Evaluating and Testing Generative AI Models

In this module, you learn to ...

01 Evaluate GenAI applications

02 Write and run automated tests for GenAI apps

Topics

01	Evaluating GenAI Models and Apps
02	Testing

Enterprise Challenge: Assessing Model Output Quality

- Quality of output can sometimes be subjective
  - “Write me a clever social media post about our new product line”
- LLMs will return incorrect responses
  - Called hallucinations
  - Need to verify the results before using them
- LLMs do not always return the exact same results
  - This makes it challenging to write automated tests

Which is the best answer?

Summary 1

You can restore a user account (including administrator accounts) up to 20 days after deleting it. After 20 days, the data is gone and you can't restore it.
To restore a user, you must be signed in as a super administrator.
On the Admin console home page, you can see the total number of recently deleted users in the Users section. To go directly to the deleted users page, click that number, and then go to step 4 below.
In the Admin console, go to Menu DirectoryUsers.
Click More optionsRecently deleted users.
Point to the userclick Recover.
Review the confirmation noticeclick Continue.
Select an organizational unit to assign the user to.
Click Recover.
Note: Usually, it takes up to 24 hours for this change to take effect, but in rare cases it can take up to 5 days.

Summary 2

You can restore a user account (including administrator accounts) up to 20 days after deleting it. After 20 days, the data is gone and you can't restore it.
To restore a user, you must be signed in as a super administrator.

1. Sign in to the Admin console.
2. Go to Menu > Directory > Users.
3. Click More options > Recently deleted users.
4. Point to the user and click Recover.
5. Review the confirmation notice and click Continue.
6. Select an organizational unit to assign the user to.
7. Click Recover.

Note: Usually, it takes up to 24 hours for this change to take effect, but in rare cases it can take up to 5 days.

Summary 3

- Restoring user accounts possible only for super administrator and only up to 20 days after deleting it

- Process (usually takes 24h, but can take up to 5 business days):
1. Go to Admin console > Menu > Directory > Users
2. Click More options > Recently deleted users
3. Point to the user and click Recover.
4. Review the confirmation notice and click Continue.
5. Click an organizational unit to assign the user.
6. Select Recover.

We asked the LLM for the gcloud commands to create a VPC with subnets in Iowa and Virginia: is it correct?

```
gcloud compute networks create my-network \
    --subnet-mode custom \
    --bgp-routing-mode global \
    --mtu 1460 \
    --region us-central1

gcloud compute networks subnets create my-subnet-iowa \
    --network my-network \
    --region us-central1 \
    --range 10.0.0.0/24

gcloud compute networks subnets create my-subnet-virginia \
    --network my-network \
    --region us-east1 \
    --range 10.1.0.0/24
```

For Gen AI tasks, evaluation is not a solved problem

Lack of Data

Customers who are using pretrained models generally do not have an adequate dataset to run evaluation

Lack of Metrics

Many generative tasks do not have well-defined metrics.

Enterprise customers have different criteria with respect to evaluating generated content.

Large Decision Space

Model development transitions from model training to pretrained model selection, customization options, and in-context learning.

Each option is expensive to explore.

Large evaluation decision space

Evaluate & Monitor

Select a model based on **performance** and **efficiency**

Choose the most **efficient** and **effective** tuning method and evaluate performance

Pick the best **performance** / **cost** tradeoff for your use case

Understand **trustworthiness**, i.e. safety, bias, and factuality

Apply different **Prompt Engineering** techniques

Choosing Evaluation Metrics

- Classification models
  - Accuracy
  - Precision
  - Recall
  - F1 score (binary classification)
  - Macro-F1 or Micro-F1 (multi-class classification)
- Text generation models
  - ROUGE-L
  - BLEU score
- Q&A
  - Exact match

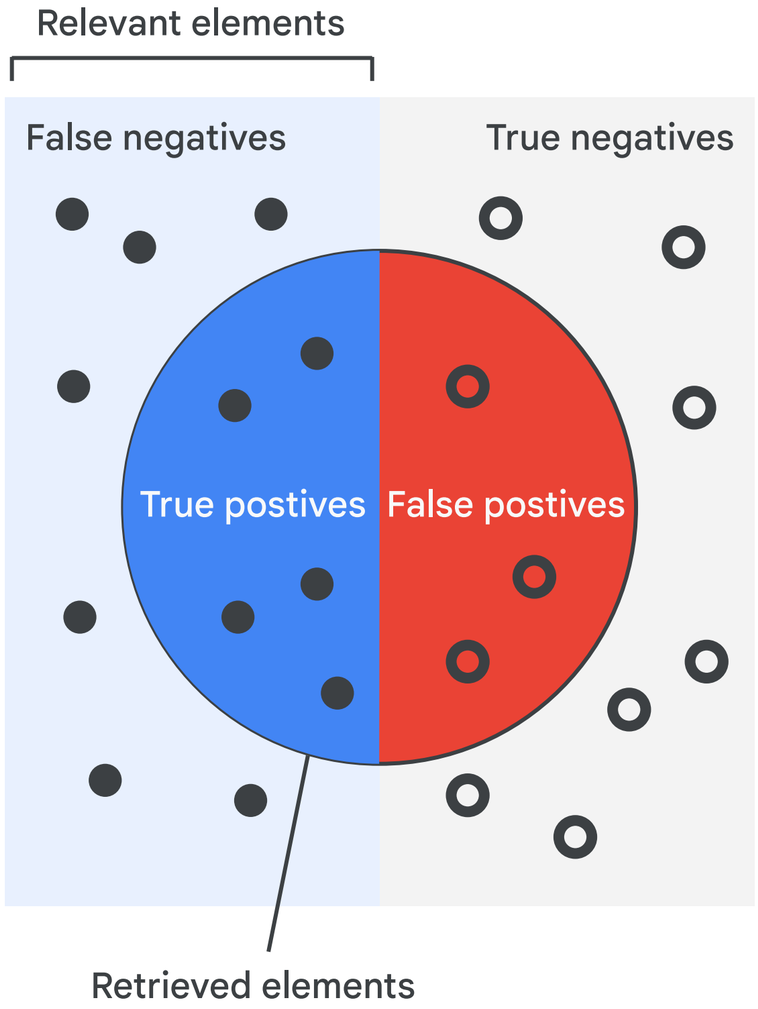
Accuracy

- The percentage of examples the model predicted correctly
  - `Accuracy = Correct Predictions / # of Examples`
- While accuracy seems like the perfect metric, it doesn't work when datasets are skewed
  - If 99% of transactions are not fraudulent, then a model would be 99% accurate if it just predicted all transactions were good

Precision and recall (for classification)

- Precision is the proportion of true predictions that were correct
  - `Precision = TP / (TP + FP)`
- Recall is the proportion of positives were identified correctly
  - `Recall = TP / (TP + FN)`
- In a classification model, higher precision means lower recall and vice versa
- You can tune the classification threshold to control whether you get more or less false positives
  - In a spam filter, you would rather have spam end up in the inbox than important emails end up in the junk folder
  - If you were predicting an illness, you would rather have false positives than false negatives

Precision and recall illustrated



How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}}$$

F1 score combines precision & recall into one metric

- Calculated from the **precision** and **recall** of the test
- Still requires a threshold to be set (as opposed to AUC)
- The F1 score is the harmonic mean of the precision and recall
  - Symmetrically represents both precision and recall in one metric
- The highest possible value of an F-score is 1, indicating perfect precision and recall
- The lowest possible value is 0, if either precision or recall are zero
- So, closer to 1 is better

$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

Or

$$\text{F1 Score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

F1 score is used for binary classification, use Macro-F1 or Micro-F1 for multi-class classification

- Macro-F1 is calculated by first computing the F1 score for each class independently, and then averaging the scores
  - Gives equal weight to each class

$$\text{Macro-F1} = \frac{\sum_{i=1}^{n} F1_i}{n}$$

- Micro-F1 is calculated by aggregating the contributions of all classes to compute the average precision, recall, and F1 score
  - Calculates precision and recall globally over all classes, and then uses these values to compute the F1 score

$$\text{Micro-F1} = 2 \cdot \frac{\text{Micro-Precision} \cdot \text{Micro-Recall}}{\text{Micro-Precision} + \text{Micro-Recall}}$$

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is used for summarization and translation

- Compare an automatically produced summary or translation against a reference (human-produced) summary or translation
- **ROUGE-N**: Overlap of n-grams between the system and reference summaries
- **ROUGE-L**: Longest Common Subsequence (LCS) based statistics
- **ROUGE-W**: Weighted LCS-based statistics that favors longer consecutive sequences
- others...

ROUGE-L: Longest Common Subsequence

- A longest common subsequence (LCS) is the longest subsequence common to all sequences in a set of sequences (often just two sequences)
- Consider the sequences (ABCD) and (ACBAD), they have:
  - 5 length-2 common subsequences: (AB), (AC), (AD), (BD), and (CD)
  - 2 length-3 common subsequences: (ABD) and (ACD)
  - So (ABD) and (ACD) are their longest common subsequences
- ROUGE_L returns a value between 0 and 1
  - 1 means the sequences are the same
  - 0 means the sequences have nothing in common
  - Closer to 1 is better

BLEU (Bilingual Evaluation Understudy) is a metric used to evaluate the quality of machine-generated text

- Compares generated text to reference human-generated text
  - Used for tasks such as machine translation, text summarization, and image captioning
- Ranges from 0 to 1
  - 0 indicates no overlap between the machine-generated text and the reference text
  - 1 indicates a perfect match

BLEU - Interpretation

BLEU Score [%]	Interpretation
< 10	Almost useless
10 - 19	Hard to get the gist
20 - 29	The gist is clear, but has grammatical errors
30 - 39	Understandable to good
40 - 49	High quality
50 - 60	Very high quality
> 60	Quality often better than human

BLEU scores from different corpora and languages **cannot** be directly compared.

Exact Match measures the percentage of predictions that match any one of the ground truth answers exactly

- For each question and answer pair, if the characters of the model's prediction exactly match the characters of (one of) the True Answer(s), then EM = 1, otherwise EM = 0
- This is a strict all-or-nothing metric
  - Being off by a single character results in a score of 0
- This metric is limited in that it outputs the same score for something that is completely wrong as for something that is correct except for a single character
- These traditional NLP metrics looking for exact matches are good for short completions or short phrases of Question-Answering results, but are harder to rely on for longer answers where there can be many ways to express something well.

Summary of Metrics

Metric	Uses cases
Accuracy	Classification where there are balanced classes
Precision	Classification where there are imbalanced classes (one class dominates) You want to minimize false positives (a spam filter)
Recall	Classification where there are imbalanced classes You want to minimize false negatives (test for a disease)
F1 Score	Classification Combines precision and recall into a single metric
ROUGE	Text summarization and translation
BLEU	Text generation, summarization, translation
Exact Match	Text generation, Q&A, Classification where the answer is unambiguous (only one right answer)

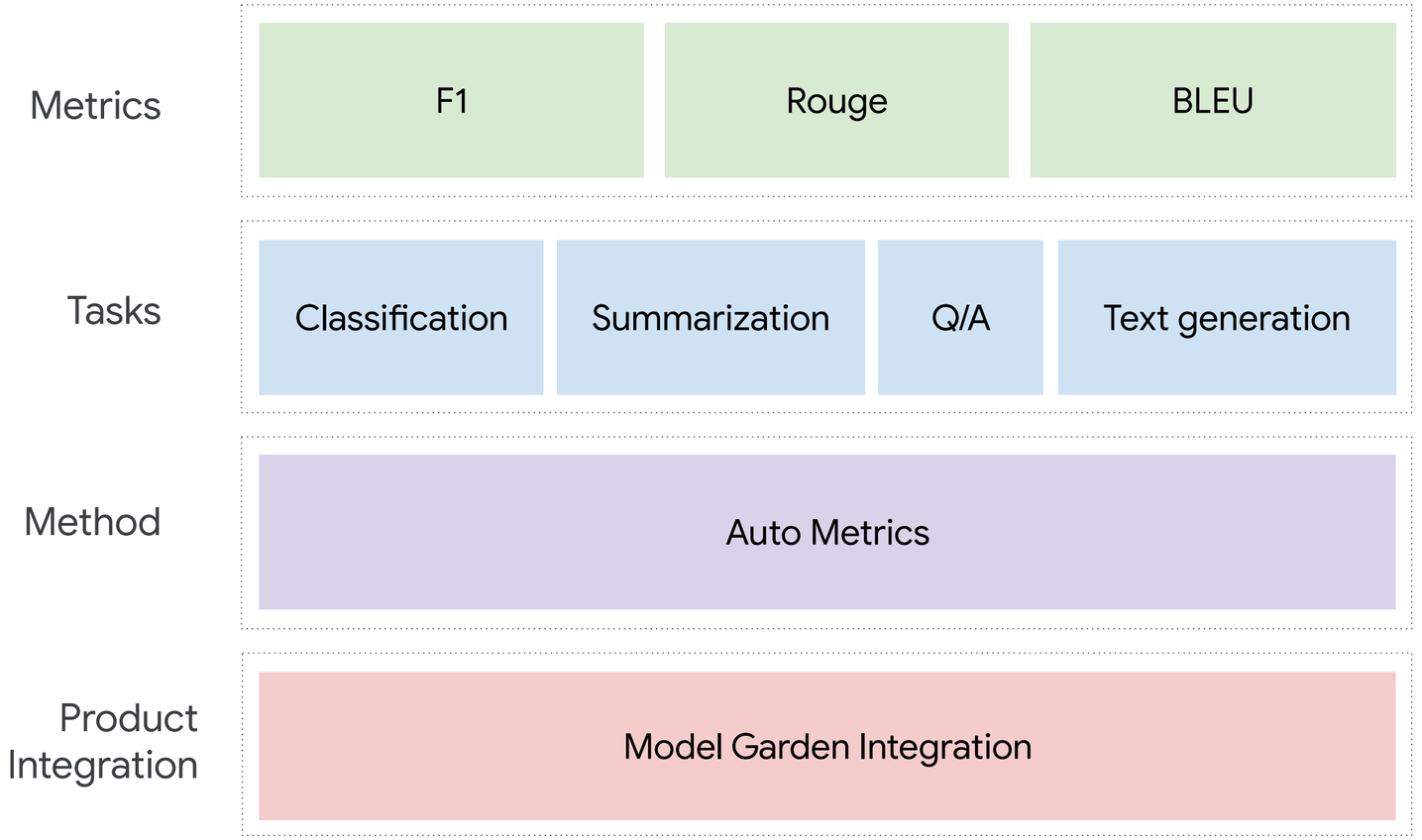
Vertex AI LLM Evaluation Services

Automatic Metrics

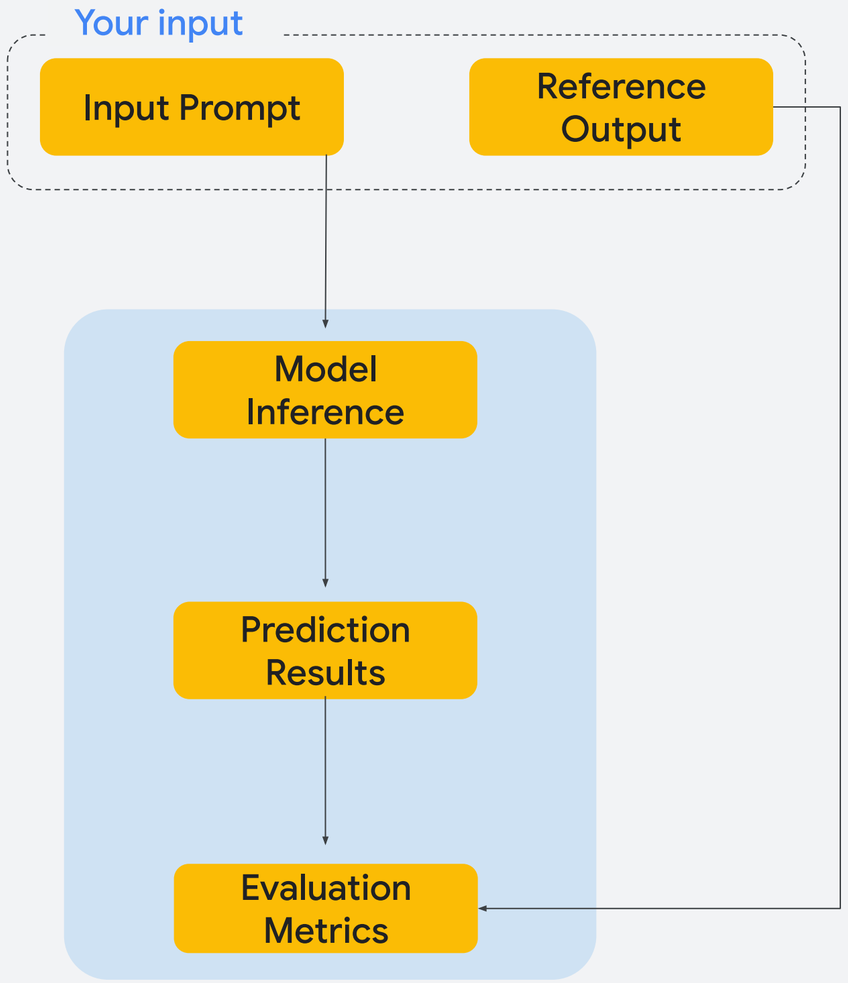
rougeLSum	0.1345
bleu	0.763

Assess the performance of a model with task-specific metrics computed based on reference data

- Fast and efficient
- Standard method used in academia and many open benchmarks



Automatic metrics



1. **Fast and efficient**
   Access a range of task-specific metrics for fast and low cost evaluation of your model

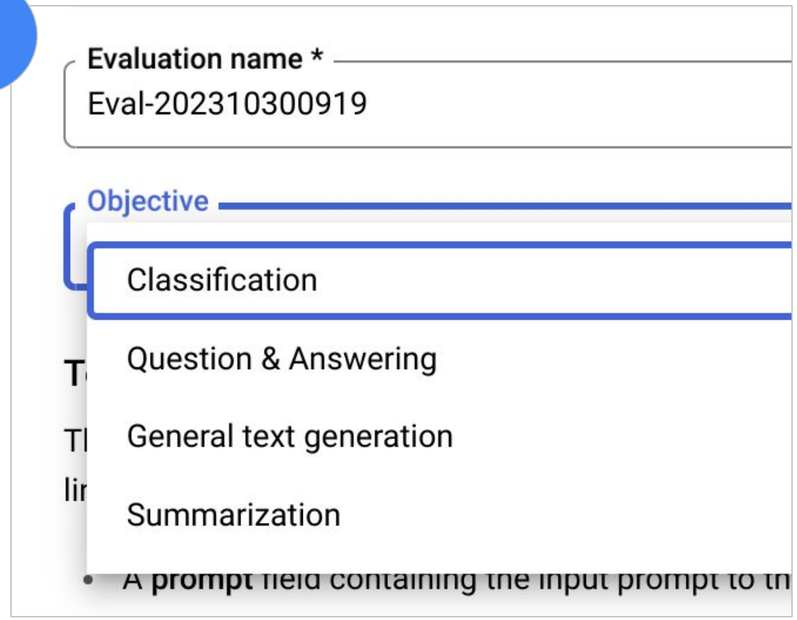
2. **Reference data required**
   Evaluate model results based on [input prompt, output response] pairs

3. **Benchmark evaluation method**
   Standard method used in academic research and many open benchmarks, with widely-used metrics for several Gen AI tasks

You can create an evaluation for a trained or fine-tuned model using Vertex AI Model Registry

1. Select the model to view its Details and click Create Evaluation
2. Choose the model objective
3. Set the location of the test dataset
   a. JSON-L file with `prompt` and `ground_truth` fields

2



3

Test dataset

The test dataset is a JSONL file that contains model prompt and ground truth (one per line). Each line in the file contains one example:

- A **prompt** field containing the input prompt to the model
- A **ground_truth** field to compare the model output against

gs:// Source path * BROWSE

Vertex AI LLM evaluation services can be used for the following tasks

Task	Evaluation Metric
Classification	Micro-F1, Macro-F1, Per class F1
Summarization	ROUGE-L
Question answering	Exact Match
Text generation	BLEU, ROUGE-L

- For the evaluation dataset, at least 10 prompt-ground truth pairs are recommended
  - The more examples given, the more meaningful the results
- Examples are written in JSON-L format
  - The prompt field represents the question you want to test
  - The ground truth field represents the ideal response
- See: https://cloud.google.com/vertex-ai/docs/generative-ai/models/evaluate-models#question-answering

Topics

01 Evaluating GenAI Models and Apps

02 Testing

You should unit test your LLM applications

- Unit testing is straightforward with classification examples where there is only one well-defined answer
  - Compare the actual result to the expected result
  - Thorough unit testing will quickly expose where prompt-tuning, more examples, and model fine-tuning might be needed
- Unit testing is more difficult in cases where there isn't a single right answer
  - One strategy is to use the model to determine if two answers are fundamentally equivalent without being exactly the same

Classification Example

```
def getPositiveOrNegative(prompt):
  response = model.predict(
    """Context: You look at messages and categorize them as
    positive, negative, or neutral.

  Output only Positive, Negative, or Neutral.

  Message: {0}.
  Output: """.format(prompt),
    **parameters
    )

  return response.text.strip()
```

Classification example unit tests

- Get the response from the model
- Assert that the response is what was expected

```
import unittest

class TestPositiveOrNegative(unittest.TestCase):

    def test_getPositiveOrNegative1(self):
      response = getPositiveOrNegative("Dinner was Great")
      self.assertEqual(response, "Positive")

    def test_getPositiveOrNegative2(self):
      response = getPositiveOrNegative("That broccoli was undercooked and cold")
      self.assertEqual(response, "Negative")

    def test_getPositiveOrNegative3(self):
      response = getPositiveOrNegative("We went to the new italian place for dinner")
      self.assertEqual(response, "Neutral")
```

This test failed

- The tester expected Neutral, the model thought the text was Positive
  - Who is right the tester or the model
  - If the tester is right, then we need some examples in the prompt or fine-tuning of the model

```
    def test_getPositiveOrNegative3(self):
      response = getPositiveOrNegative("We went to the new italian place for dinner")
      self.assertEqual(response, "Neutral")
```

```
test_getPositiveOrNegative1 (__main__.TestNotebook) ... ok
test_getPositiveOrNegative2 (__main__.TestNotebook) ... ok
test_getPositiveOrNegative3 (__main__.TestNotebook) ... FAIL
```

Text generation is harder to test because there are many correct answers

```
def writeTweet(prompt):
  response = model.predict(
    """Context: You write Tweets for the Marketing Department at Luigi's Italian Cafe.

    1. Keep your Tweets below 100 characters
    2. Include the hashtag #EatAtLuigis at the end of every tweet

  Input: Write me a tweet for our All You Can eat Spaghetti Wednesdays deal.
  Output: Spaghetti Wednesday is here! Come get all you can eat spaghetti and meatballs for only $9.99! #EatAtLuigis

  Input: {0}
  Output: """.format(prompt),
    **parameters
    )

  return response.text.strip()
```

One strategy is to ask the LLM if the response and the expected response are the same

- This might need some prompt engineering as well

```
def are_tweets_the_same(tweet1, tweet2):
  response = model.predict(
    """Compare the following Tweets. Are they fundamentally the same?

    Only return Yes or No

    Tweet 1: {0}
    Tweet 2: {1}
    Output:

 """.format(tweet1, tweet2),
    **parameters
    )

  return response.text.strip()
```

Unit tests for text generation

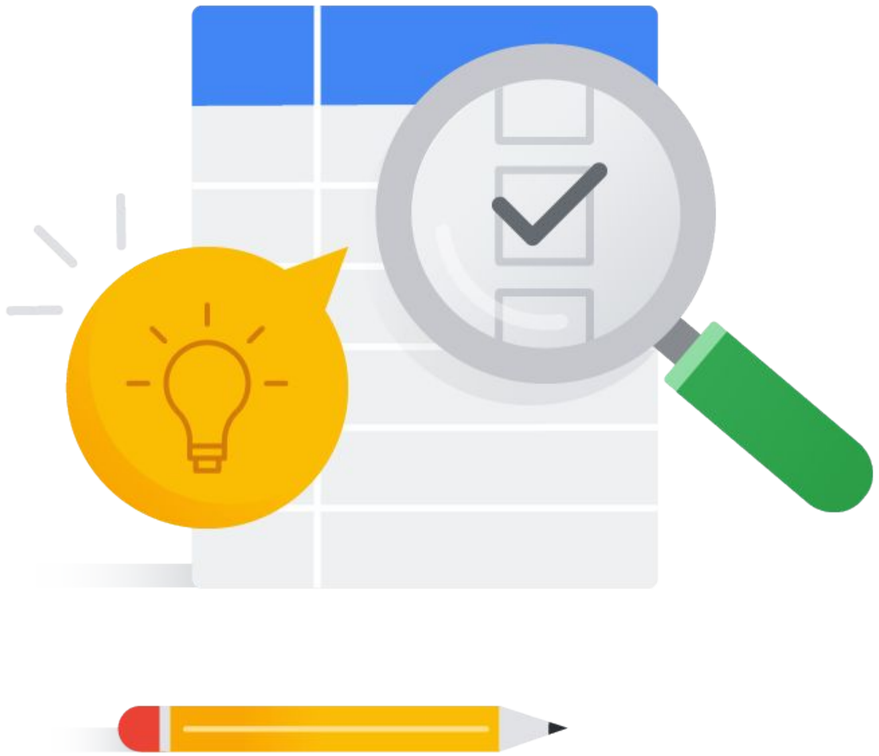
- In the example below, the second test is failing on purpose

```
class TestTextGeneration(unittest.TestCase):

    def test_tweet_results_1(self):
      actual_result = writeTweet("Write a tweet about our half-price bottles of wine every Thursday")
      expected_result = """Thirsty Thursday is here! Come enjoy half-price bottles of
      wine every Thursday at Luigi's! #EatAtLuigis"""
      same = are_tweets_the_same(actual_result, expected_result)
      self.assertEqual(same, "Yes")

    def test_tweet_results_Should_fail(self):
      actual_result = writeTweet("Write a tweet about our half-price bottles of wine every Thursday")
      expected_result = """Halloween party at the community center! Dress up and come out for a
      night of fun, food, and games. #Halloween"""
      same = are_tweets_the_same(actual_result, expected_result)
      self.assertEqual(same, "Yes")
```

Lab

30 min

Lab: Evaluating ROUGE-L Text Similarity Metric



In this module, you learned to ...

01 Evaluate GenAI applications

02 Write and run automated tests for GenAI apps

Questions and answers

Quiz question

What might be a good evaluation metric for a Classification problem?

A: RMSE

B: F1

C: ROUGE-L

D: BLEU score

Quiz question

What might be a good evaluation metric for a Classification problem?

A: RMSE

B: F1

C: ROUGE-L

D: BLEU score

Quiz question

What might be a good evaluation metric for a text generation problem? (Choose two)

A: RMSE

B: F1

C: ROUGE-L

D: BLEU score

Quiz question

What might be a good evaluation metric for a text generation problem? (Choose two)

A: RMSE

B: F1

C: ROUGE-L

D: BLEU score

Quiz question

When using F1, ROUGE-L, or BLEU to evaluate model versions, how do you know which is better?

A: Closest to 0 is best

B: Closest to 1 is best

C: The greater the number the better

D: The smaller the number the better

Quiz question

When using F1, ROUGE-L, or BLEU to evaluate model versions, how do you know which is better?

A: Closest to 0 is best

B: Closest to 1 is best

C: The greater the number the better

D: The smaller the number the better


Google Cloud