a) Custom training job and prediction using managed datasets

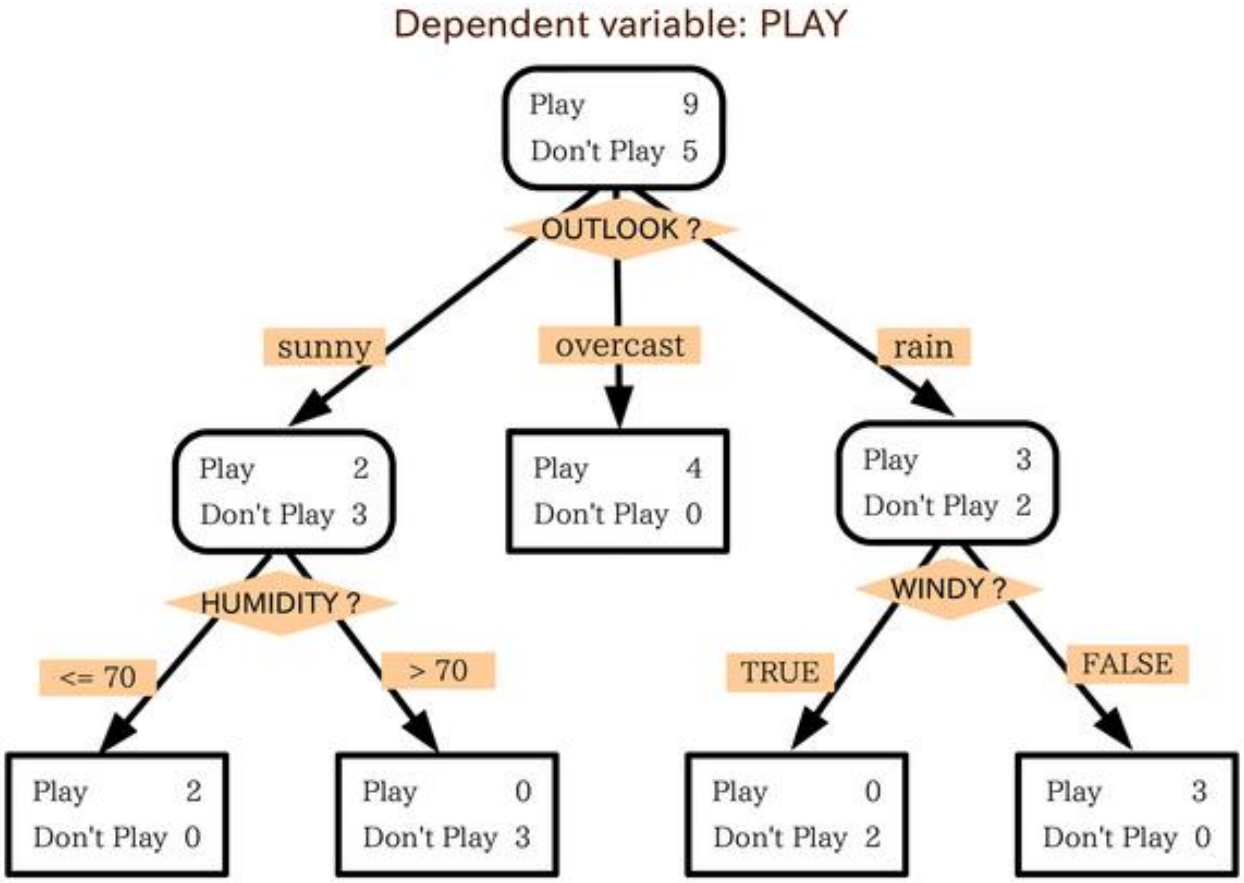
Reference: https://codelabs.developers.google.com/vertex-xgb-wit#0

Objective:

- Train an XGBoost model on a public mortgage dataset in a hosted notebook
- Analyze the model using the What-if Tool
- Deploy the XGBoost model to Vertex AI

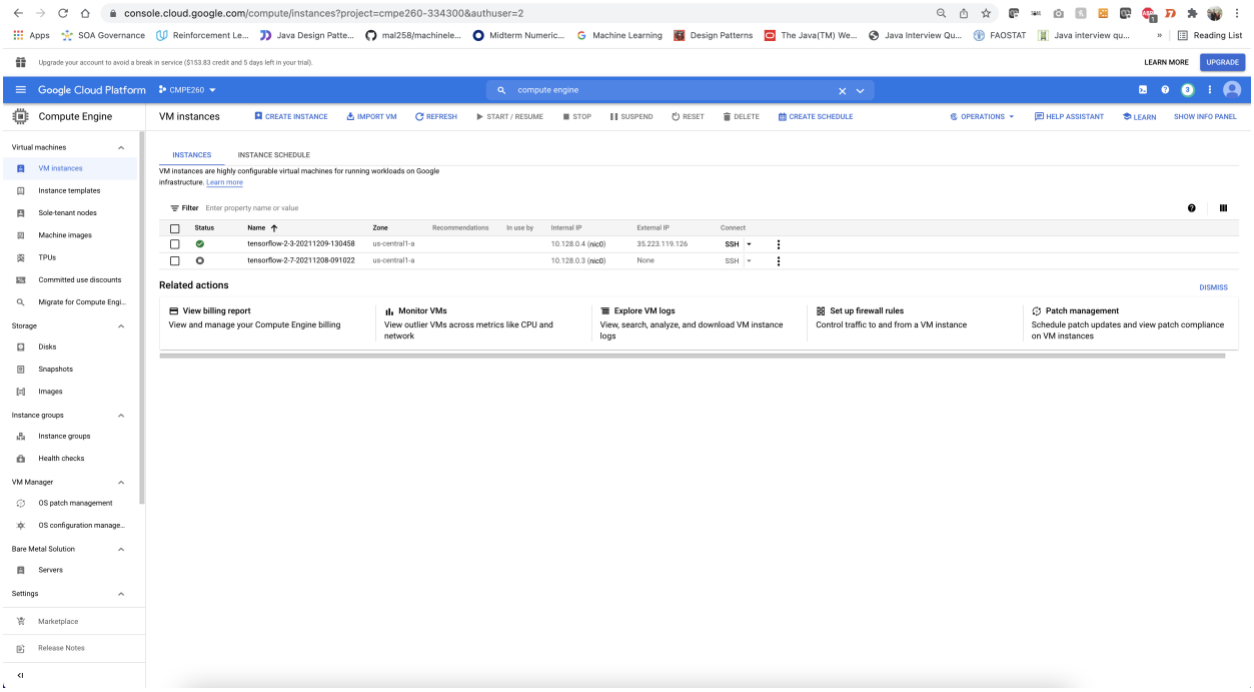
XGBoost is a machine learning framework that uses decision trees and gradient boosting to build predictive models. It works by ensembling multiple decision trees together based on the score associated with different leaf nodes in a tree.

The diagram below is a visualization of a simple decision tree model that evaluates whether a sports game should be played based on the weather forecast:

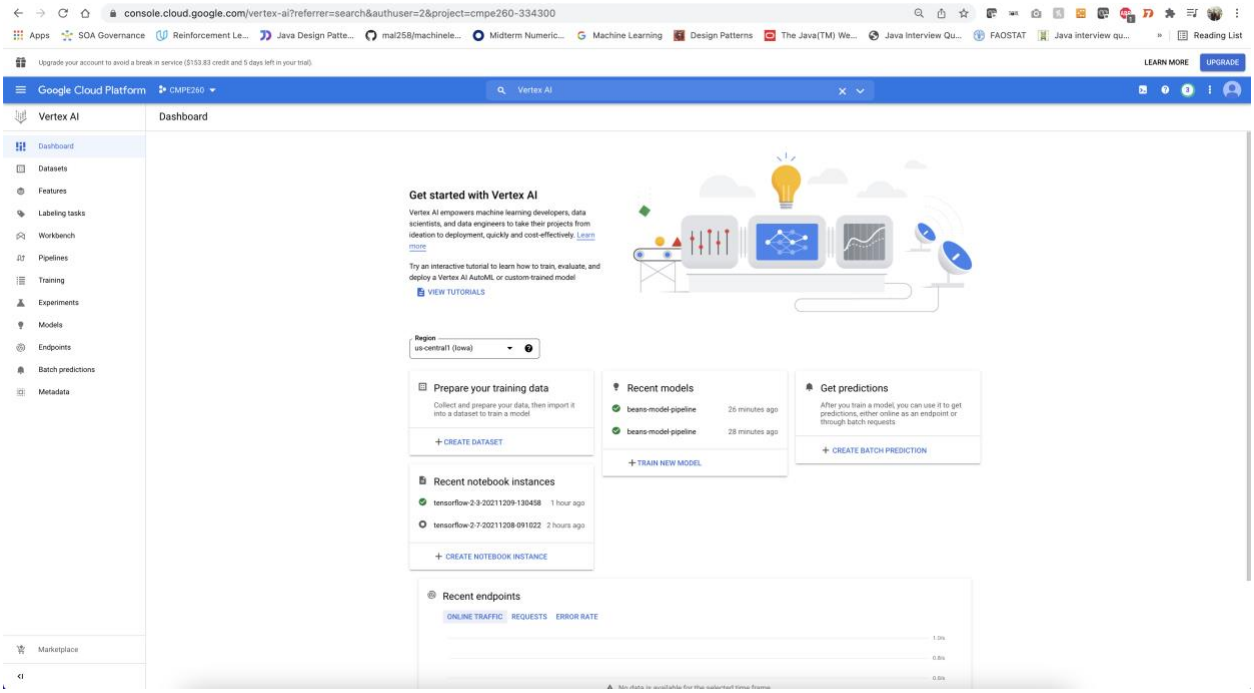


Why are we using XGBoost for this model? While traditional neural networks have been shown to perform best on unstructured data like images and text, decision trees often perform extremely well on structured data like the mortgage dataset we'll be using in this codelab.

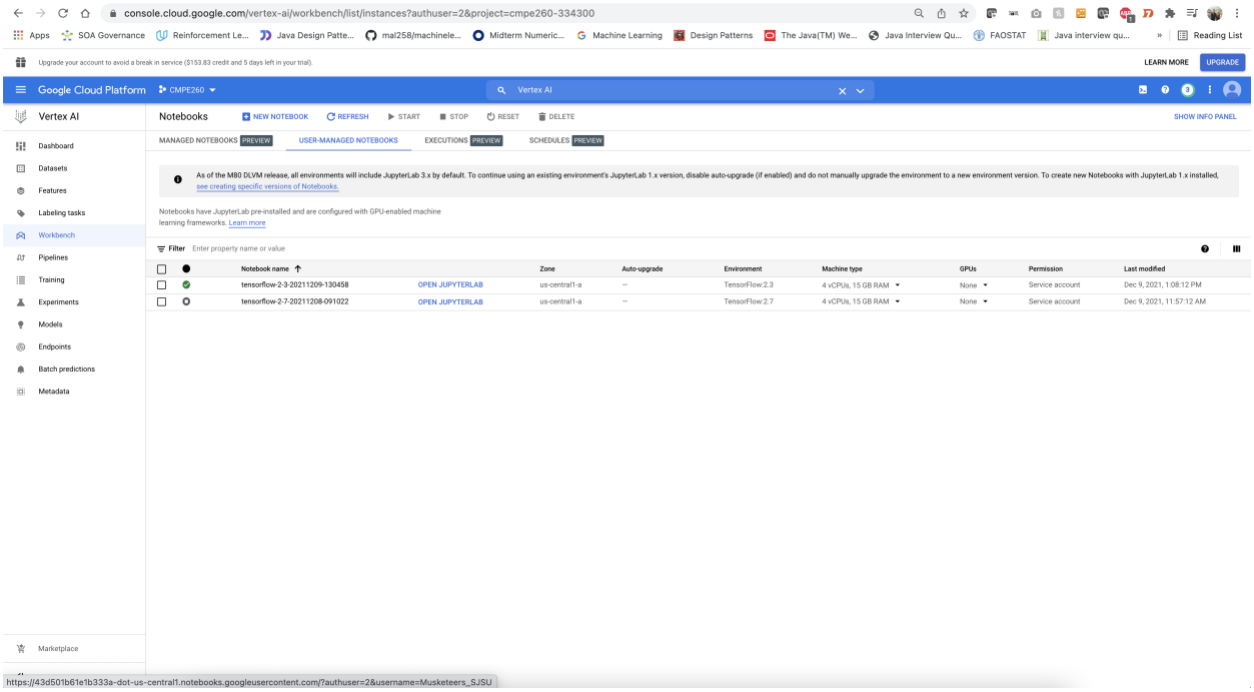
Step 1: Setup Cloud environment



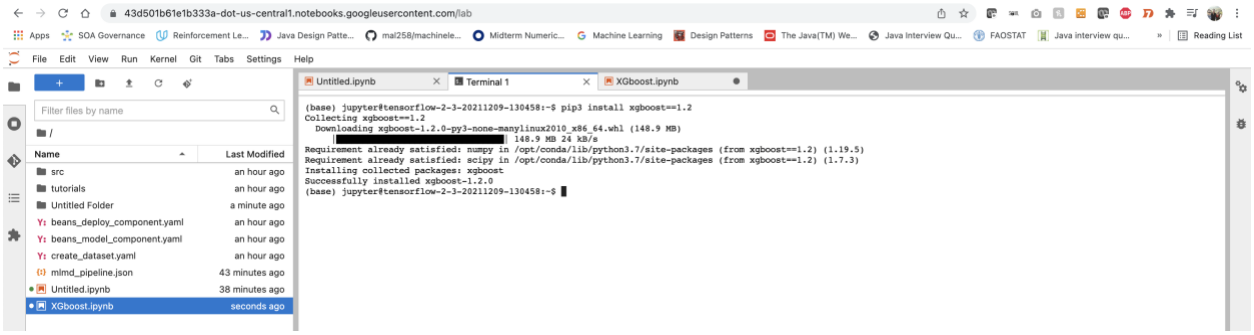
Step 2: Enable the Vertex AI API

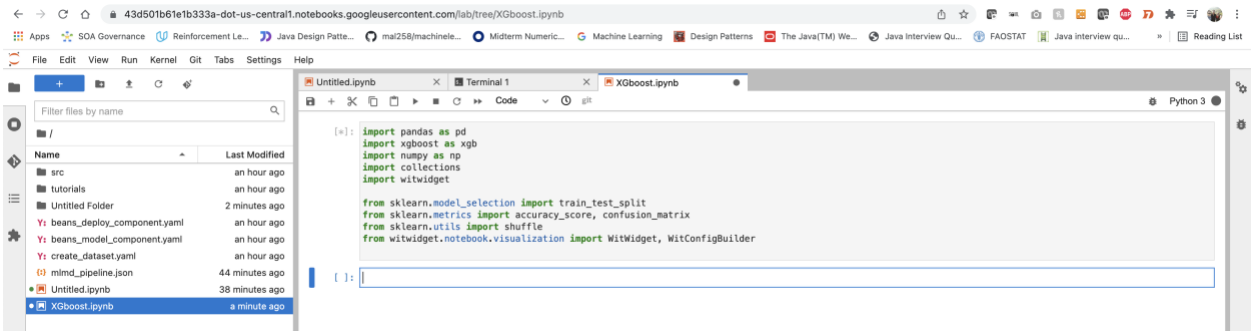


Step 3: Create a Notebooks instance



Step 4: Install XGBoost

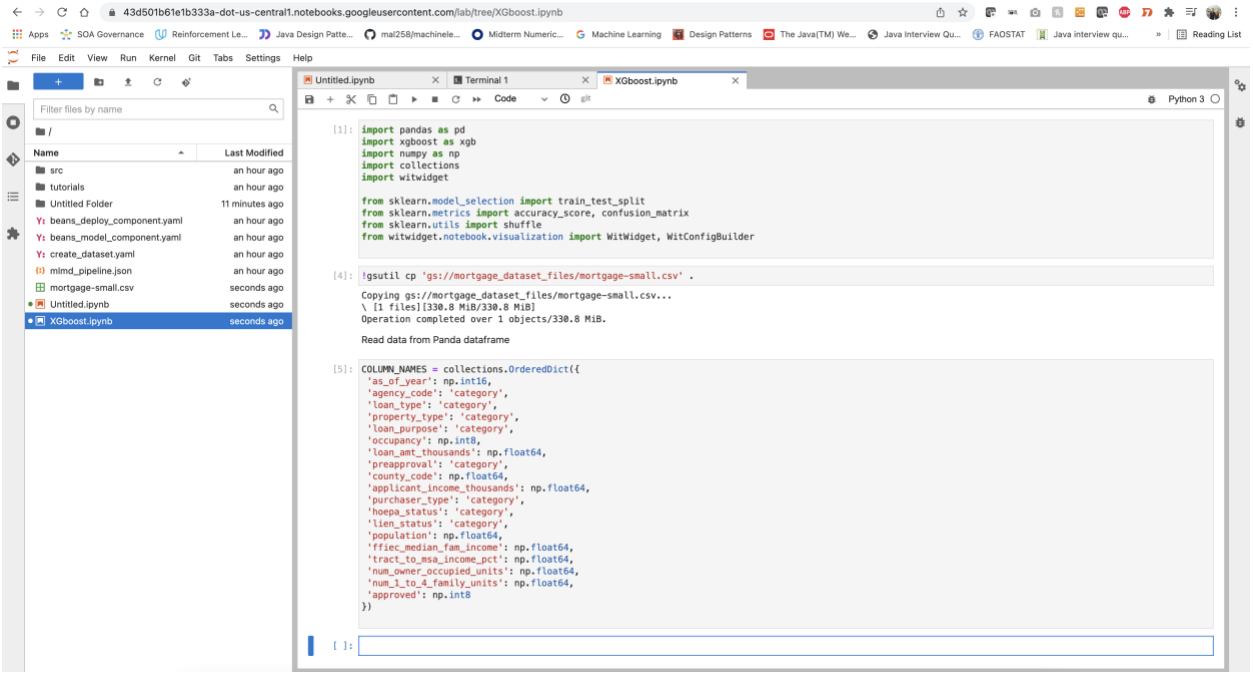


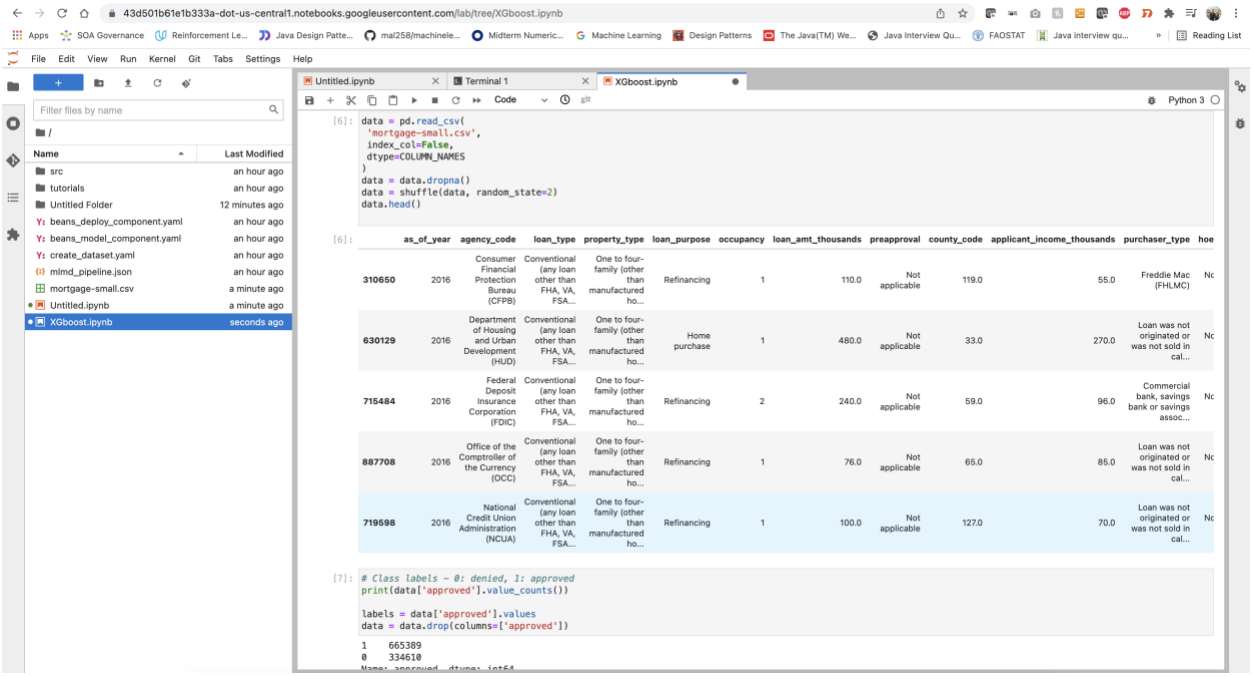


Step 5: Download and process data

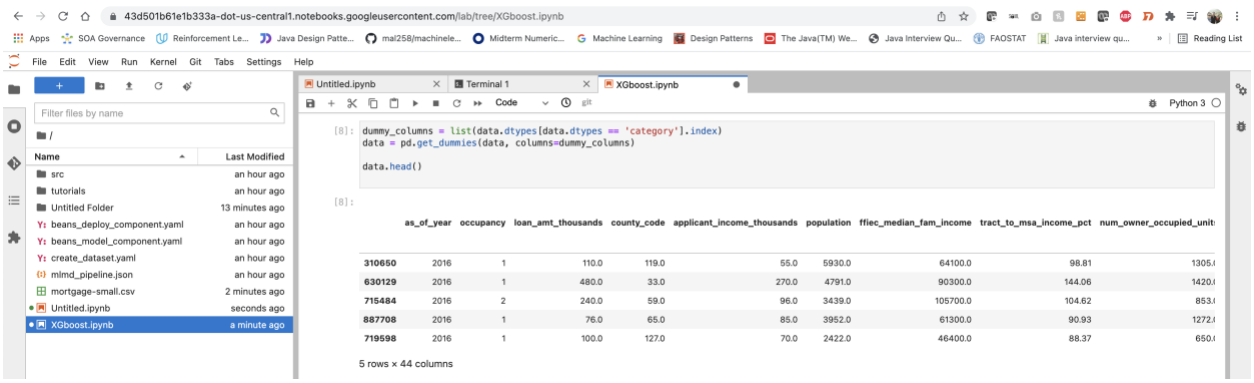
Download the pre-processed dataset

We'll use a mortgage dataset from ffiec.gov to train an XGBoost model. We've done some preprocessing on the original dataset and created a smaller version for you to use to train the model. The model will predict *whether or not a particular mortgage application will get approved*.





Creating dummy column for categorical values

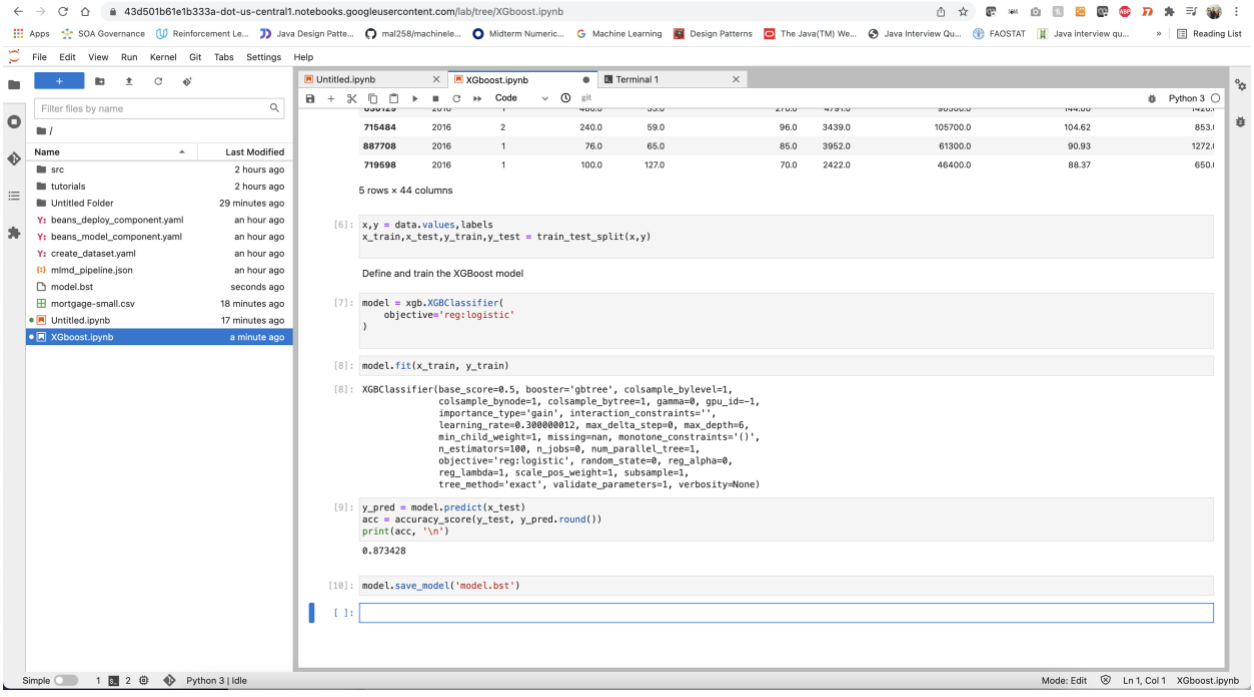


Splitting data into train and test sets

```
[9]: x,y = data.values,labels
     x_train,x_test,y_train,y_test = train_test_split(x,y)
```

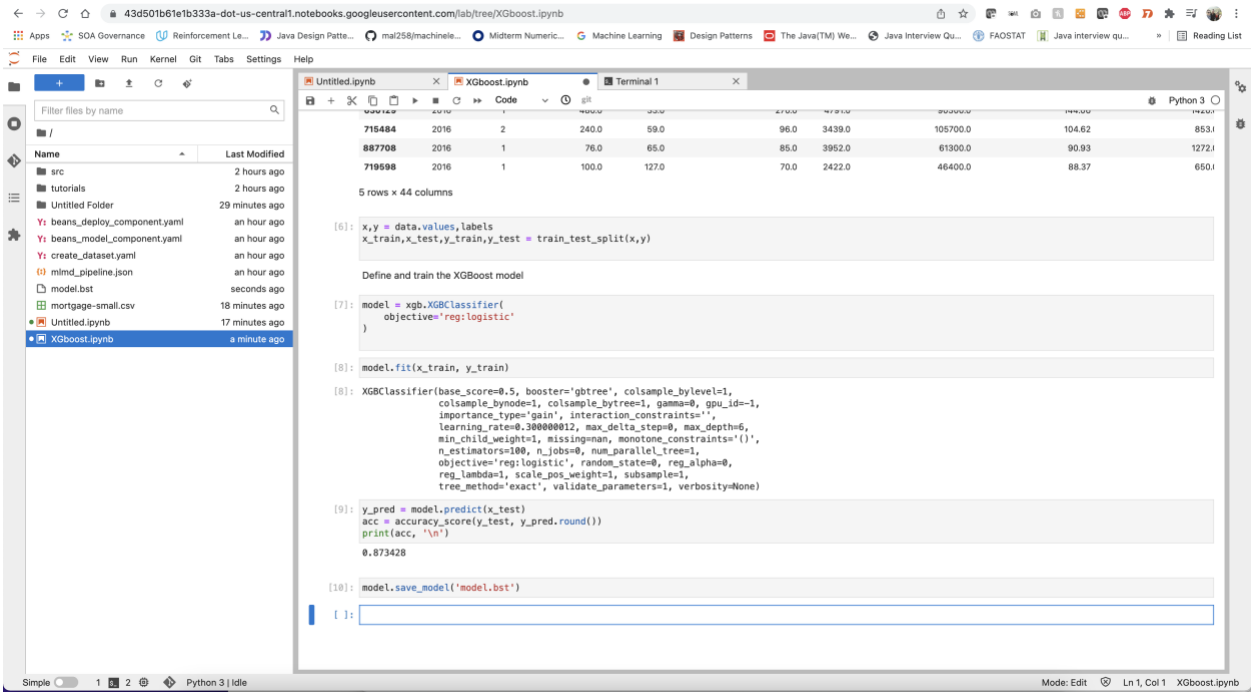
Define and train the XGBoost model

Creating a model in XGBoost is simple. We'll use the XGBClassifier class to create the model, and just need to pass the right objective parameter for our specific classification task. In this case we use reg:logistic since we've got a binary classification problem and we want the model to output a single value in the range of (0,1): 0 for not approved and 1 for approved



Evaluate the accuracy of your model and save the model

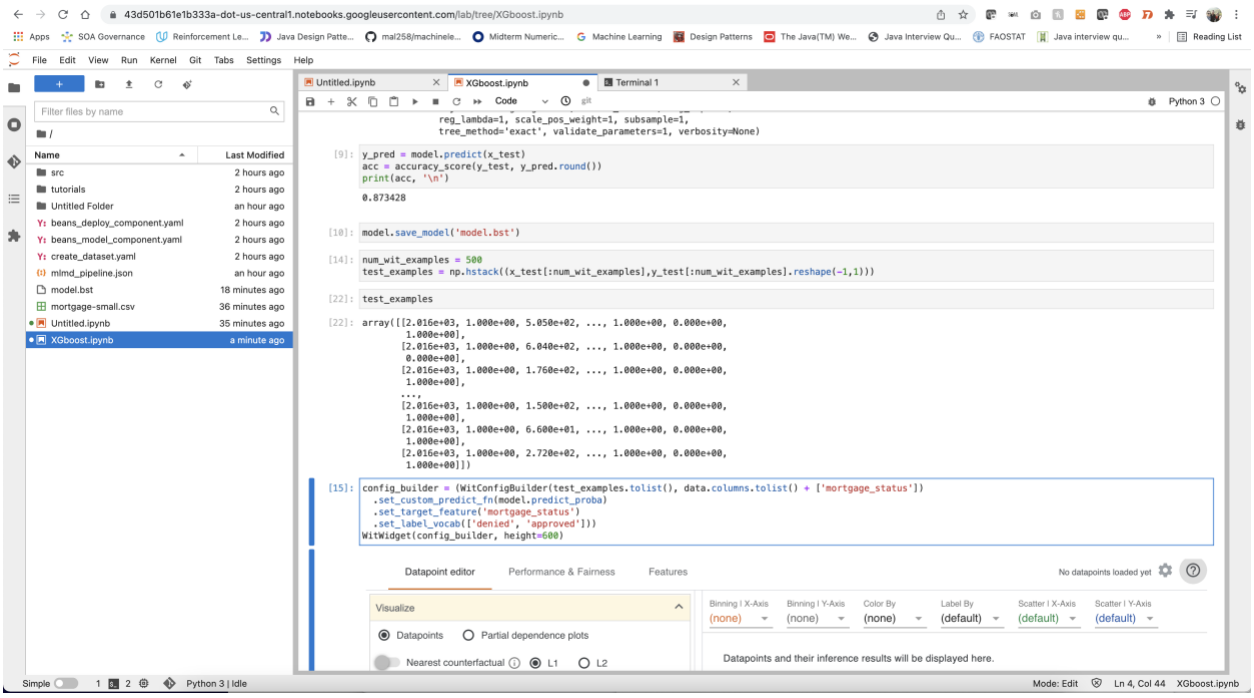
We can now use our trained model to generate predictions on our test data with the predict() function. Then we'll use Scikit-learn's accuracy_score() function to calculate the accuracy of our model based on how it performs on our test data. We'll pass it the ground truth values along with the model's predicted values for each example in our test set:

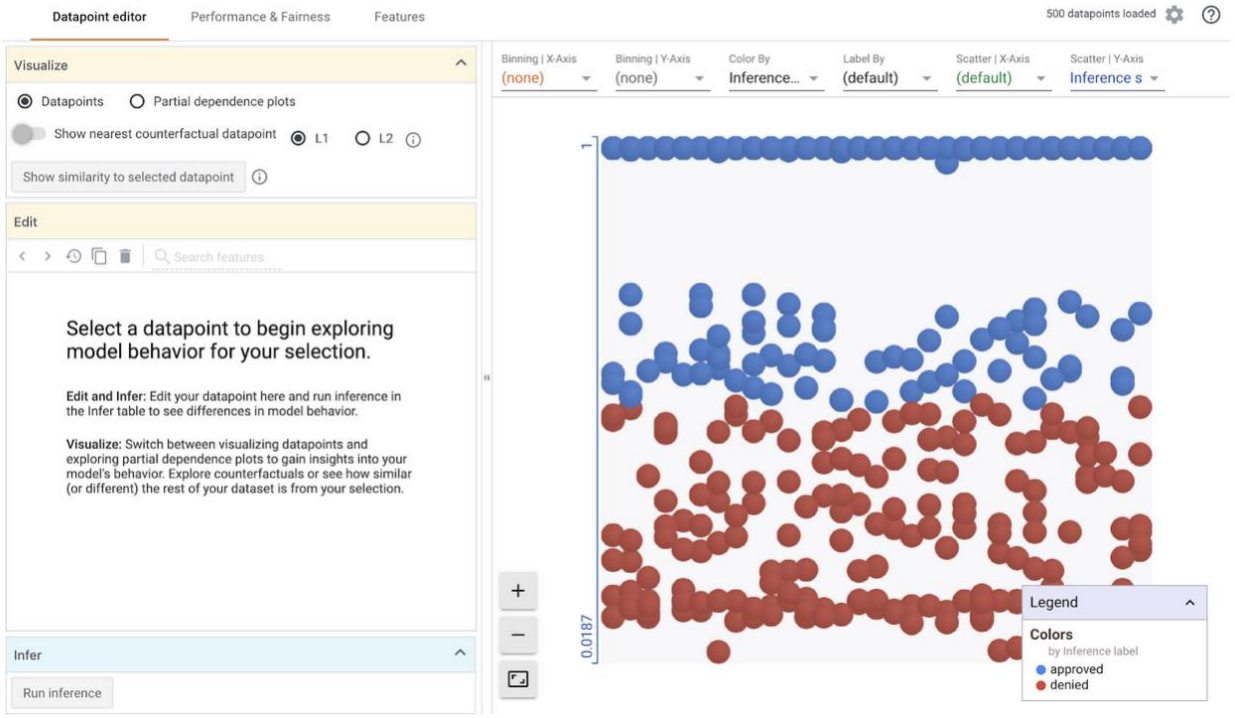


Step 6: Use the What-if Tool to interpret your model

Create the What-if Tool visualization

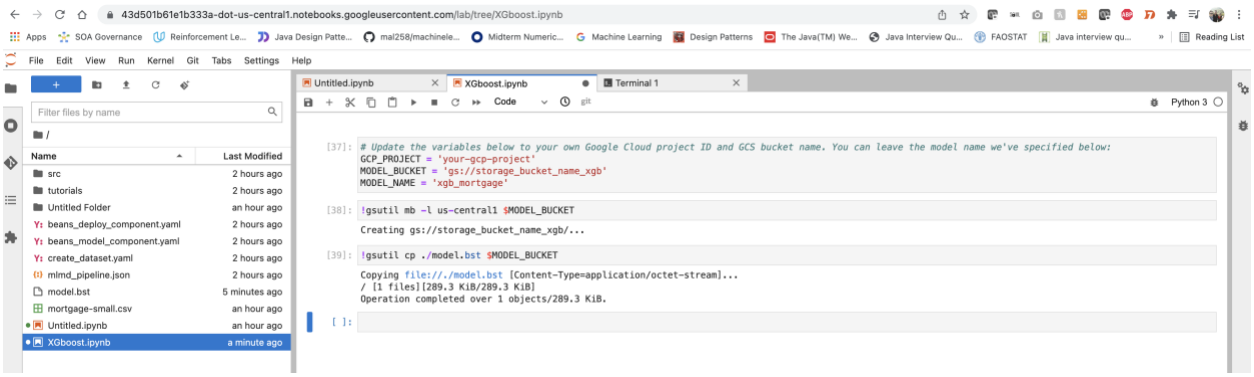
To connect the What-if Tool to your local model, you need to pass it a subset of your test examples along with the ground truth values for those examples. Let's create a Numpy array of 500 of our test examples along with their ground truth labels:



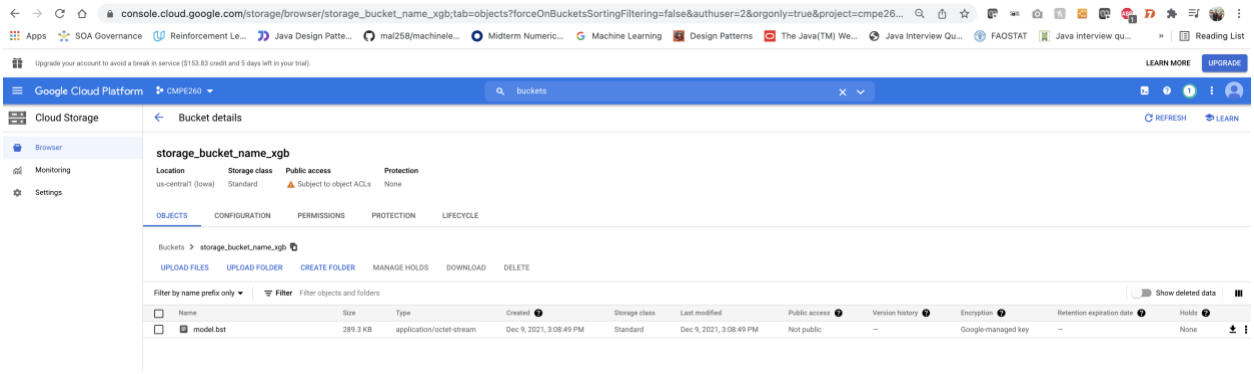


Step 7: Deploy model to Vertex AI

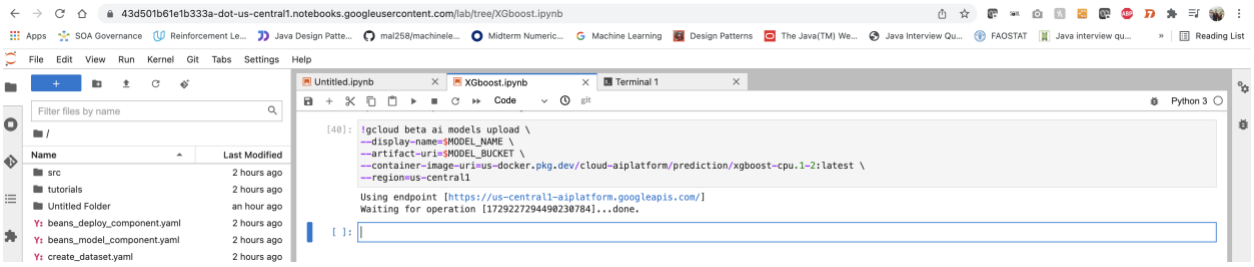
Create a Cloud Storage bucket for our model

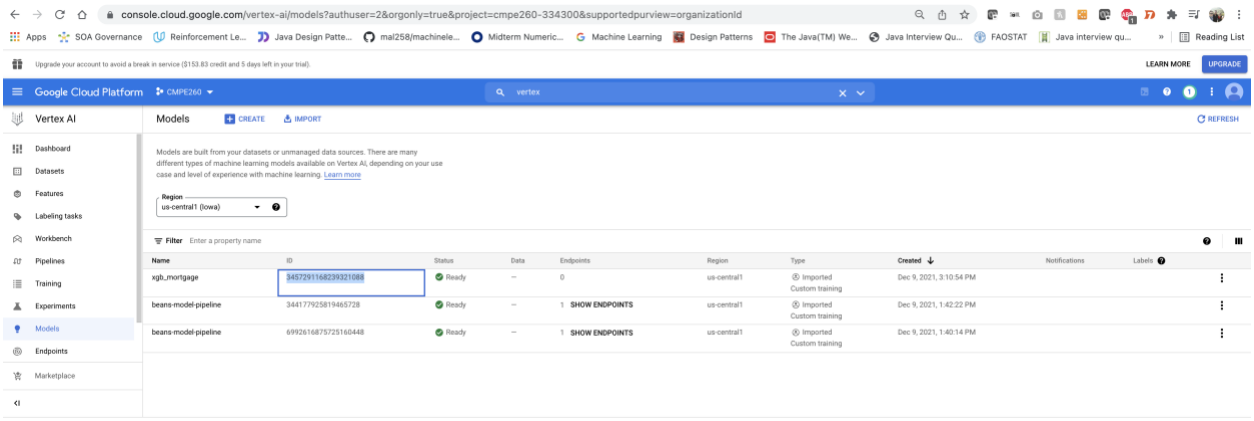


Copy the model file to Cloud Storage

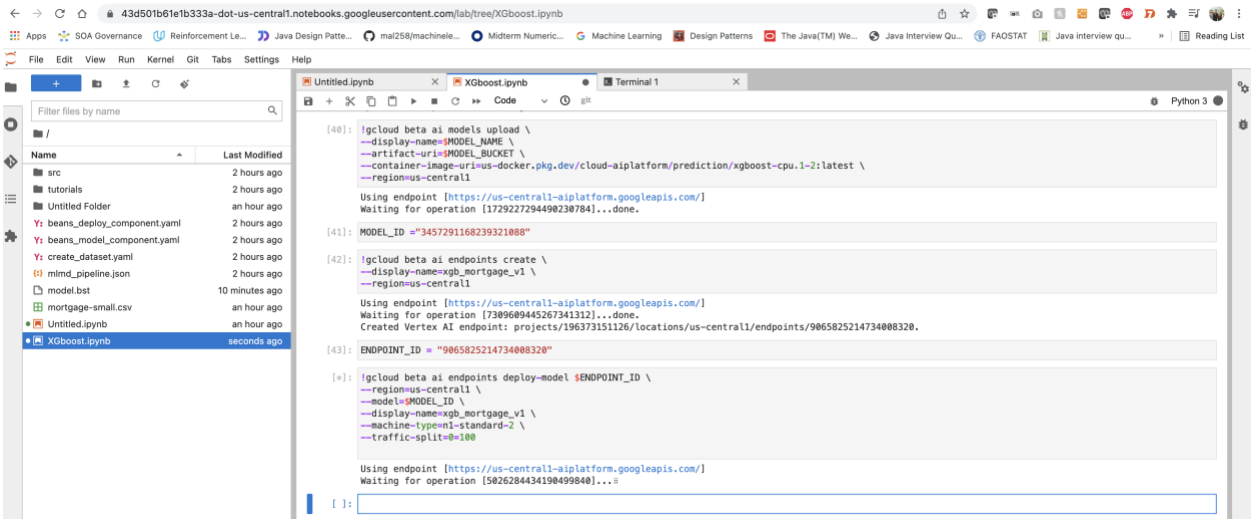


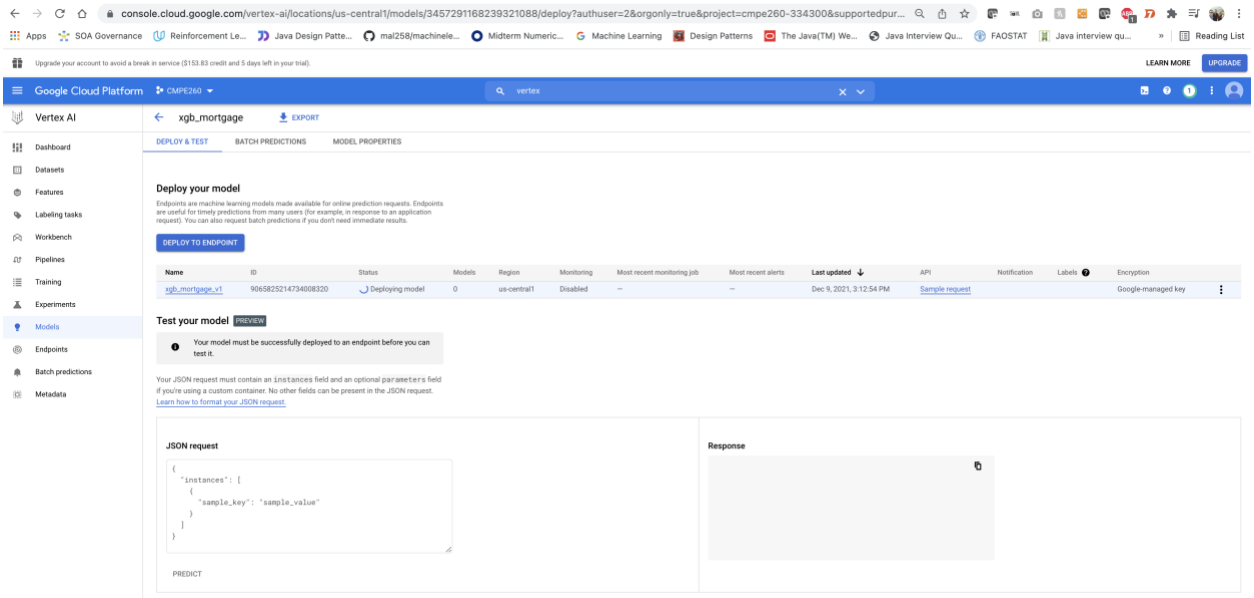
Create the model and deploy to an endpoint



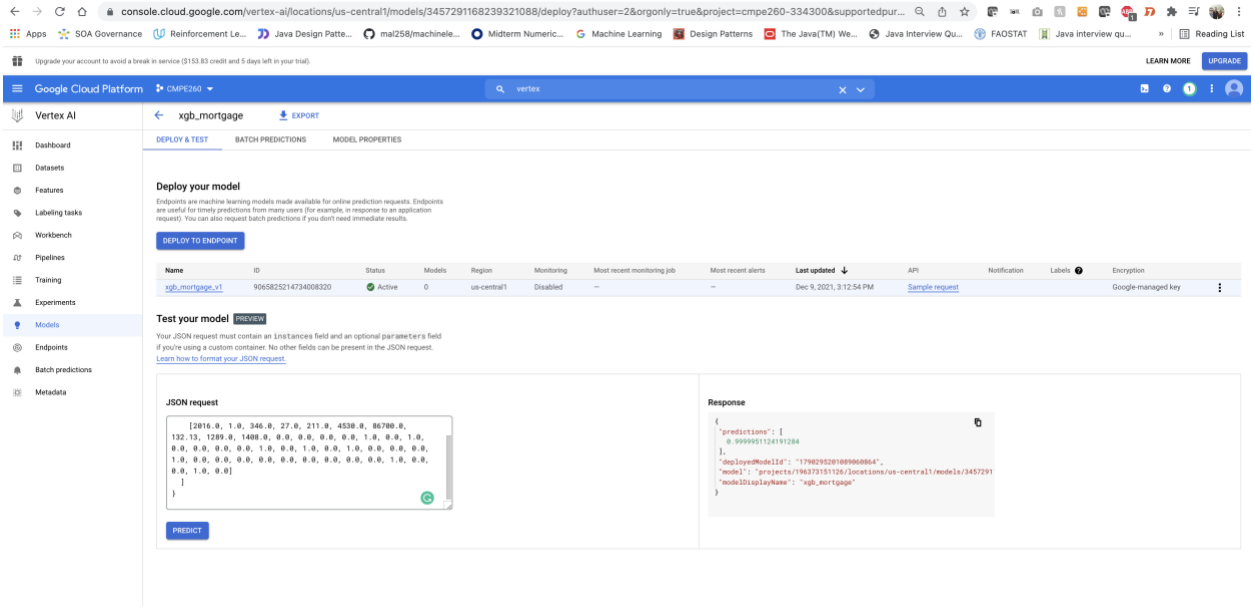


To deploy your endpoint, run the gcloud command below:

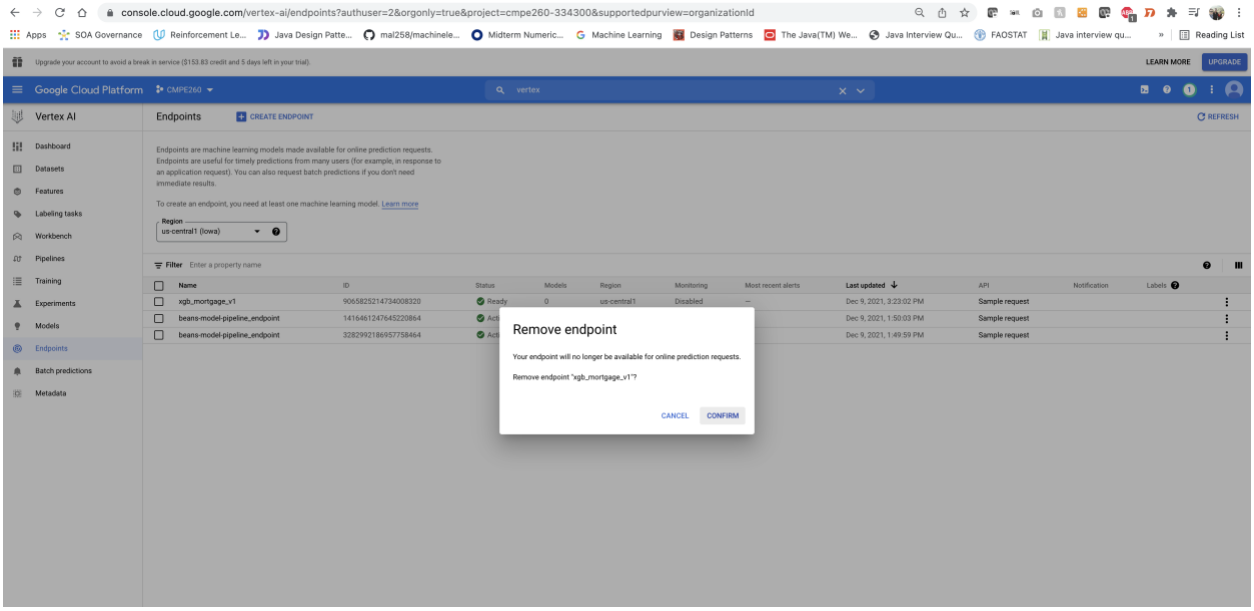


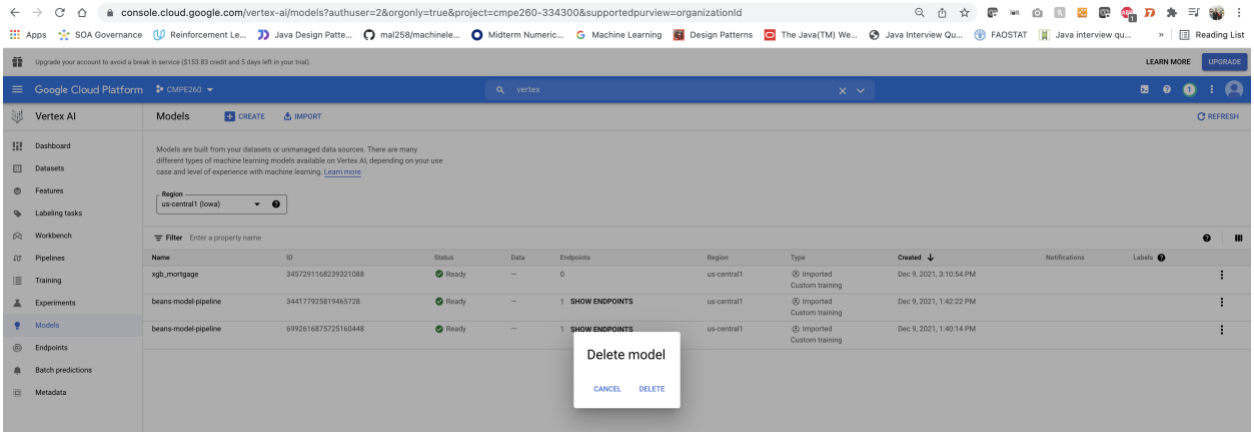


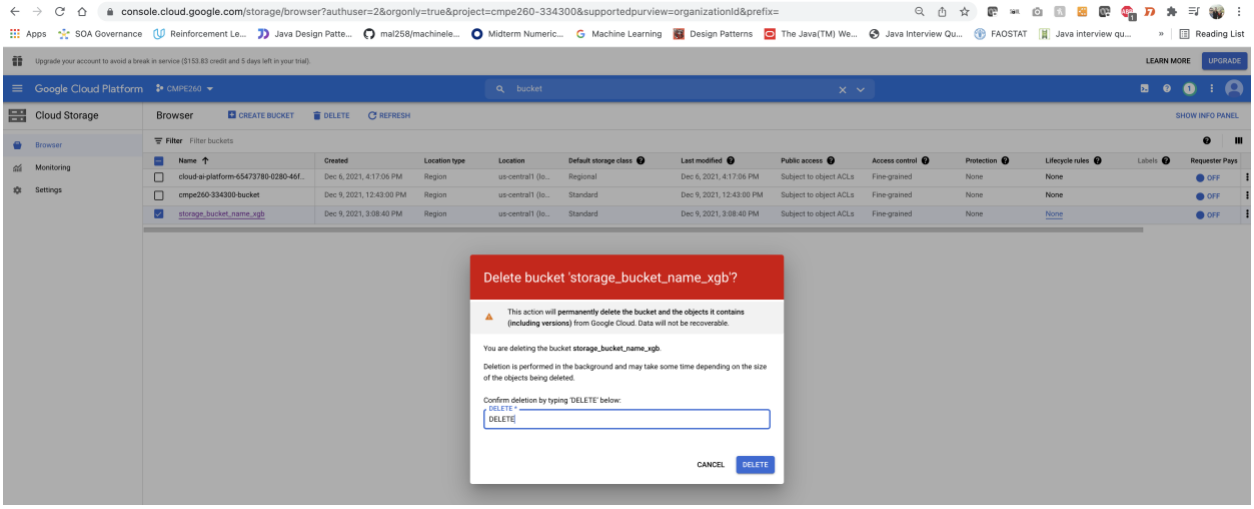
Test the deployed model



Clean Up remove endpoint , delete Model and delete bucket.




Delete model
CANCEL
DELETE


Delete bucket 'storage_bucket_name_xgb'?
This action will permanently delete the bucket and the objects it contains (including versions) from Google Cloud. Data will not be recoverable.
You are deleting the bucket storage_bucket_name_xgb.
Deletion is performed in the background and may take some time depending on the size of the objects being deleted.
Confirm deletion by typing 'DELETE' below:
DELETE *
DELETE
CANCEL
DELETE