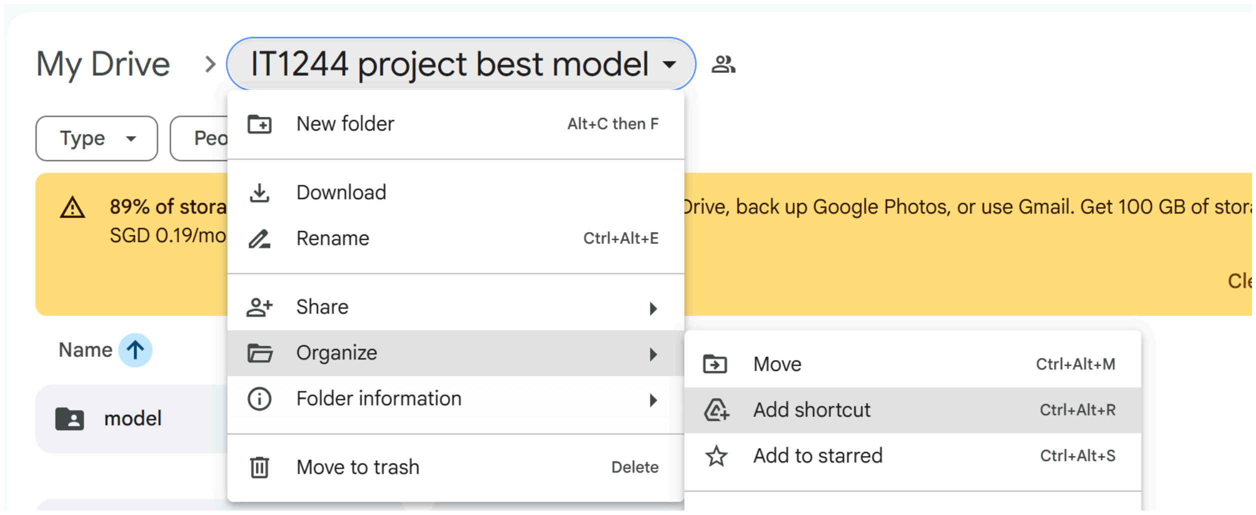
INSTRUCTIONS TO RUN AND TEST MODEL

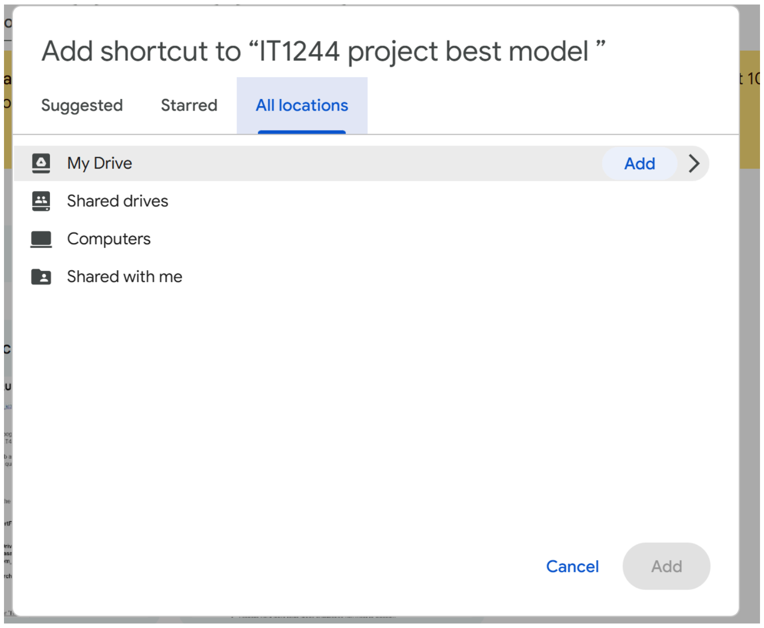
Google Drive Folder:
https://drive.google.com/drive/folders/1-nO7be_kl3Y9nEbxov89QGwnWM8JwOSs?usp=sharing

After obtaining the shared Google Drive folder right click the file name and click add shortcut as shown below.



89% of stora
SGD 0.19/mo
Drive, back up Google Photos, or use Gmail. Get 100 GB of stor

Click “All locations” and add it to “My Drive”



Steps to run model

1. Open the notebook main.ipynb in Google Colab (or any Python environment).
    - Make sure your runtime is on T4 GPU (Top right corner to change)

2. Mount google drive (This gives collab access to your Google Drive: Google Drive has to be used here as our model is quite large) Use:

```
from google.colab import drive
drive.mount('/content/drive')
```

3. Load the model and tokenizer from the Google Drive folder

```
import torch
from transformers import DistilBertForSequenceClassification,
DistilBertTokenizer

model_path = "/content/drive/MyDrive/IT1244 project best model /model"
model = DistilBertForSequenceClassification.from_pretrained(model_path)
tokenizer = DistilBertTokenizer.from_pretrained(model_path)

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)
model.eval()
```

***pls take note there is a space after “IT1244 project best model”
Also for the model_path it may be different depending on the directory of your Google Drive.

4. Prepare your input texts (book summaries you want to classify).
Eg:

```
texts = [
   "A young wizard discovers his magical powers and attends a school of
magic.",
   "A historical account of the life of a famous king in the 17th century."
]
```

5. Tokenize inputs:

```
max_len = 256
inputs = tokenizer(
    texts,
    truncation=True,
    padding="max_length",
    max_length=max_len,
    return_tensors="pt"
)
inputs = {k: v.to(device) for k, v in inputs.items()}
```

6. Run inference using the model to get predicted genre indices.

```
with torch.no_grad():
    outputs = model(**inputs)
    predictions = torch.argmax(outputs.logits, dim=1).cpu().numpy()

print("Predicted class indices:", predictions)
```

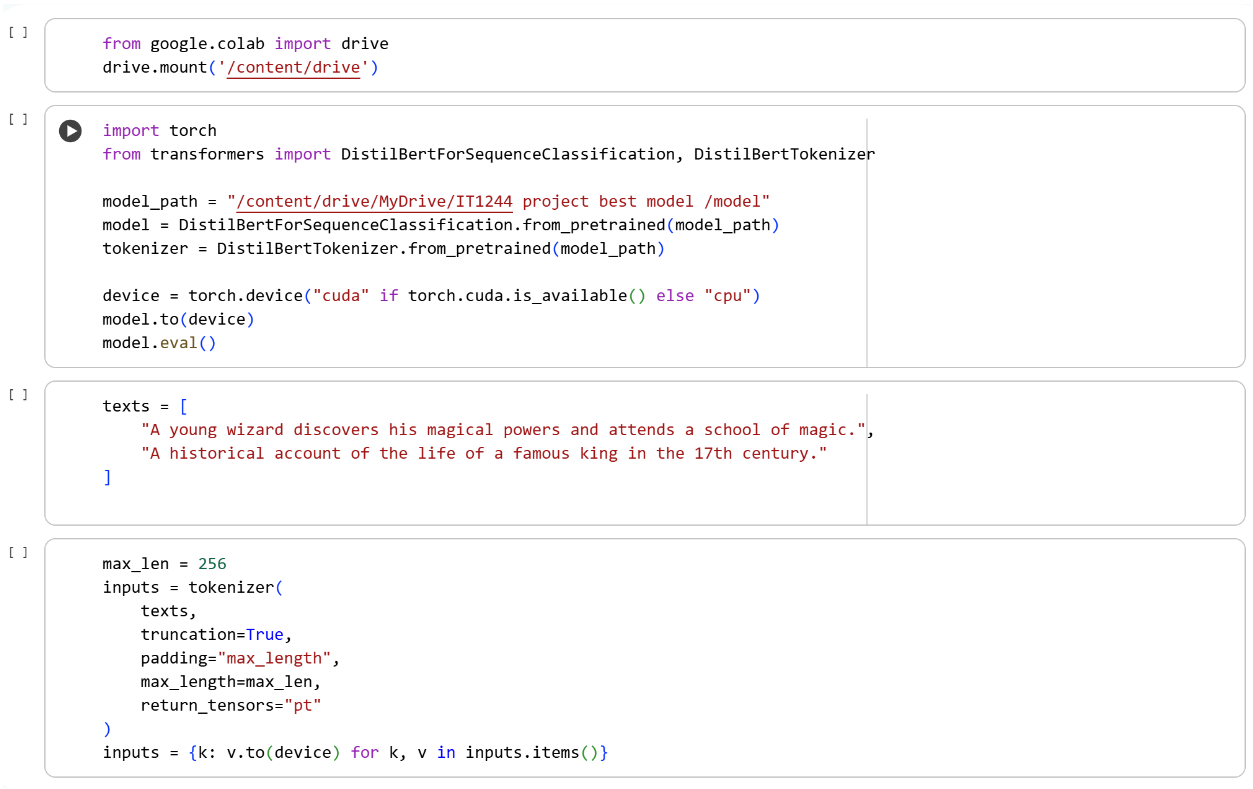
7. Map the predictions back to the corresponding genre names (please use this genre mapping as we changed it slightly when doing data cleaning).

```
genre_mapping = {
    0: "crime",
    1: "fantasy",
    2: "history",
    3: "horror",
    4: "psychology",
    5: "romance",
    6: "science",
    7: "sports",
    8: "thriller",
    9: "travel"
}

predicted_genres = [genre_mapping[int(p)] for p in predictions]
print("Predicted genres:", predicted_genres)
```

8. Run all cells to get predicted genres

How it should look like on Google Collab



```
[ ]
from google.colab import drive
drive.mount('/content/drive')
```

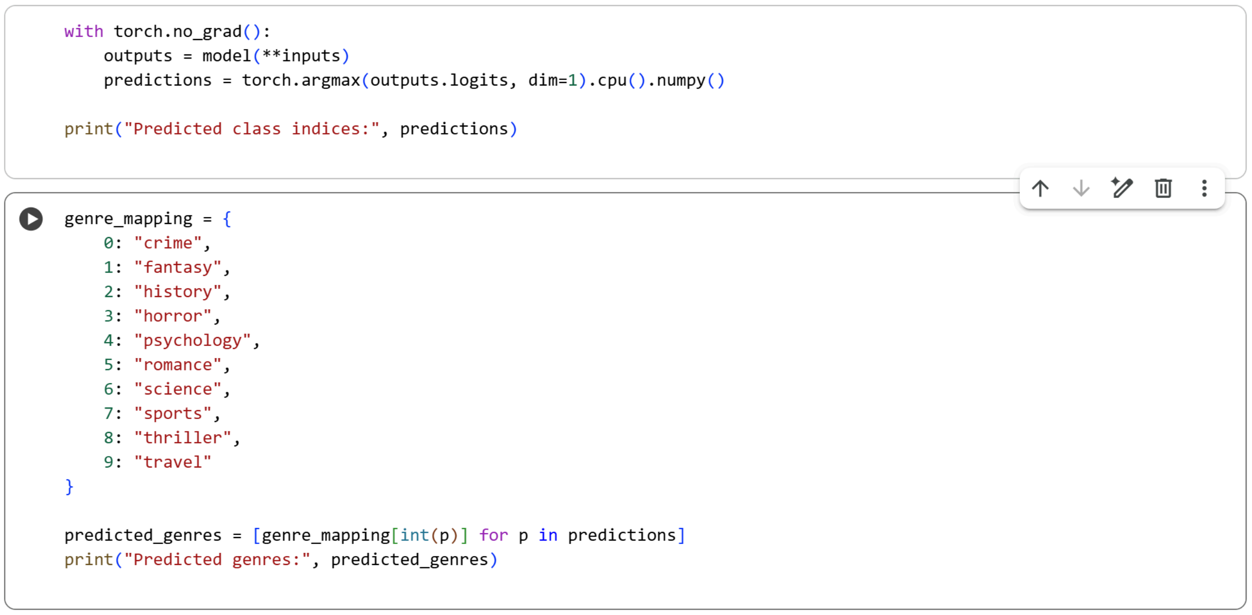
```
[ ]
import torch
from transformers import DistilBertForSequenceClassification, DistilBertTokenizer

model_path = "/content/drive/MyDrive/IT1244 project best model /model"
model = DistilBertForSequenceClassification.from_pretrained(model_path)
tokenizer = DistilBertTokenizer.from_pretrained(model_path)

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)
model.eval()
```

```
[ ]
texts = [
    "A young wizard discovers his magical powers and attends a school of magic.",
    "A historical account of the life of a famous king in the 17th century."
]
```

```
[ ]
max_len = 256
inputs = tokenizer(
    texts,
    truncation=True,
    padding="max_length",
    max_length=max_len,
    return_tensors="pt"
)
inputs = {k: v.to(device) for k, v in inputs.items()}
```



```
with torch.no_grad():
    outputs = model(**inputs)
    predictions = torch.argmax(outputs.logits, dim=1).cpu().numpy()

print("Predicted class indices:", predictions)
```

```
genre_mapping = {
    0: "crime",
    1: "fantasy",
    2: "history",
    3: "horror",
    4: "psychology",
    5: "romance",
    6: "science",
    7: "sports",
    8: "thriller",
    9: "travel"
}

predicted_genres = [genre_mapping[int(p)] for p in predictions]
print("Predicted genres:", predicted_genres)
```