

**Car-ing is sharing**, an auto dealership company for car sales and rental, is taking their services to the next level thanks to **Large Language Models (LLMs)**.

As their newly recruited AI and NLP developer, you've been asked to prototype a chatbot app with multiple functionalities that not only assist customers but also provide support to human agents in the company.

The solution should receive textual prompts and use a variety of pre-trained Hugging Face LLMs to respond to a series of tasks, e.g. classifying the sentiment in a car's text review, answering a customer question, summarizing or translating text, etc.

## sentiment analysis

```
# Import necessary packages
import pandas as pd
import torch

# Your code makes Transformers less chatty, hiding unnecessary info messages while
keeping warnings and errors visible.
from transformers import logging
logging.set_verbosity(logging.WARNING)
```

```
# Read the reviews and assign the review and class into two lists.
df = pd.read_csv("data/car_reviews.csv", sep=';')
print(df.head())

reviews = df['Review'].tolist()
real_labels = df['Class'].tolist()
```

	Review	Class
0	I am very satisfied with my 2014 Nissan NV SL....	POSITIVE
1	The car is fine. It's a bit loud and not very ...	NEGATIVE
2	My first foreign car. Love it, I would buy ano...	POSITIVE
3	I've come across numerous reviews praising the...	NEGATIVE
4	I've been dreaming of owning an SUV for quite ...	POSITIVE

## Instruction 1 : Sentiment classification

```
# Sentiment analysis pipeline.  
from transformers import pipeline  
  
sentiment_pipeline = pipeline(task = "sentiment-analysis" , model = "distilbert-  
base-uncased-finetuned-sst-2-english")
```

```
predicted_labels = sentiment_pipeline(reviews)  
print(predicted_labels)
```

Device set to use cpu

```
[{'label': 'POSITIVE', 'score': 0.929397702217102}, {'label': 'POSITIVE', 'score':  
0.8654273152351379}, {'label': 'POSITIVE', 'score': 0.9994640946388245}, {'label':  
'NEGATIVE', 'score': 0.9935314059257507}, {'label': 'POSITIVE', 'score':  
0.9986565113067627}]
```

```
# converting the labels into binary output for evaluation.  
predictions = [1 if pred['label'] == 'POSITIVE' else 0 for pred in predicted_labels]  
print(predictions)
```

```
[1, 1, 1, 0, 1]
```

```
# Converting the true labels into binary list.  
true_labels =[1 if value=='POSITIVE' else 0 for value in real_labels ]  
print(true_labels)
```

```
[1, 0, 1, 0, 1]
```

```
print(predictions)  
print(true_labels)
```

```
[1, 1, 1, 0, 1]  
[1, 0, 1, 0, 1]
```

How likely are you to recommend DataLab to a friend or co-worker?

Not at all likely



Extremely likely

```
# Evaluating the result with accuracy and f1_score .
import evaluate
accuracy = evaluate.load("accuracy")
f1 = evaluate.load("f1")

accuracy_result_dic = accuracy.compute(references = true_labels, predictions =
predictions)
accuracy_result = accuracy_result_dic['accuracy']

f1_result_dic = f1.compute(references = true_labels , predictions = predictions )
f1_result = f1_result_dic['f1']

print ( accuracy_result , f1_result)
```

0.8 0.8571428571428571

## Instruction 2 : Translation

```
# Load translation LLM into a pipeline and translate car review
first_review = reviews[0]
translator = pipeline("translation", model="Helsinki-NLP/opus-mt-en-es")
translated_review = translator(first_review, max_length=27)[0]['translation_text']
print(f"Model translation:\n{translated_review}")
```

Device set to use cpu

Your input\_length: 365 is bigger than 0.9 \* max\_length: 27. You might consider increasing your max\_length manually, e.g. translator(..., max\_length=400)

Model translation:

Estoy muy satisfecho con mi 2014 Nissan NV SL. Uso esta furgoneta para mis entregas de negocios y uso personal.

```
# Load reference translations from file
with open("data/reference_translations.txt", 'r') as file:
    lines = file.readlines()
references = [line.strip() for line in lines]
print(f"Spanish translation references:\n{references}")
```

How likely are you to recommend DataLab to a friend or co-worker?

Not at all likely



Extremely likely

powered by InMoment

```
# Load and calculate BLEU score metric
bleu = evaluate.load("bleu")
bleu_score = bleu.compute(predictions=[translated_review], references=[references])
print(bleu_score['bleu'])
```

0.6022774485691839

### Instruction 3 : QNA

```
# Second review from the dataset .
second_review = reviews[1]
print(second_review)
```

The car is fine. It's a bit loud and not very powerful. On one hand, compared to its peers, the interior is well-built. The transmission failed a few years ago, and the dealer replaced it under warranty with no issues. Now, about 60k miles later, the transmission is failing again. It sounds like a truck, and the issues are well-documented. The dealer tells me it is normal, refusing to do anything to resolve the issue. After owning the car for 4 years, there are many other vehicles I would purchase over this one. Initially, I really liked what the brand is about: ride quality, reliability, etc. But I will not purchase another one. Despite these concerns, I must say, the level of comfort in the car has always been satisfactory, but not worth the rest of issues found.

How likely are you to recommend DataLab to a friend or co-worker?

Not at all likely

0

1

2

3

4

5

6

7

8

9

10

Extremely likely

```
# A pre-trained question-answering model.

qa_pipeline = pipeline(
    "question-answering",
    model="deepset/minilm-uncased-squad2"
)

context = reviews[1]
question = "What did he like about the brand?"

answer_dic = qa_pipeline(question=question, context=context)
answer = answer_dic['answer']
print(answer)
```

Some weights of the model checkpoint at deepset/minilm-uncased-squad2 were not used when initializing BertForQuestionAnswering: ['bert.pooler.dense.bias', 'bert.pooler.dense.weight']  
- This IS expected if you are initializing BertForQuestionAnswering from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from a BertForPreTraining model).  
- This IS NOT expected if you are initializing BertForQuestionAnswering from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenceClassification model).  
Device set to use cpu

ride quality, reliability

## Instruction 4 : summarization .

How likely are you to recommend DataLab to a friend or co-worker?

Not at all likely



powered by InMoment

```
# 5th review
print(f"Main text review:\n {reviews[4]}")
```

Main text review:

I've been dreaming of owning an SUV for quite a while, but I've been driving cars that were already paid for during an extended period. I ultimately made the decision to transition to a brand-new car, which, of course, involved taking on new payments. However, given that I don't drive extensively, I was inclined to avoid a substantial financial commitment. The Nissan Rogue provides me with the desired SUV experience without burdening me with an exorbitant payment; the financial arrangement is quite reasonable. Handling and styling are great; I have hauled 12 bags of mulch in the back with the seats down and could have held more. I am VERY satisfied overall. I find myself needing to exercise extra caution when making lane changes, particularly owing to the blind spots resulting from the small side windows situated towards the rear of the vehicle. To address this concern, I am actively engaged in making adjustments to my mirrors and consciously reducing the frequency of lane changes. The engine delivers strong performance, and the ride is really smooth.

```
# summarization pipeline.
summarizer = pipeline("summarization", model="sshleifer/distilbart-cnn-12-6")
text = reviews[4]

summarized_text_dic = summarizer(text, max_length=53, do_sample=False)
summarized_text = summarized_text_dic[0]['summary_text']
# [0] because it return list. in previous instruction we get single dictionary .
print(summarized_text)
```

Device set to use cpu

Your min\_length=56 must be inferior than your max\_length=53.

Nissan Rogue provides the desired SUV experience without burdening me with an exorbitant payment . Handling and styling are great; I have hauled 12 bags of mulch in the back with the seats down and could have held more . The engine delivers strong performance

How likely are you to recommend DataLab to a friend or co-worker?

Not at all likely



Extremely likely