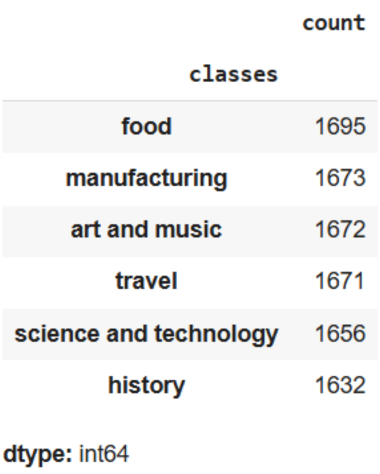
```
print(data.shape)
data = data.dropna(how='any')
print(data.shape)
```

```
(10333, 4)
(9999, 4)
```

Exploratory Data Analysis

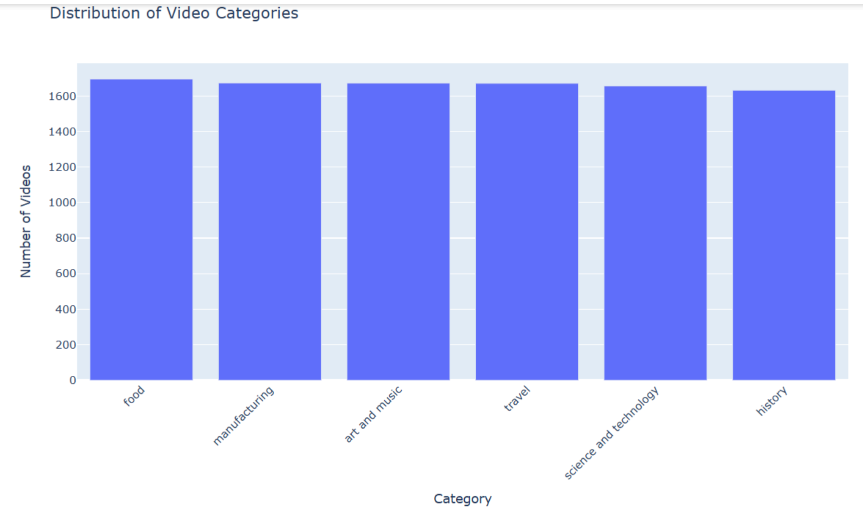
Category Distribution

The dataset contained videos from six main categories:



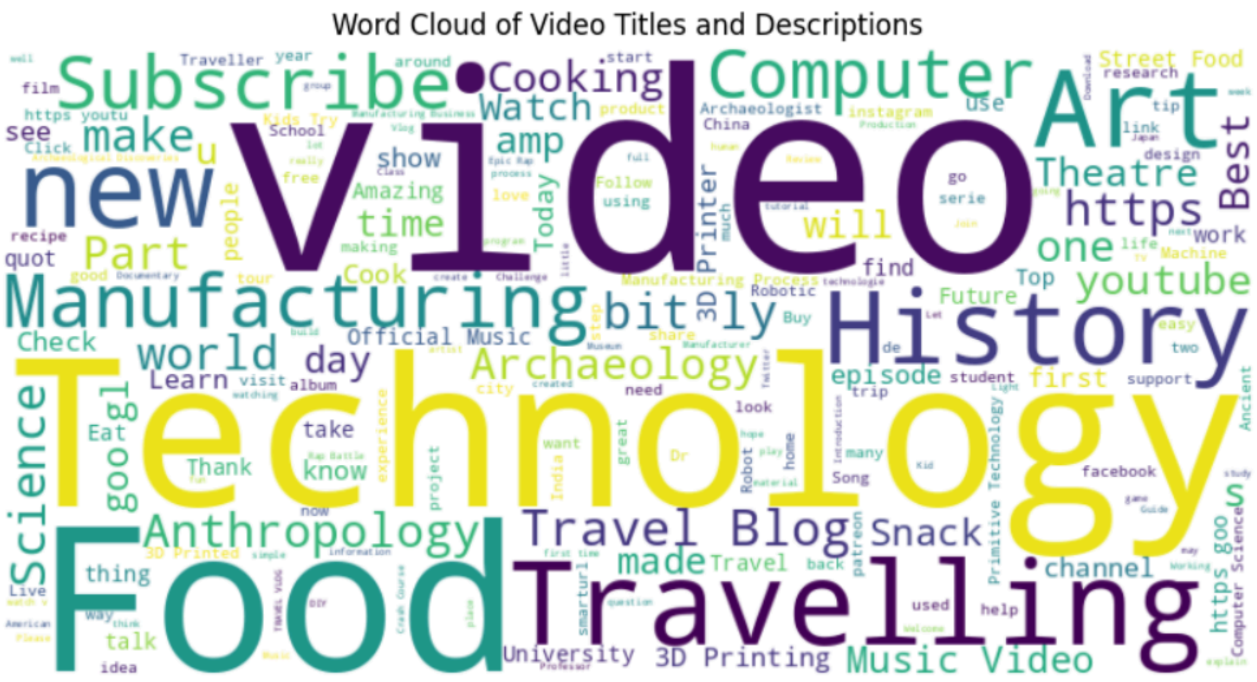
	count
classes	
food	1695
manufacturing	1673
art and music	1672
travel	1671
science and technology	1656
history	1632

dtype: int64



Word Cloud

This word cloud visualizes the most common terms found in the descriptions of travel videos in the dataset.The size of the words indicates their frequency in the dataset, offering insights into the most common topics and trends among travel videos.



Word Cloud of Video Titles and Descriptions

Model Development and Results

For our video classification task, we implemented two models: LSTM and RoBERTa. Both models were trained to classify videos into specific categories based on their titles and descriptions.

LSTM Model

Long Short-Term Memory (LSTM) is an advanced type of recurrent neural network architecture designed to handle sequential data. It is particularly effective for tasks involving time series or text data.

Advantages for our use case:

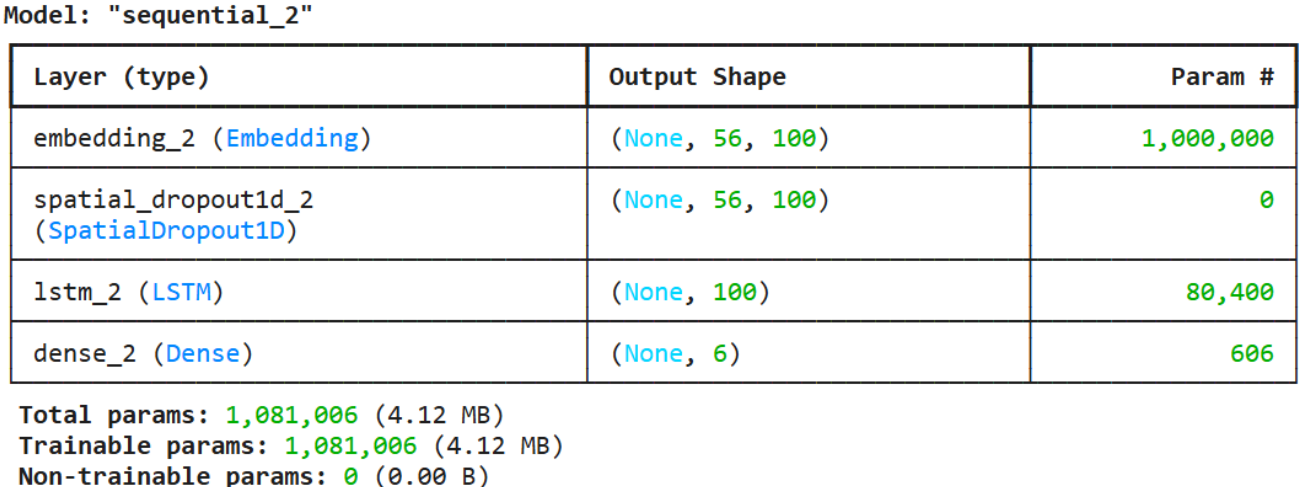
- Captures long-term dependencies in text
- Handles variable-length input sequences
- Effective at learning temporal patterns in video titles and descriptions

LSTM Model Architecture

Our LSTM model consists of:

- **Embedding Layer**: Converts input tokens into dense vectors of fixed size (100 dimensions).
- **SpatialDropout1D**: Helps prevent overfitting by randomly setting input units to 0 during training.
- **LSTM Layer**: Processes the sequence data, capturing long-term dependencies.
- **Dense Layer**: The output layer with 6 units, corresponding to the number of video categories.

The model has a total of 1,081,006 trainable parameters.

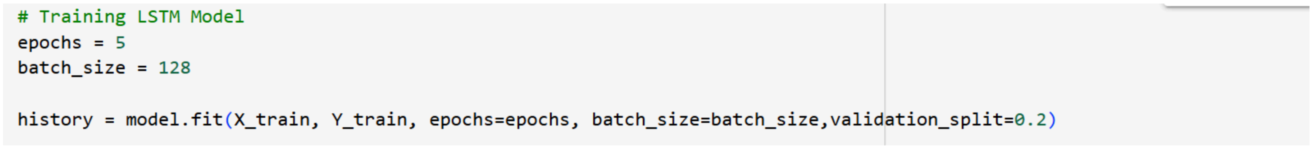


```
Model: "sequential_2"
┌─────────────────────────────────┬────────────────────────┬───────────────┐
│ Layer (type)                    │ Output Shape           │       Param # │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ embedding_2 (Embedding)         │ (None, 56, 100)        │     1,000,000 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ spatial_dropout1d_2             │ (None, 56, 100)        │             0 │
│ (SpatialDropout1D)              │                        │               │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ lstm_2 (LSTM)                   │ (None, 100)            │        80,400 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_2 (Dense)                 │ (None, 6)              │           606 │
└─────────────────────────────────┴────────────────────────┴───────────────┘
 Total params: 1,081,006 (4.12 MB)
 Trainable params: 1,081,006 (4.12 MB)
 Non-trainable params: 0 (0.00 B)
```

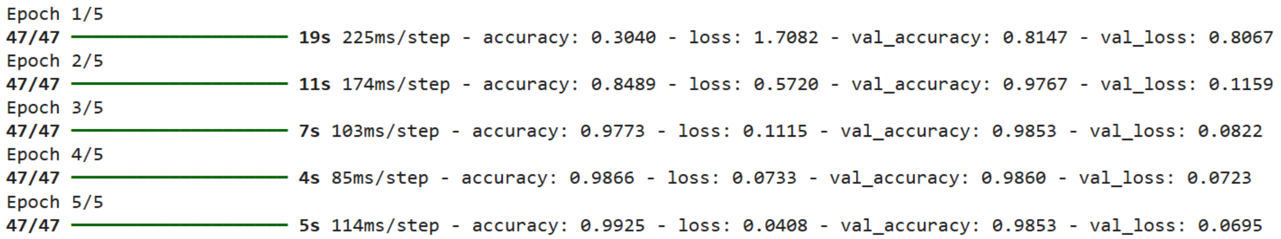
LSTM Training and Test Results

The LSTM model was trained for 5 epochs with a batch size of 128. The model training loss and accuracy curves are shown below. As evident,The LSTM model demonstrated strong performance in classifying videos based on their titles and descriptions. The model achieved a final training accuracy of 99.25% and a validation accuracy of 98.53% after 5 epochs.

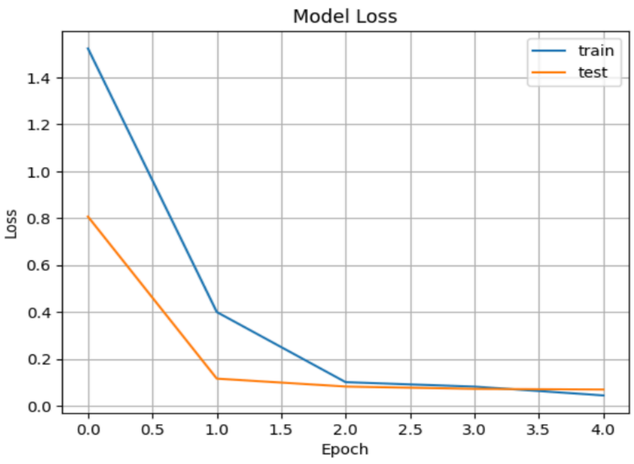
The model showed significant improvement in accuracy from 30.40 % in the first epoch to over 90% by the third epoch and the high validation accuracy suggests good generalization to unseen data.

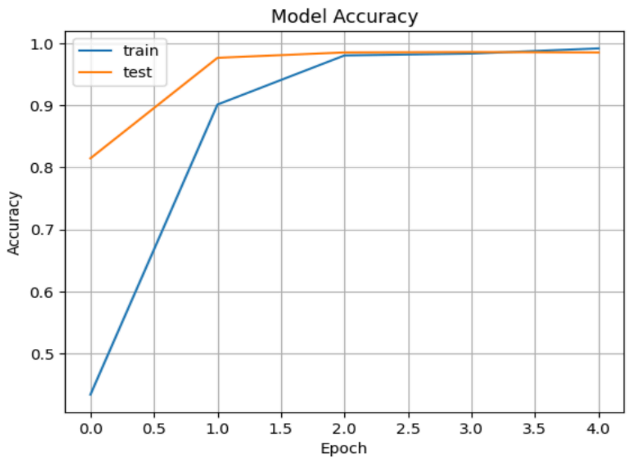


```
# Training LSTM Model
epochs = 5
batch_size = 128

history = model.fit(X_train, Y_train, epochs=epochs, batch_size=batch_size,validation_split=0.2)
```

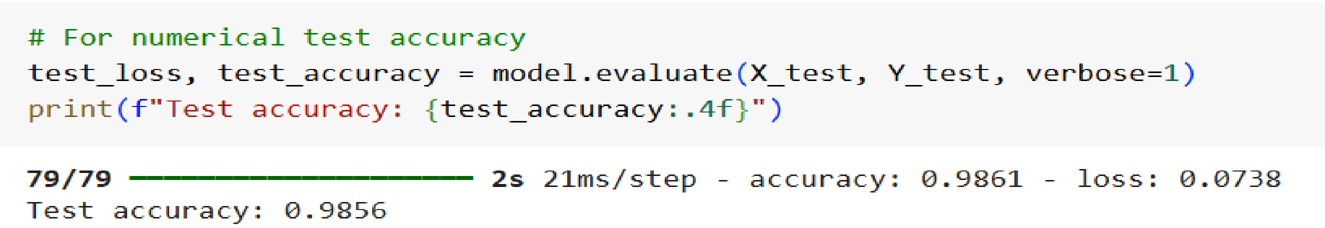


```
Epoch 1/5
47/47 ━━━━━━━━━━━━━━━━━━━━ 19s 225ms/step - accuracy: 0.3040 - loss: 1.7082 - val_accuracy: 0.8147 - val_loss: 0.8067
Epoch 2/5
47/47 ━━━━━━━━━━━━━━━━━━━━ 11s 174ms/step - accuracy: 0.8489 - loss: 0.5720 - val_accuracy: 0.9767 - val_loss: 0.1159
Epoch 3/5
47/47 ━━━━━━━━━━━━━━━━━━━━ 7s 103ms/step - accuracy: 0.9773 - loss: 0.1115 - val_accuracy: 0.9853 - val_loss: 0.0822
Epoch 4/5
47/47 ━━━━━━━━━━━━━━━━━━━━ 4s 85ms/step - accuracy: 0.9866 - loss: 0.0733 - val_accuracy: 0.9860 - val_loss: 0.0723
Epoch 5/5
47/47 ━━━━━━━━━━━━━━━━━━━━ 5s 114ms/step - accuracy: 0.9925 - loss: 0.0408 - val_accuracy: 0.9853 - val_loss: 0.0695
```





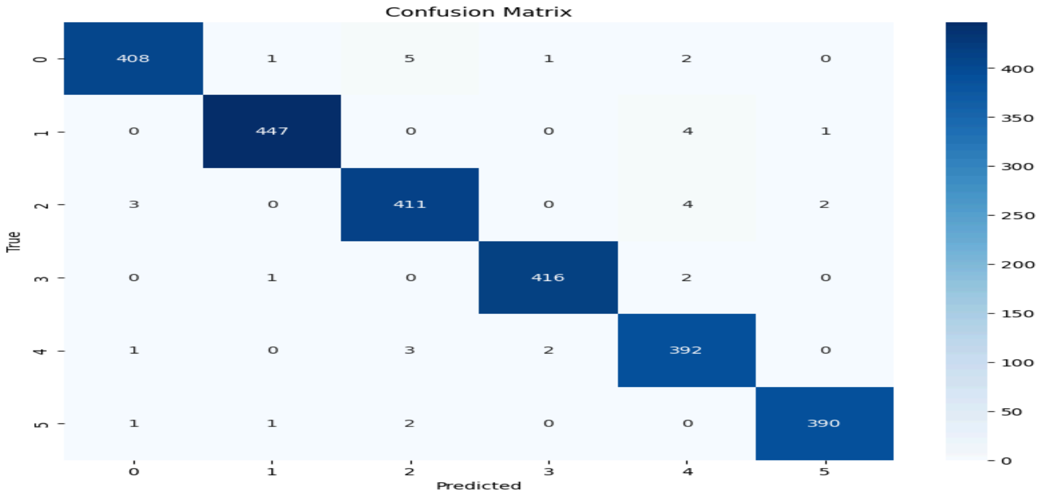
The LSTM model achieved the following results on the test set: **98% Accuracy**



```
# For numerical test accuracy
test_loss, test_accuracy = model.evaluate(X_test, Y_test, verbose=1)
print(f"Test accuracy: {test_accuracy:.4f}")
```

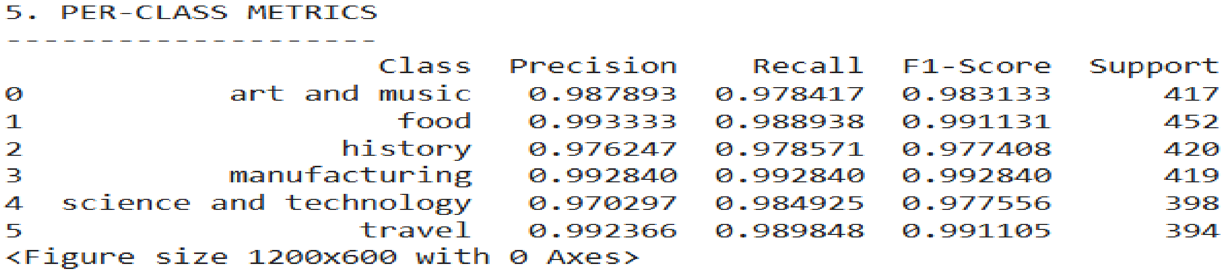
```
79/79 ━━━━━━━━━━━━━━━━━━━━ 2s 21ms/step - accuracy: 0.9861 - loss: 0.0738
Test accuracy: 0.9856
```

LSTM Model Performance Report: A confusion matrix was generated to visualize the model's performance across different target categories. In addition, per class performance metrics are shown below



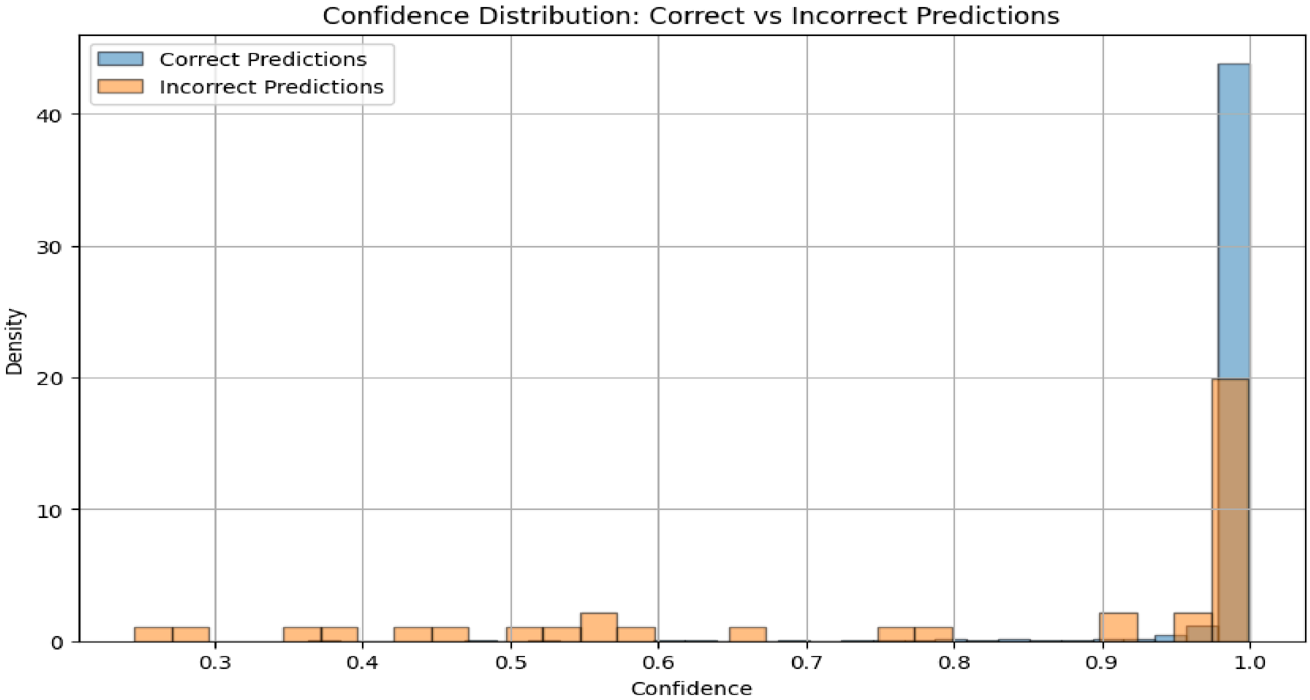
True \ Predicted	0	1	2	3	4	5
0	408	1	5	1	2	0
1	0	447	0	0	4	1
2	3	0	411	0	4	2
3	0	1	0	416	2	0
4	1	0	3	2	392	0
5	1	1	2	0	0	390

The confusion matrix shows the performance of the LSTM model across six target classes, with strong diagonal dominance indicating high accuracy for each class. The diagonal values represent correctly classified instances, such as 408 for Class 0, 447 for Class 1, and so on, showcasing the model's ability to predict most samples accurately. However, minor misclassifications are observed, such as Class 0 being predicted as Classes 1, 2, and 4 a few times, and similar patterns for other classes. For example, Class 4 had 3 instances misclassified into Class 2. These off-diagonal values are relatively low compared to correct predictions, suggesting that while the model performs well overall, there is room for improvement in handling edge cases. Fine-tuning the model or addressing class imbalances may help minimize these errors and further enhance the model's performance.



```
5. PER-CLASS METRICS
----------------------
                    Class  Precision    Recall  F1-Score  Support
0           art and music   0.987893  0.978417  0.983133      417
1                    food   0.993333  0.988938  0.991131      452
2                 history   0.976247  0.978571  0.977408      420
3           manufacturing   0.992840  0.992840  0.992840      419
4  science and technology   0.970297  0.984925  0.977556      398
5                  travel   0.992366  0.989848  0.991105      394
<Figure size 1200x600 with 0 Axes>
```

Examined some of the misclassified videos to understand common errors and potential areas for improvement.



RoBERTa Model

RoBERTa (Robustly Optimized BERT Pretraining Approach) is a state-of-the-art transformer-based language model that improves upon the original BERT architecture.

Advantages for our use case:

- Pretrained on a large corpus of text, allowing for transfer learning
- Captures complex contextual relationships in text
- Highly effective for various natural language processing tasks, including text classification

RoBERTa Model Architecture and Training

The RoBERTa model implementation consists of:

1. A pre-trained RoBERTa base model for feature extraction
2. A custom classifier built on top of RoBERTa, including:
   - A dropout layer for regularization
   - An output layer with softmax activation for multi-class classification

Model Architecture

The RobertaClassifier class defines the model architecture:

- The base RoBERTa model is loaded with RobertaModel.from_pretrained('roberta-base')
- Most layers of the base model are frozen, with only the last 2 layers fine-tuned
- A dropout layer with a rate of 0.3 is added for regularization
- The final classifier layer maps the RoBERTa output to the number of classes

Training Parameters

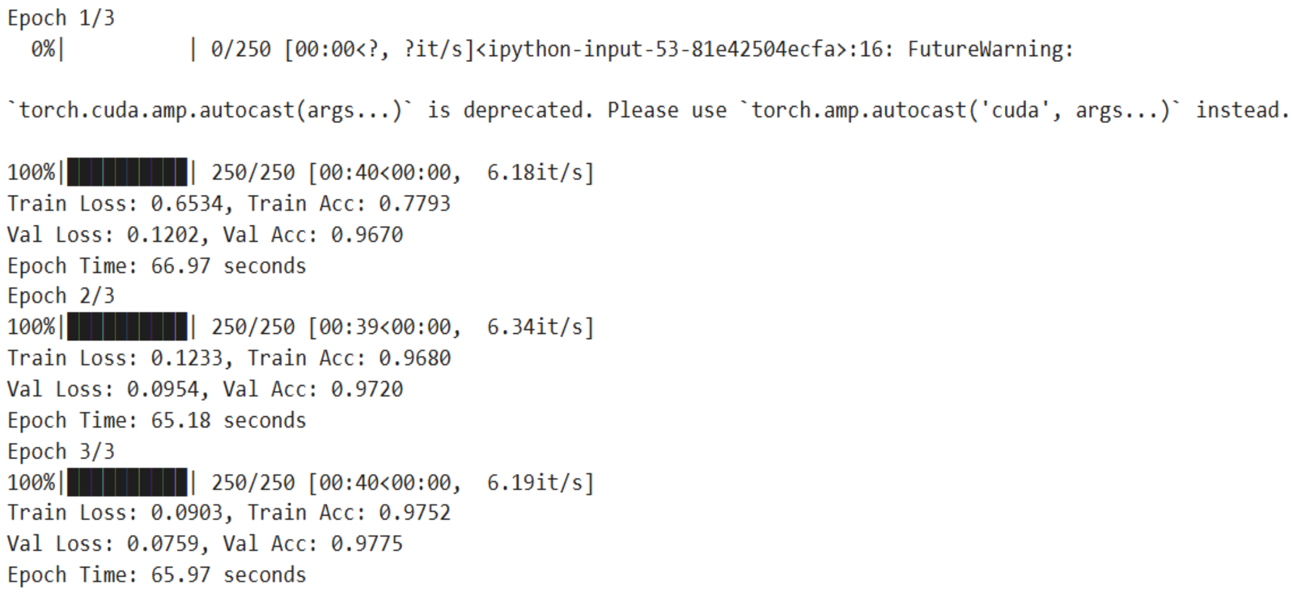
The model was trained using the following parameters:

- Batch size: 32 (effective batch size of 64 with gradient accumulation)
- Learning rate: 3e-5
- Number of epochs: 3 (with early stopping)
- Optimizer: AdamW
- Loss function: Cross-Entropy Loss

Training Process

The training process incorporated several optimizations:

1. Mixed-precision training using torch.cuda.amp.GradScaler for improved speed and reduced memory usage
2. Gradient accumulation with accumulation_steps = 2 for an effective batch size of 64
3. Data loading optimizations:
   - Parallel data loading with num_workers=4
   - pin_memory=True for faster data transfer to GPU
4. Early stopping with a patience of 2 epochs to prevent overfitting
5. Model checkpointing to save the best performing model

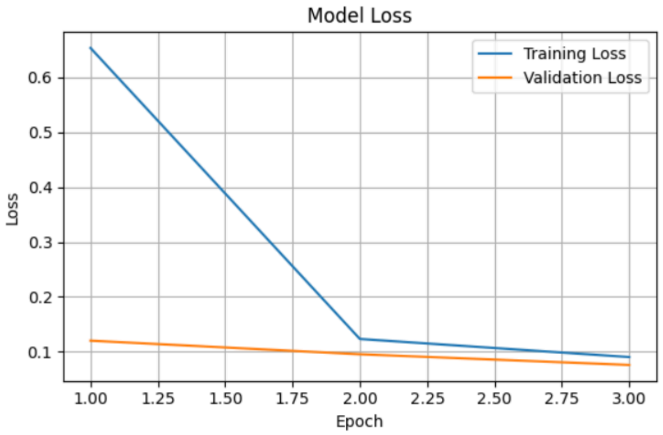


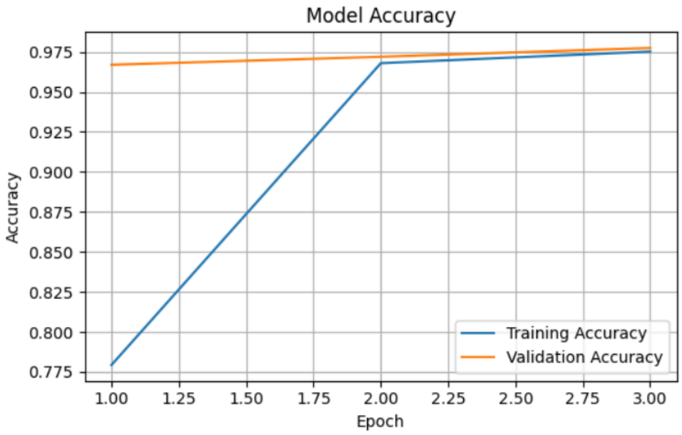
```
Epoch 1/3
  0%|          | 0/250 [00:00<?, ?it/s]<ipython-input-53-81e42504ecfa>:16: FutureWarning:

`torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.

100%|██████████| 250/250 [00:40<00:00,  6.18it/s]
Train Loss: 0.6534, Train Acc: 0.7793
Val Loss: 0.1202, Val Acc: 0.9670
Epoch Time: 66.97 seconds
Epoch 2/3
100%|██████████| 250/250 [00:39<00:00,  6.34it/s]
Train Loss: 0.1233, Train Acc: 0.9680
Val Loss: 0.0954, Val Acc: 0.9720
Epoch Time: 65.18 seconds
Epoch 3/3
100%|██████████| 250/250 [00:40<00:00,  6.19it/s]
Train Loss: 0.0903, Train Acc: 0.9752
Val Loss: 0.0759, Val Acc: 0.9775
Epoch Time: 65.97 seconds
```

Performance Tracking

During training, the following metrics were tracked for each epoch:

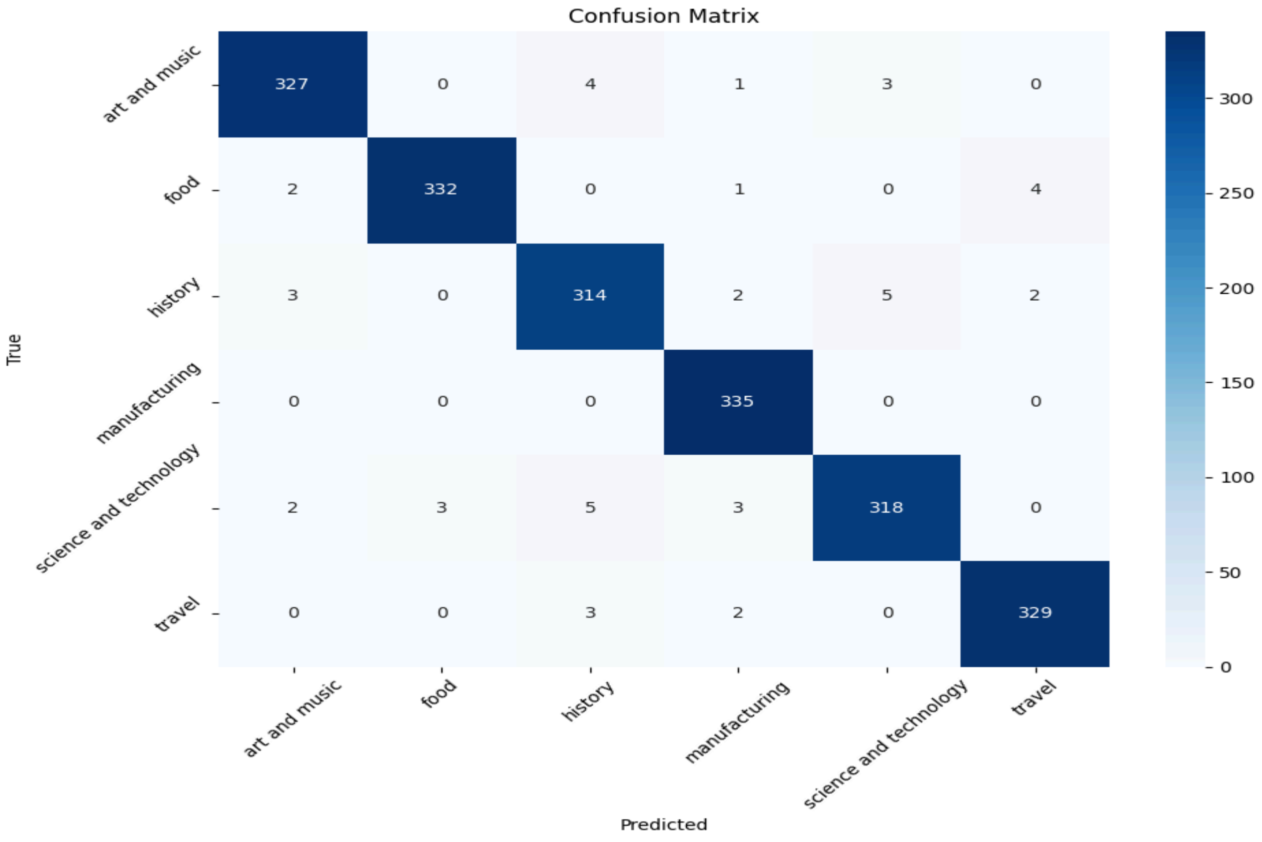
- Training loss and accuracy
- Validation loss and accuracy
- Epoch execution time
- Learning rate



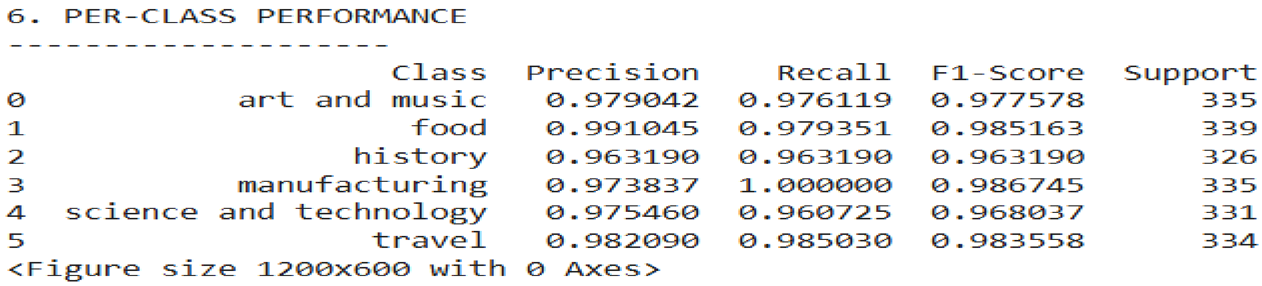


Results

The final model performance on the test set: **Accuracy: 0.9775**

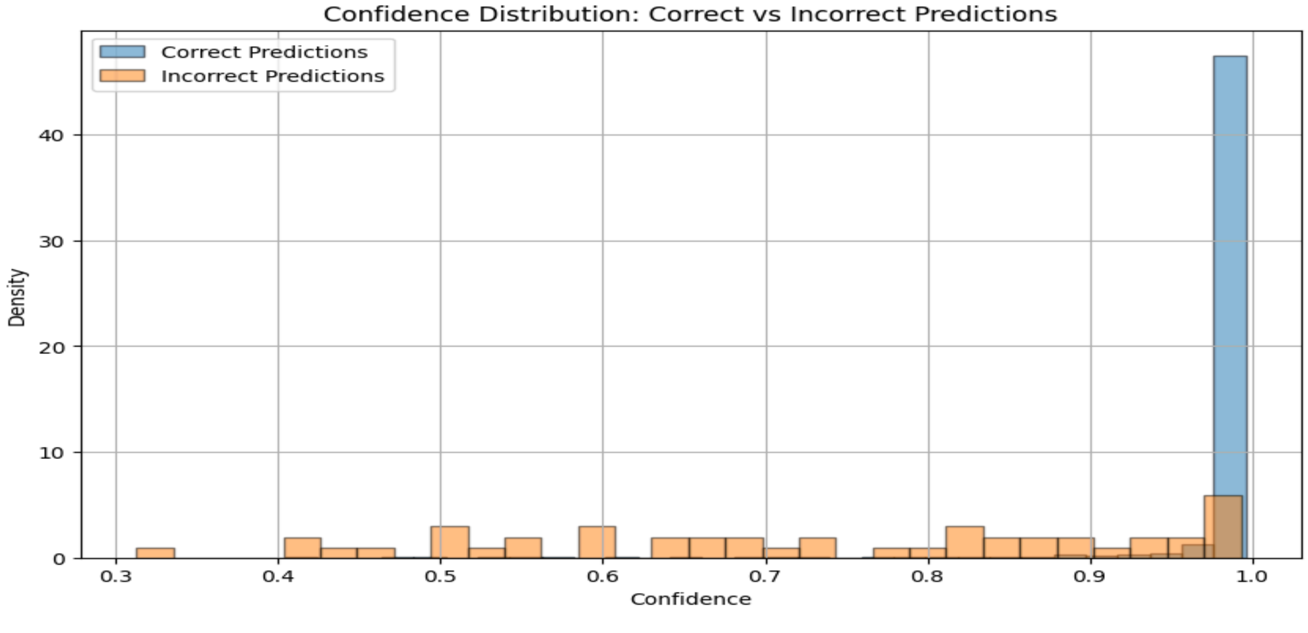


The confusion matrix and per-class performance table show the RoBERTa model's strong accuracy across six categories: art and music, food, history, manufacturing, science and technology, and travel. The diagonal dominance indicates most predictions were correct, such as art and music with 327 out of 335 and manufacturing with a perfect 335 correct predictions. Minor misclassifications, like 4 art and music samples predicted as history, are minimal. The model handles overlapping features well, maintaining consistent performance across classes, which shows its ability to generalize effectively.



```
6. PER-CLASS PERFORMANCE
--------------------
                  Class  Precision    Recall  F1-Score  Support
0         art and music   0.979042  0.976119  0.977578      335
1                  food   0.991045  0.979351  0.985163      339
2               history   0.963190  0.963190  0.963190      326
3         manufacturing   0.973837  1.000000  0.986745      335
4  science and technology   0.975460  0.960725  0.968037      331
5                travel   0.982090  0.985030  0.983558      334
<Figure size 1200x600 with 0 Axes>
```

The per-class metrics further confirm the model's reliability. Manufacturing achieved perfect Recall (1.000), while food and travel had high Precision (0.991 and 0.982) and F1-Scores around 0.98. Slight dips in recall for history and science and technology suggest a few missed samples, which could improve with additional tuning or feature enhancements. Overall, the model balances precision and recall effectively, making it a strong solution for multi-class classification tasks.



The RoBERTa-based model demonstrated strong performance in classifying video content based on titles and descriptions. The use of transfer learning from the pre-trained RoBERTa model, combined with the optimized training process, allowed for effective feature extraction and classification across multiple categories.

Model Comparison

Both LSTM and RoBERTa models demonstrated strong performance in classifying video content based on titles and descriptions. The high performance of both models indicates their effectiveness in capturing the nuances of video content descriptions, enabling accurate classification into predefined categories. This capability can be invaluable for content recommendation systems, targeted advertising, and content moderation in video-sharing platforms. However, we select LSTM over Roberta for the following reasons:

- **Model Efficiency**: LSTM is a lighter model that works efficiently on CPU, making it more accessible for deployment in various environments.
- **Computational Requirements**: RoBERTa typically requires GPU acceleration to execute faster, which may not be available in all deployment scenarios.
- **Speed**: In our tests, LSTM demonstrated faster execution times compared to RoBERTa, allowing for quicker processing of large volumes of video content.
- **Resource Optimization**: The LSTM model's ability to perform well without the need for extensive computational resources makes it a more cost-effective solution for large-scale deployment.

Conclusions and Future Work

The developed system demonstrates strong performance in classifying travel-related video content across multiple categories. The use of RoBERTa embeddings and a neural network classifier proved effective in capturing the nuances of video titles and descriptions.
Future work could include:

1. Incorporating video metadata (e.g., view counts, likes, comments) to improve classification accuracy
2. Exploring more advanced architectures, such as advanced transformer-based models fine-tuned for video classification
3. Implementing a more granular classification system to identify specific travel destinations or activities
4. Developing a user interface for real-time video classification and trend analysis

This project provides a robust framework for classifying and analyzing travel-related video content, offering valuable insights for travel bloggers, tourism boards, and travel companies looking to understand viewer preferences and trends in the travel industry.

References

[1] Cui, Yu, Qing He, and Alireza Khani. "Travel behavior classification: an approach with social network and deep learning." Transportation research record 2672, no. 47 (2018): 68-80.

[2] Nguyen, Phuong Minh Binh, Lan Xuan Pham, Dang Khoa Tran, and Giang Nu To Truong. "A systematic literature review on travel planning through user-generated video." Journal of Vacation Marketing 30, no. 3 (2024): 553-581.

[3] Islam, Md Shofiqul, Shanjida Sultana, Uttam Kumar Roy, and Jubayer Al Mahmud. "A review on video classification with methods, findings, performance, challenges, limitations and future work." Jurnal Ilmiah Teknik Elektro Komputer dan Informatika 6, no. 2 (2020): 47-57.

[4] Karpathy, Andrej, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-scale video classification with convolutional neural networks." In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 1725-1732. 2014.

[5] Yao, Lirong, and Yazhuo Guan. "An improved LSTM structure for natural language processing." In *2018 IEEE international conference of safety produce informatization (IICSPI)*, pp. 565-569. IEEE, 2018.

[6] https://huggingface.co/docs/transformers/en/model_doc/roberta