Produce a classification model

Tom Rochette <tom.rochette@coreteks.org>

August 30, 2025 — 861fb9d0

1 Prompt

We want to produce a new classification model and the associated infrastructure to serve real-time predictions. What should our next steps be?

2 Response

There are several steps you can take to produce a new classification model and associated infrastructure for serving real-time predictions:

Define the problem: Define the problem you want to solve with your classification model. What are the inputs to your model, and what are the outputs you want to predict? For example, you may want to classify images into different categories, or classify text into different sentiment categories.

Collect and preprocess data: Collect data that is representative of the problem you want to solve. Preprocess the data by cleaning and transforming it so that it is in a format that can be used for training the model.

Select a machine learning algorithm: Select a machine learning algorithm that is appropriate for your problem. For example, you may want to use a decision tree, random forest, or support vector machine.

Train the model: Train the model using the preprocessed data and the selected machine learning algorithm. Split the data into training and testing sets, and use cross-validation to evaluate the model's performance.

Tune hyperparameters: Tune the hyperparameters of the machine learning algorithm to improve the performance of the model. Hyperparameters are parameters that are set before training the model, such as the learning rate or the number of hidden layers in a neural network.

Evaluate the model: Evaluate the model's performance on a holdout dataset. Use metrics such as accuracy, precision, recall, and F1-score to evaluate the model's performance.

Deploy the model: Deploy the model and associated infrastructure to serve real-time predictions. This may involve setting up a server, creating an API, and integrating the model into your application.

Monitor and update the model: Monitor the model's performance in real-time, and update the model as needed to improve its performance. This may involve retraining the model with new data or tweaking the hyperparameters.

3 Observations

Given a high level request, ChatGPT correctly responds with a high level response. The next logical step is to start answering the first step with more details. One thing that ChatGPT does not do is prompt the user for further detail. It offers its whole answer and then leaves it to the user to decide what to do next. Another potential observation is that we might have reached the maximum token limit for the response and thus their might be additional details that ChatGPT would have provided if it had more tokens to work with.

For now we will give details to the first step ChatGPT provided and will see how it responds.

4 Prompt

I want to be able to classify whether a person prefers one brand over another.
I might know about the age of that person, the country where they live, how much time they use a given app per day, how many friends they have on this app.
As output, I want to classify whether they would prefer brand A or brand B.

5 Response

Great! Here's an overview of the steps you could take to create a classification model to predict whether a person prefers brand A or brand B based on their age, country, app usage, and number of friends:

Data collection: Collect data from a representative sample of people that includes their age, country, app usage, number of friends, and which brand they prefer (A or B).

Data preprocessing: Clean and preprocess the data by removing any duplicates, filling in missing values, and converting categorical variables (like country) into numerical features.

Feature engineering: Create additional features from the data that might help improve the model's performance. For example, you could create features like "average app usage per friend" or "age group" based on ranges of ages.

Split the data: Split the data into training and testing sets. This is important so that you can evaluate the model's performance on data that it hasn't seen before.

Choose a classification algorithm: Choose an appropriate algorithm for your data. Some popular choices include logistic regression, decision trees, and random forests.

Train the model: Train the classification model on the training data. Use the testing data to evaluate the model's performance and adjust the model's hyperparameters to improve its accuracy.

Deploy the model: Once the model is trained and performs well on the testing data, deploy it and set up an API to receive input data and provide predictions in real-time.

Monitor and update the model: Continuously monitor the model's performance and update it as needed. As new data becomes available, retrain the model to improve its accuracy over time.

Remember to keep in mind the ethical considerations of using personal data for classification purposes, and ensure that you comply with all relevant data protection laws and regulations.

6 Observations

Again, ChatGPT's answer is high level. It mostly reused its previous answer but adapted it slightly with the context it was provided. Once again, all these high level tasks are too broad to be actionable yet. We will continue to provide more details to the first step and see how ChatGPT responds.

7 Prompt

Help me Collect data from a representative sample of people that includes their age, country, app usage, number of friends, and which brand they prefer (A or B).

8 Response

Collecting data from a representative sample of people can be done in several ways, depending on the resources and time available. Here are some possible methods:

Online surveys: Conduct an online survey using a platform like Google Forms or SurveyMonkey. You can create a questionnaire that includes questions about age, country, app usage, number of friends, and which brand they prefer (A or B). You can then distribute the survey to a representative sample of people using social media or email lists.

In-person interviews: Conduct in-person interviews with people in public places such as malls or parks. Ask them questions about their age, country, app usage, number of friends, and which brand they prefer (A or B).

Online panels: Join an online panel service such as Toluna or YouGov. These services have a pool of people who have agreed to participate in surveys or other research studies. You can create a questionnaire that includes questions about age, country, app usage, number of friends, and which brand they prefer (A or B) and distribute it to a representative sample of people.

Social media groups: Join relevant social media groups or communities and ask members to participate in a survey or provide data about their preferences. You can create a questionnaire that includes questions about age, country, app usage, number of friends, and which brand they prefer (A or B) and post it in the group or community.

When collecting data, it's important to ensure that the sample is representative of the population you are interested in. This can be achieved by using random sampling or stratified sampling techniques. Additionally, it's important to ensure that participants provide informed consent and that their privacy is protected.

9 Observations

Things are starting to get more concrete, but still not concrete enough. At this point we simply copy/paste the first step as prompt.

10 Prompt

Conduct an online survey using a platform like Google Forms or SurveyMonkey. You can create a questionnaire that includes questions about age, country, app usage, number of friends, and which brand they prefer (A or B). You can then distribute the survey to a representative sample of people using social media or email lists.

11 Response

Here's an example of how you can create an online survey using Google Forms:

Go to Google Forms and log in to your Google account.
Click on the "Blank" template to create a new survey.
Add a title to your survey and include a brief introduction that explains the purpose of the survey and how the data will be used.
Add questions about age, country, app usage, number of friends, and brand preference (A or B). You can use a variety of question types, such as multiple choice, dropdown, or text input fields.
Customize the appearance of your survey by choosing a theme or adding images.
Preview your survey to ensure that it looks the way you want it to.
Set up distribution options, such as a link to the survey or embedding it on a website. You can also send the survey via email or social media.
Test your survey to ensure that it is working properly.
Monitor responses as they come in and analyze the data once you have a sufficient sample size.
When distributing your survey, it's important to ensure that you reach a representative sample of people. You

can use social media, email lists, or targeted advertising to reach your desired audience. It's also important to provide participants with informed consent and ensure that their privacy is protected.

12 Observations

Now we're at a level where there are actionable steps. The problem however is that we need some form of intelligence to recognize that a step is at a low level enough to be actionable. The next problem is that we need the necessary processes to execute said action. Let's see what happens when we ask ChatGPT to help us with the first step (again).

13 Prompt

Go to Google Forms and log in to your Google account.

14 Response

I'm sorry, as an AI language model, I don't have the ability to log in to your Google account. However, you can follow these steps to log in to your Google account and access Google Forms:

Go to the Google homepage (www.google.com) on your web browser.
Click on the "Sign in" button located on the top right corner of the page.
Enter your Google account email address and password in the provided fields.
Click on the "Next" button to proceed.
If prompted, enter the verification code sent to your registered phone number or email address to verify your identity.
Once you have successfully signed in, click on the "Apps" icon (represented by 9 small squares) located on the top right corner of the page.
Click on the "Forms" icon to access Google Forms.
You can now create a new form or edit an existing one.

15 Observations

At this point it has recognized that as an AI language model, it does not "have the ability to log in to your Google account". We could technically use this as an indication that we've reached a point at which the model would now need to delegate execution to another system capable of executing said steps.

Interestingly enough, a simple approach is to simply ask ChatGPT to produce a program from the prior list of steps.

16 Prompt

Return the list of previous steps as if they were a program.

17 Response

Sure, here is an example program that outlines the steps to create an online survey using Google Forms:

```
# Import the necessary modules
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
import time
```

```
# Open Google homepage
browser = webdriver.Chrome()
browser.get("https://www.google.com")

# Click the "Sign in" button
sign_in_button = browser.find_element_by_link_text("Sign in")
sign_in_button.click()

# Enter email and password
email_field = browser.find_element_by_name("identifier")
email_field.send_keys("your_email_address")
email_field.send_keys(Keys.RETURN)
time.sleep(2)
password_field = browser.find_element_by_name("password")
password_field.send_keys("your_password")
password_field.send_keys(Keys.RETURN)

# Navigate to Google Forms
browser.find_element_by_class_name("gb_S").click()
time.sleep(1)
browser.find_element_by_xpath("//div[contains(text(), 'Forms')]").click()

# Create a new form
browser.find_element_by_xpath("//span[contains(text(), 'Blank')]").click()

# Add a title and introduction to the form
title_field = browser.find_element_by_xpath("//input[@aria-label='Untitled form']")
title_field.send_keys("Brand preference survey")
intro_field = browser.find_element_by_xpath("//textarea[@aria-label='Form description']")
intro_field.send_keys("Please answer the following questions about your brand preferences.")

# Add questions to the form
age_question = browser.find_element_by_xpath("//div[@aria-label='Untitled question']")
age_question.click()
age_question.send_keys("What is your age?")
country_question = browser.find_element_by_xpath("//div[@aria-label='Untitled question']")
country_question.click()
country_question.send_keys("What country do you live in?")
app_usage_question = browser.find_element_by_xpath("//div[@aria-label='Untitled question']")
app_usage_question.click()
app_usage_question.send_keys("How much time do you use the app per day?")
friends_question = browser.find_element_by_xpath("//div[@aria-label='Untitled question']")
friends_question.click()
friends_question.send_keys("How many friends do you have on this app?")
brand_preference_question = browser.find_element_by_xpath("//div[@aria-label='Untitled question']")
brand_preference_question.click()
brand_preference_question.send_keys("Do you prefer brand A or brand B?")

# Customize the appearance of the form
theme_button = browser.find_element_by_xpath("//div[contains(text(), 'Theme')]")
theme_button.click()
theme_dropdown = browser.find_element_by_xpath("//div[@aria-label='Choose a theme']")
theme_dropdown.click()
theme_options = browser.find_elements_by_xpath("//div[@aria-label='Theme options']")
```

```
theme_options[2].click()

# Preview and distribute the form
preview_button = browser.find_element_by_xpath("//div[contains(text(), 'Preview')]")
preview_button.click()
distribute_button = browser.find_element_by_xpath("//div[contains(text(), 'Send')]")
distribute_button.click()
time.sleep(1)
email_option = browser.find_element_by_xpath("//div[contains(text(), 'Email')]")
email_option.click()
recipient_field = browser.find_element_by_xpath("//input[@aria-label='To']")
recipient_field.send_keys("example@gmail.com")
subject_field = browser.find_element_by_xpath("//input[@aria-label='Add a subject']")
subject_field.send_keys("Brand preference survey")
send_button = browser.find_element_by_xpath("//div[contains(text(), 'Send')]")
send_button.click()

# Close the browser
browser.quit()
```

Note that this program is written in Python using the Selenium WebDriver library to automate web browser actions. It assumes that the user has the necessary credentials to access Google Forms and has the ChromeDriver executable installed on their system.

18 Observations

Here it produced some python code using the Selenium webdriver library. I have not tested whether the code actually works as intended, but I can tell from this that a necessary next step would be to have some kind of sandboxed environment in which the code can be executed. Another necessary thing would be to have python installed, with the necessary python libraries. It is not clear however how we should recognize this. Maybe we could force ChatGPT to always produce code in a given programming language. We would also know that we need to prompt it to return a list of libraries dependency, or use an external tool that would be responsible for retrieving the list of dependencies based on a code snippet, similar to how Go does dependency management.

Assuming the code works, then it will have successfully created a Google Form. In this case however, it is unlikely to be successful simply because the username/password isn't likely to result in a successful login. At this point we need to recognize success from failure.