



TAROTBOT

12.05.2020

CHRISTOPHER RICO

MSDS 434

PROJECT OVERVIEW

Our group was hired by Zoltar, Inc., to develop a proof of concept system that would detect and classify tarot cards from an image submitted via website. The solution was developed using Google AutoML Vision to train a custom object detection and multi-class classification model. Using AutoML to build and deploy the model allowed for rapid development and integration of computer vision capability within a basic Flask app.

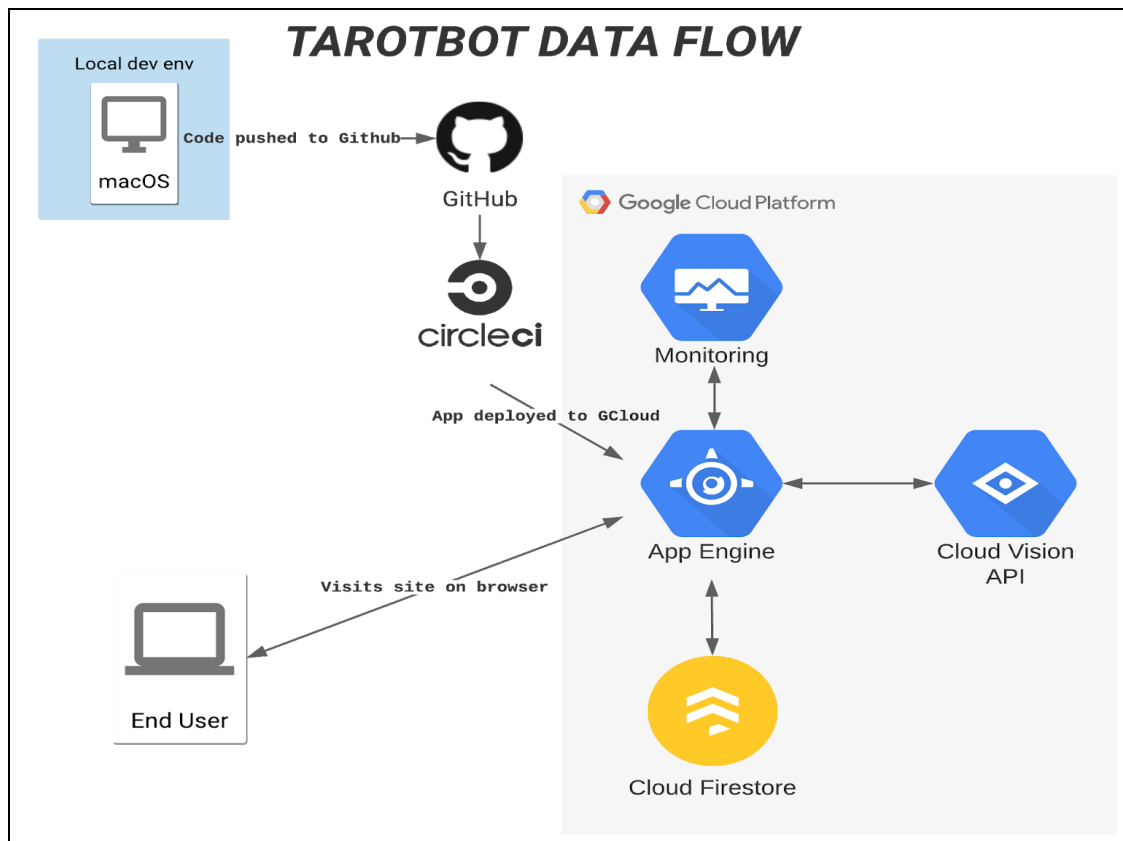
TOOLS USED

- **Pipenv:** Pipenv was used to create a virtual development environment within a Unix shell running on my local macOS machine.
- **Flask:** Flask operates as the central point of data flow within the app -- it serves out an interactive website where users can submit a photo and see the results of the machine learning inference.
- **Git/Github:** Development branch on Git stored commits to the dev build of the app. Once a working version was developed and tested via Pytest, the dev branch was merged with the main branch and pushed to GitHub.
- **CircleCI:** CircleCI was used to incorporate continuous integration into the project. Anytime code was pushed to the main branch of the GitHub repository, CircleCI was triggered to build and test the newest version of the app before deploying to Google App Engine.
- **Google App Engine:** GAE was used as the server to serve out the Flask app, which is a basic website that accepts user input of a photo, connects to the AutoML Vision API to get an inference, then stores said inference in Firestore. When users view the results page of the app, it retrieves past stored inferences from Firestore.
- **Google AutoML Vision:** This is the service that was used to train and evaluate the object detection and classification model, as well as deploy the completed model to the cloud so that the Flask app could connect to it via the Vision API client libraries.
- **Google Firestore:** Firestore is used as a NoSQL database to store tarot inference results. The Flask app stores individual results to Firestore as they are returned, and also requests all existing inference results to display to the user.
- **Google Cloud Monitoring:** Cloud monitoring provides logging of any errors that may occur. Cloud monitoring was also used to set uptime checks, and send alerts in case the app was unreachable for more than one minute.

DATA FLOW

Figure 1 below shows the entire dataflow pipeline, beginning with a user submitting a photo within the webpage form and ending with the results of the image inference displayed in text on a webpage after being stored in Firestore.

Further shown is the development/deployment pipeline, from a local development environment, to GitHub to testing on CircleCI to production deployment in Google App Engine.



RECOMMENDATIONS

The project team recommends the following project improvements, which were outside the scope of this project:

- Train a model to detect and classify all 78 classes of tarot card instead of just the 22 major arcana, as well as relative positions of the cards.
- Integrate Firebase into the project to allow for mobile users to sign in and see their results via the web browser.
- Integrate AJAX or JQuery into the webpage to display AutoML inference as soon as it is returned from the Vision API.
- Set up a QC environment to perform system testing within Google Cloud before build is pushed to GAE production environment.

APPENDIX A

Filename	Directory	Description
Pipfile(.lock)	./	These detail pipenv requirements
GCPkeys.json	./	Google Cloud Platform service credentials.
Config.yml	./circleci	CircleCI build, test, and deploy instructions.
App.yaml	./app	GAE app deployment configuration
GCPkeys.json	./app	Google Cloud Platform service credentials.
main.py	./app	This is the main program. Flask app that sets up routes to serve a website. Integrated with Vision API helper and Firebase helper.
main_test.py	./app	Unit tests to check that server, AutoML model, and firebase integrations are all online.
image_handler.py	./app	Code to integrate Flask app with AutoML Vision API. Sends image data to model, then translates model inference results into plain text.
firestore_handler.py	./app	Code to integrate Flask app with with Firestore. Set/Get methods store model inference results and retrieve all past inference results.
config.py	./app	Basic configuration for Flask app in dev environment.
index.html	./app/templates	Homepage for app, with form to submit photo
results.html	./app/templates	Results page which shows all previous AutoML inference results from Firestore..
noresults.html	./app/templates	Page that displays when no results are found in Firestore.
404.html	./app/templates	Page to display a pretty 404 error
500.html	./app/templates	Page to display a pretty 500 error.

APPENDIX B

Local dev environment

```
[06:23:43 (tb-wa-M-VRbLgj) christophrico@Hackintosh tb-wa ±|dev ×|→ ls
GCPkeys.json  Pipfile  Pipfile.lock  README.md  app  assets
[06:23:45 (tb-wa-M-VRbLgj) christophrico@Hackintosh tb-wa ±|dev ×|→ ls ./app
GCPkeys.json  config.py  image_handler.py  main_test.py  templates
app.yaml  firestore_handler.py  main.py  static
[06:23:50 (tb-wa-M-VRbLgj) christophrico@Hackintosh tb-wa ±|dev ×|→ python3 ./app/main.py
* Serving Flask app "main" (lazy loading)
* Environment: production
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 390-994-615
```

GAE production environment

Google Cloud Platform MSDS-fail-final-proj										
Search products and resources										
Versions REFRESH DELETE STOP START MIGRATE TRAFFIC SPLIT TRAFFIC LEARN SHOW INFO PANEL										
Filter versions										
Version	Status	Traffic Allocation	Instances	Runtime	Environment	Size	Deployed	Diagnose	Config	
20201124200727	Serving	100%	1	python37	Standard	158.2 KB	Nov 24, 2020, 12:08:10 PM by app-permission-for-gae@final-proj-292722.iam.gserviceaccount.com	Tools	View	
202011231072129	Serving	0%	0	python37	Standard	152.7 KB	Nov 22, 2020, 11:22:08 PM by app-permission-for-gae@final-proj-292722.iam.gserviceaccount.com	Tools	View	
202011231051935	Serving	0%	0	python37	Standard	151.6 KB	Nov 22, 2020, 9:20:21 PM by app-permission-for-gae@final-proj-292722.iam.gserviceaccount.com	Tools	View	
202011231042500	Serving	0%	0	python37	Standard	151.6 KB	Nov 22, 2020, 8:28:18 PM by app-permission-for-gae@final-proj-292722.iam.gserviceaccount.com	Tools	View	

CircleCI successful build

DashboardProjectBranchWorkflow

All Pipelines > Tarot-Bot-Webapp > main > build_test_deploy

build_test_deploy

Success

Insights NEW Rerun ...

Duration / Finished
2m 4s / 6h ago

Branch
main

Commit
126c61f

Author & Message
Added a few comments

build_test45s

gae_deploy1m 11s

Submission page

final-proj-292722.wn.r.appspot.com

NW Pages AI Reading Canvas GH CircleCI Other Bookmarks

TarotBot

Upload your image to look into the future!

*At present, this app only functions with still photos of an individual tarot card.
Card must be one of the [major arcana](#) from the [Rider-Waite-Smith deck](#).*

Drop files here or click to upload.

[Click here to see your predictions!](#)

Results page

https://final-proj-292722.wn.r. x +

final-proj-292722.wn.r.appspot.com/retrieve-predictions

NW Pages AI Reading Canvas GH CircleCI GCloud Playlists Other Bookmarks

TarotBot

Most Recent Prediction

Card	Confidence Score	Ethereal Prediction Time
The Emperor	0.9034	2020-11-25 02:29:42.629000+00:00

Previous Predictions

Card	Confidence Score	Ethereal Prediction Time
Your future is murky. Try another card.	666	2020-11-24 20:40:08.321000+00:00

[Click here to make another prediction!](#)

Firestore stored data

Google Cloud PlatformMSDS-fall-final-projSearch products and resources

DataDatabase location: us-west4

/ > tarot_bot > SY80XVCDx8xYT4iNnTK7

Root	tarot_bot	SY80XVCDx8xYT4iNnTK7
+ START COLLECTION	+ ADD DOCUMENT	+ START COLLECTION
⋮ tarot_bot >	JnasodDQ0zkb937kiTVD	+ ADD FIELD
	⋮ SY80XVCDx8xYT4iNnTK7 >	createdAt: November 24, 2020 at 6...
		display_name: "The Emperor"
		score: 0.9034

Cloud vision model training results

Google Cloud Platform

MSDS-fall-final-proj

Search products and resources

462_final_dataset

LABEL STATS

EXPORT DATA

Up to dataset list page

IMAGES

TRAIN

EVALUATE

TEST & USE

All images305

Labeled305

Unlabeled0

Filter images

Select all

Filter labels

chariot66

death69

devil62

emperor66

empress64

fool67

fortune65

hanged67

hermit68

hierophant65

judgment65

justice66

lovers67


magician66

moon67

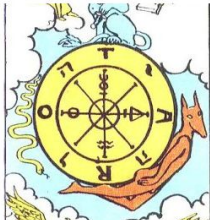
priestess67

star65


strength67




emperor(1)




fortune(1)




death(1), hanged(1), devil(1), sun(1), judgment(1)



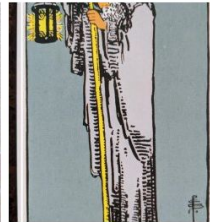
judgment(1), emperor(1), hanged(1), tower(1), ...




empress(1)



strength(1), temperance(1), death(1), hanged(1), ...



hermit(1)



tower(1), magician(1), hermit(1), judgment(1), h...

462_final_dataset

LABEL STATS

EXPORT DATA

IMPORT

IMAGES

TRAIN

EVALUATE

TEST & USE

Model

tarot_MA_462

Confidence threshold

0.5

IoU threshold

0.5

Filter labels

All labels

chariot

death

devil

emperor

empress

fool

fortune

hanged

hermit

hierophant

judgment

justice

lovers

magician

moon

priestess

star

strength

All labels

Total images273

Test items32

Total objects146

Object to image avg4.56

Precision95.74%

Recall92.47%

Use the slider to see which confidence threshold works best for your model on the precision-recall tradeoff curve.

Learn more about these metrics and graphs.

Precision-Recall curve

Recall

Precision

Precision-Recall curve

Confidence

Recall

Precision

Monitoring dashboard

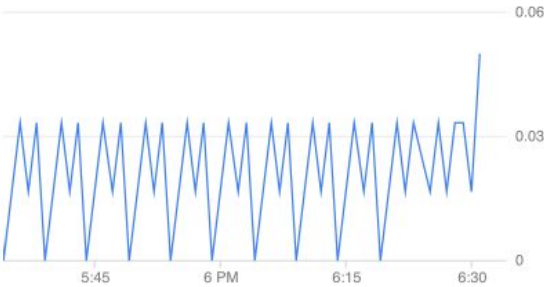
Charts

ADD CHART ^

GAE Application - Response count



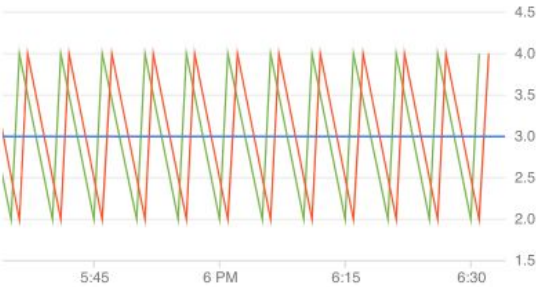
1 min interval (rate)



Uptime check passed



1 min interval (count true)



Firestore reads



1 min interval (rate)



Firestore writes



sum 1 min interval (rate)



