COVID -19 Predictor using End-to-End AIOps

Final Report Submitted
in Fulfillment of the Requirements
for the Course of
Major Project - II
In
Fourth Year – VIII Semester of
Bachelor of Technology
specialization
In
DevOps, CCVT and GG

Under

Dr. Hitesh Kumar Sharma

By

500060720	**R171217041**	**Nishkarsh Raj**
500060911	**R110217071**	**Karan Nautiyal**
500059999	**R110217083**	**Megha Rawat**
500060227	**R142217022**	**Rishabh Negi**

DEPARTMENT OF CYBERNETICS AND VIRTUALIZATION
SCHOOL OF COMPUTER SCIENCE
UNIVERSITY OF PETROLEUM AND ENERGY STUDIES,
BIDHOLI, DEHRADUN, UTTRAKHAND, INDIA

May 2021

Abstract

COVID-19 situation was declared as a pandemic by WHO as it was spreaded across the world. To understand the impact of COVID-19 it is imperative we stay up to date with the numbers of this pandemic. With India on the second rise of covid cases the burden on frontline workers has been greater than ever.
This project intends to offload the testing load from the medical workers in big cities where around 44,000 Covid tests are being conducted daily.
This project uses a model trained using the database made public by the Israeli Ministry of Health containing data of all individuals who were tested for SARS-CoV-2 via RT-PCR assay of a nasopharyngeal swab.

Keywords: *covid-19, RT-PCR, Covid Testing*

Introduction

The Coronavirus flare-up became exposed on 31 December 2019, when China reported the World Health Organization regarding numerous cases of pneumonia in Wuhan City. The illness spread to more places in China, and the world. The World Health Organization has now proclaimed it a pandemic. The infection has been named SARS-CoV-2 and the sickness is currently called COVID-19[1].

COVID-19 is a sickness brought about by a newfound CoronaVirus. The majority of people infected with the COVID-19 infection will encounter respiratory disease and recover without requiring unique treatment. This disease mostly affects senior citizens or the person with pre-existing medical conditions like cancer, diabetes, cardiovascular problem, etc.

There have been millions of Coronavirus cases to date and more than a million deaths across the world. The pandemic is in its midst and medical professionals across the globe are working on creating a coronavirus vaccine, there are hundreds of teams working to find the covid vaccine as the vaccine could save millions of lives globally. However, the distribution of the vaccine is still a very big hurdle and might take a lot of time[2]. The vaccine is only a way to control the spread and would not be able to eradicate it as it is a form of seasonal flu that can easily mutate. Covid-19 infection would last for the coming years as informed by WHO.

The project aim is to create an AIOPs pipeline for deploying a containerized application that helps detect covid from medical scans of an individual, since the virus is constantly mutating we might need to constantly modify and update the model and deploy the new model this agile model will help us achieve this with minimum time delays. This will ensure the project can be utilized to the maximum capacity.

Motivation

With India reporting around 3 lakh cases (as of May, 2021) there is immense burden on the medical system of the nation. The time spent testing can be better utilized for attending to the patients. Testing also puts individuals who suspect they might have covid but are healthy. Therefore, it is better to have an automated screening process before the testing to ensure only highly susceptible individuals arrive at physical testing locations.

Problem Statement

As COVID-19 pandemic is at a dangerous peak we need to ensure we can test and quarantine as efficiently and safely as possible. Crowding and test centers may create a risk in individuals that suspect they are infected but maybe healthy. Can we have a method to pre-screen individuals so only individuals who are highly likely to be infected have to be physically present at the test centre and healthy individuals can avoid exposure risks?

Objective

The aim is to engineer an AIOPs pipeline to deploy COVID-19 predictor application that can work as a pre-screener before needing to visit the physical location for the RT-PCR test. The intent is to get a more accurate estimate of people who might be suffering from the disease to get a better understanding of the extent of the disease and serve as a guideline for precautionary measures that should be taken to reduce further transmission of the virus, so that we can be sure when a strict lockdown might be necessary.

Literature Review

This project uses the database made public by the Israeli Ministry of Health containing data of all individuals who were tested for SARS-CoV-2 via RT-PCR assay of a nasopharyngeal swab

Medical Explanations of coronavirus and facts are validated from THE NEW ENGLAND JOURNAL OF MEDICINE which states the strategies and evidences as Coronaviruses are RNA viruses that are divided into four general; alpha coronaviruses and beta coronaviruses are known to infect humans.

SARS-CoV-2 is related to bat coronaviruses and to SARS-CoV, the virus that causes SARS. Similar to SARS-CoV, SARS-CoV-2 enters human cells through the angiotensin-converting–enzyme 2 (ACE2) receptor. SARS-CoV-2 has

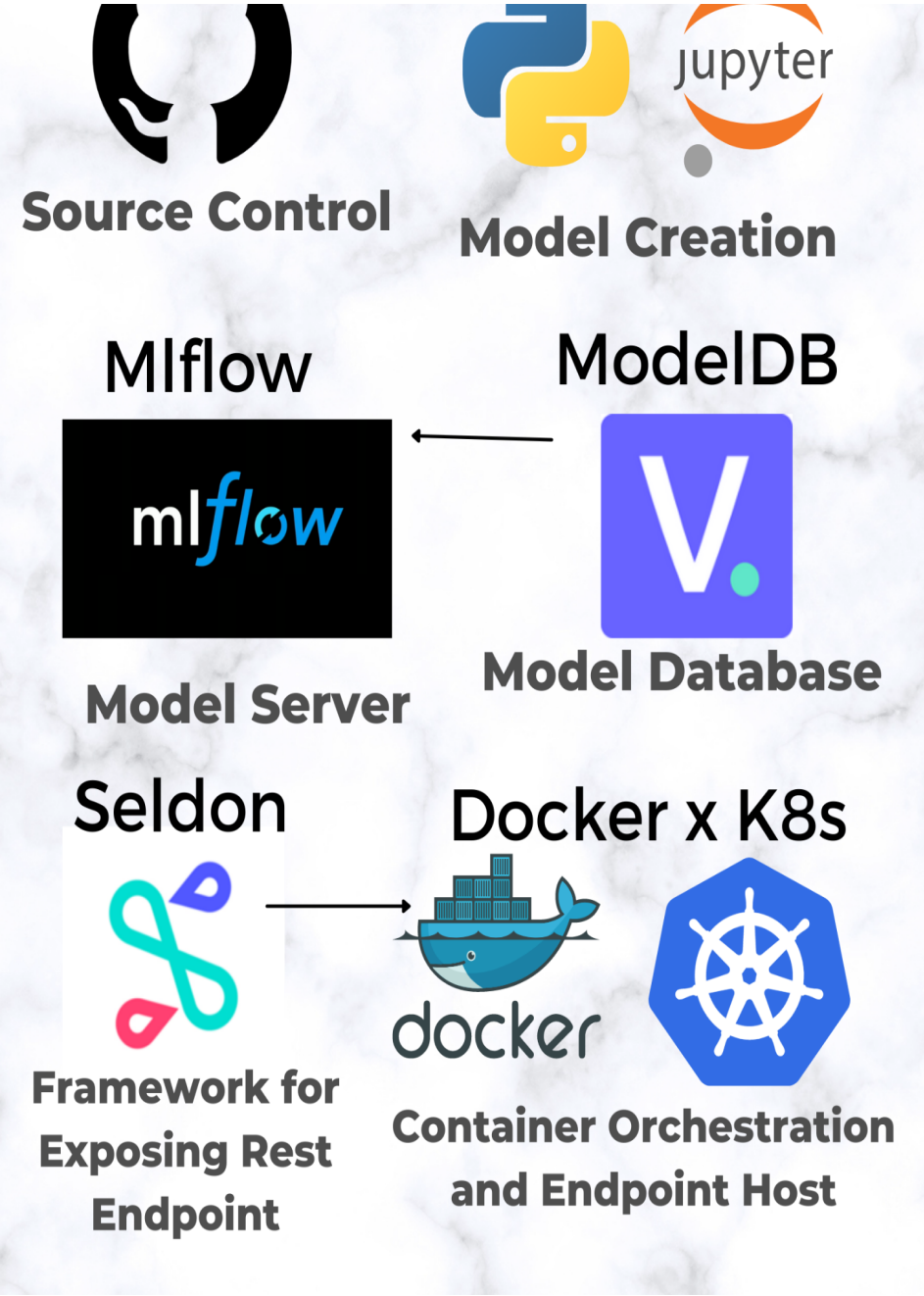
RNA-dependent RNA polymerase and proteases, which are targets of drugs under investigation.[3]

The model takes into consideration how the virus differently affects males and females.[17]

Proposed Method

1. This project uses public data set populated by Israel's health ministry for training
2. A pytorch based machine learning pipeline is created to build a two-classification model (Covid and Non-Covid)
3. The machine learning model is then serialized with cloud pickle.
4. A predictor function is wrapped around the model to take images of CT scan as an input and return which class the image belongs to.
5. For hosting the model's predictor function publicly, we expose REST and gRPC endpoints by migrating the predictor function to the Seldon framework.
6. Containerized application wrapper is created with Docker and hosted publicly on managed Kubernetes.
7. Model storage is handled with an instance of the Verta AI's ModelDB platform.
8. For logging and registration of the model for endpoints in gRPC, Mlflow functionality support is added.

Methodology



System Requirements:

1. **Software:**

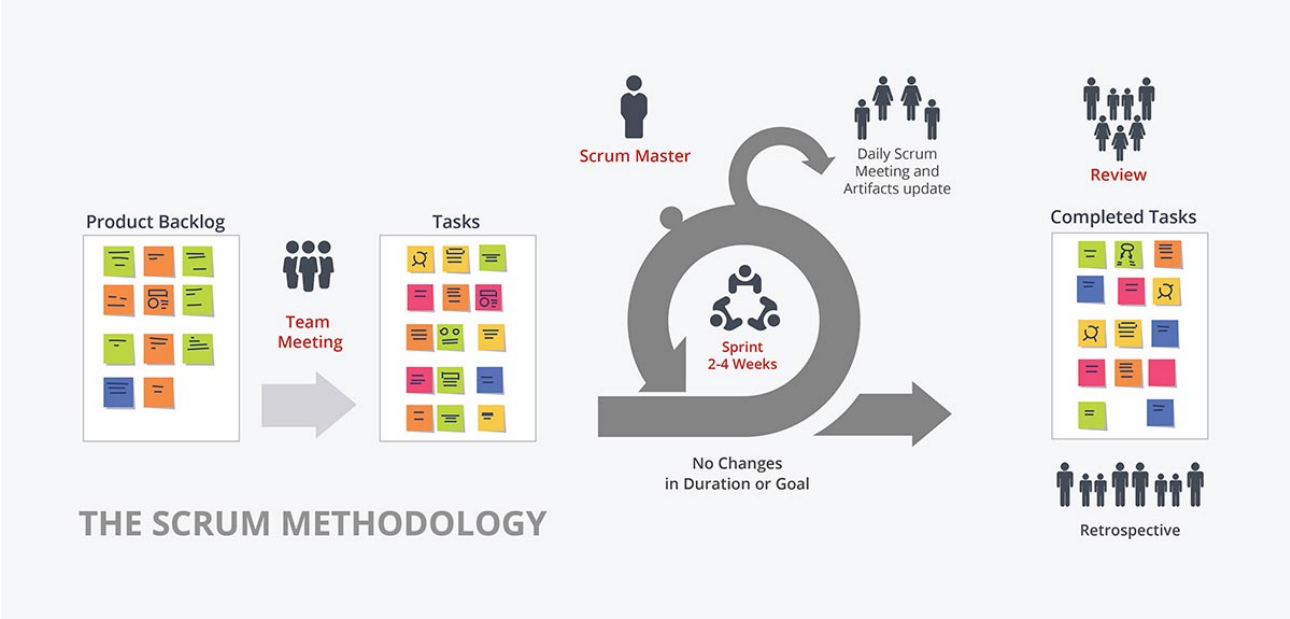
- Python 3.8.5 or above
- Jupyter Notebook
- VSCode/ IntelliJ/ Eclipse IDE
- Seldon, Mlflow and ModelDB dependencies of Python
- Docker Engine
- Azure CLI
- Git Bash

2. **Hardware:**

- 4 GB RAM or above
- Intel Pentium 4 processor or later that's SSE2 capable

SDLC Model: SCRUM Model

The entire project is divided into individual modules, based on their functionalities. Which will be developed in isolation by the entire team, based on the priority of implementation. Using the Version Control tool Git.

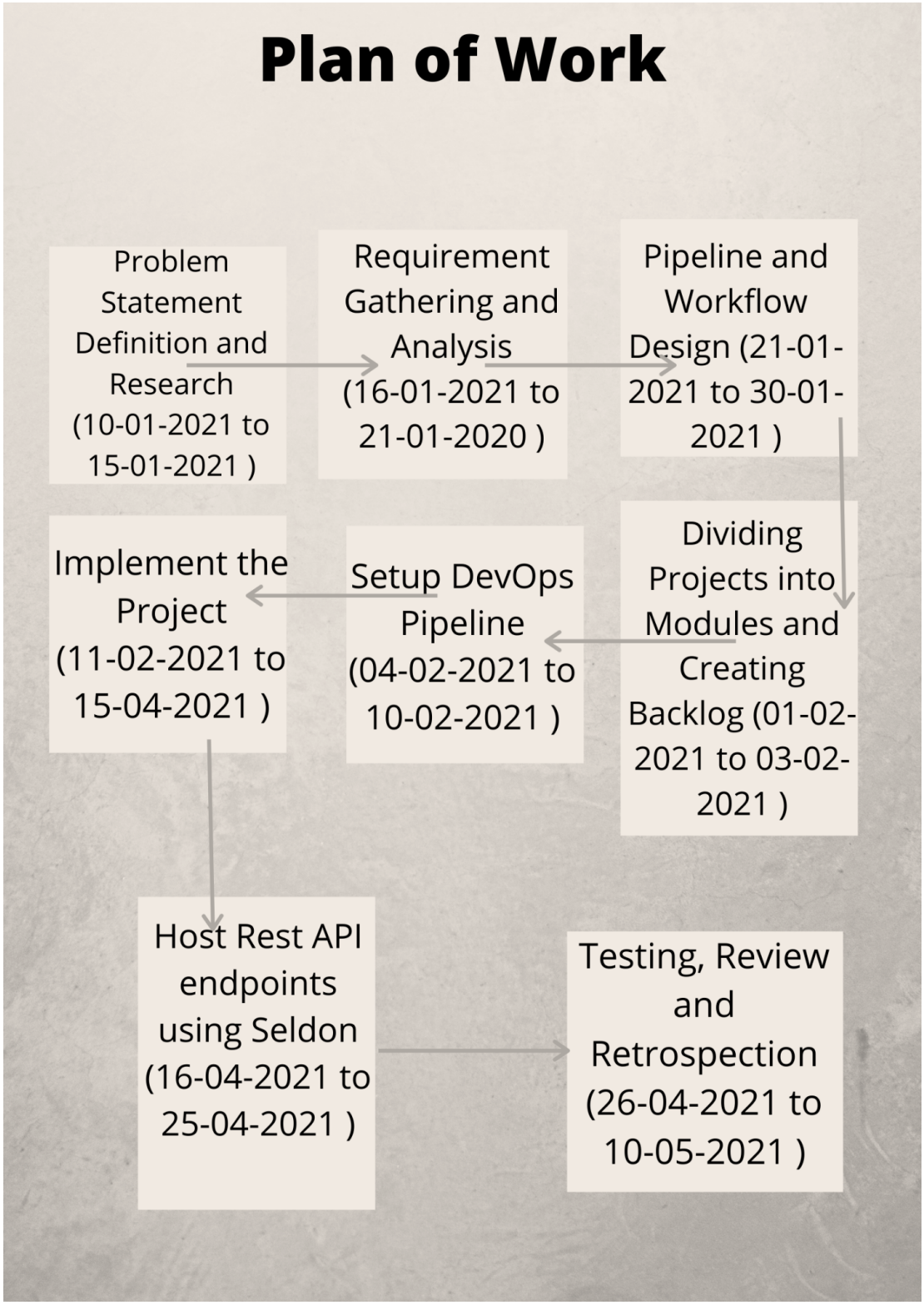


Scrum Model

Efficiency Parameters

1. **Sensitivity:** The test should be sensitive enough to not miss many positive cases as that defeats the purpose.
2. **Specificity:**The model requires moderate specificity as someone with a false positive result can always go to a physical screening location.
3. **Frequency:** The frequency of updates should be as much as possible to ensure the model is relevant.

Plan of Work


Plan of Work
Problem Statement Definition and Research (10-01-2021 to 15-01-2021)
Requirement Gathering and Analysis (16-01-2021 to 21-01-2020)
Pipeline and Workflow Design (21-01-2021 to 30-01-2021)
Dividing Projects into Modules and Creating Backlog (01-02-2021 to 03-02-2021)
Setup DevOps Pipeline (04-02-2021 to 10-02-2021)
Implement the Project (11-02-2021 to 15-04-2021)
Host Rest API endpoints using Seldon (16-04-2021 to 25-04-2021)
Testing, Review and Retrospection (26-04-2021 to 10-05-2021)

Project progress

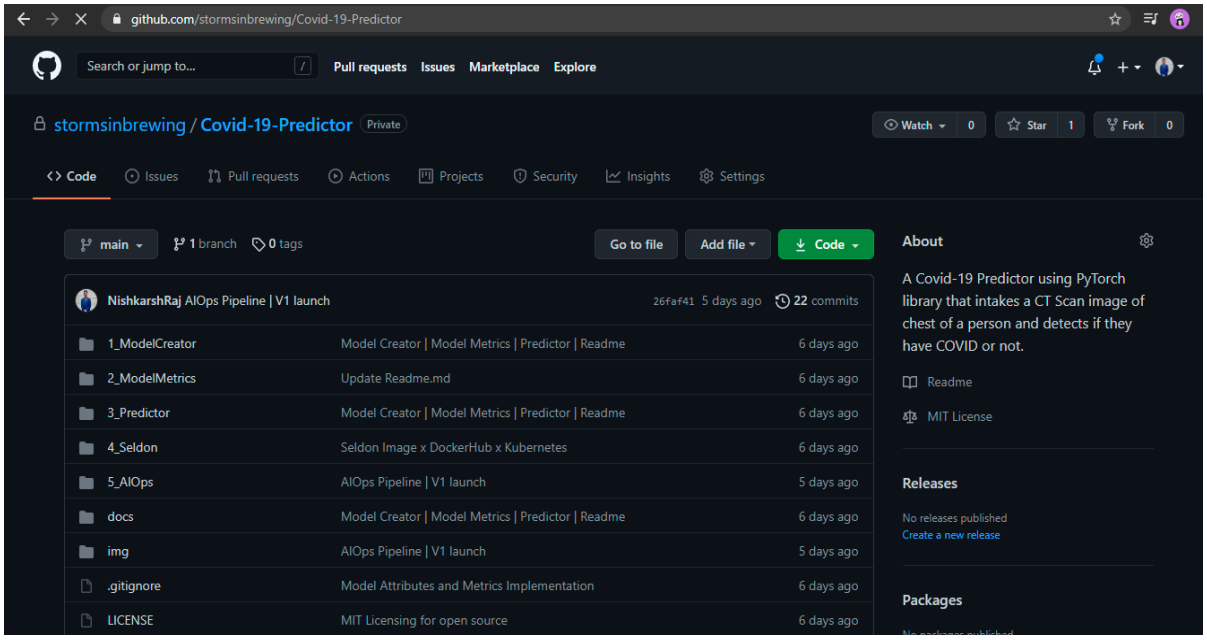
We are using the SCRUM model so we have divided the project into different modules. All these modules are listed in the product log

Product Log :
This is the most crucial document. It lists all the requirements of the project and can be considered as a to-do list. All these modules provide some functionality to the business process. Our product log is as follows

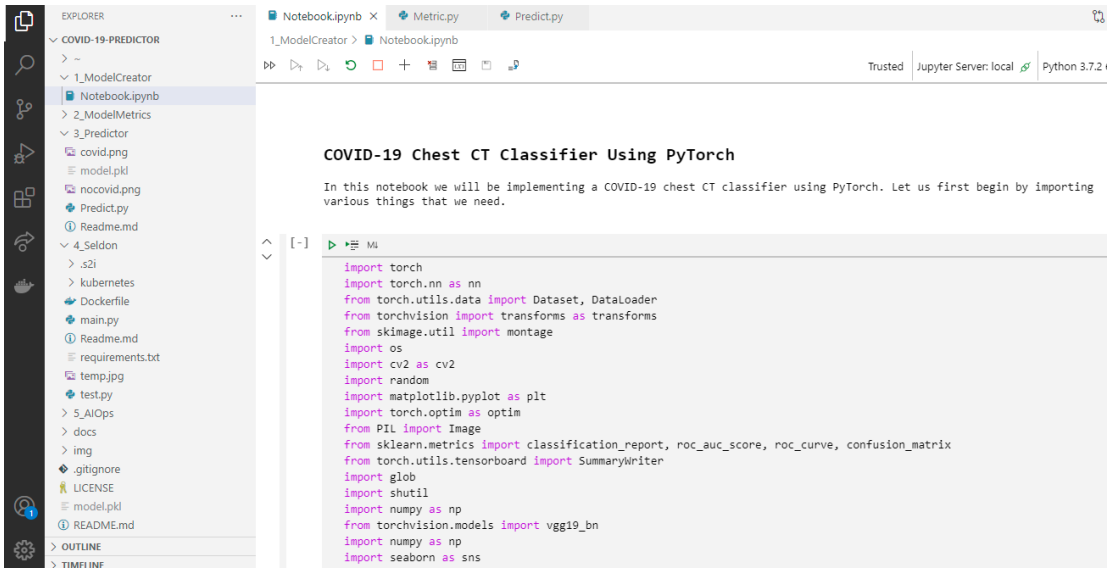
1. Research about a reliable and up-to-date data source for Covid-19
2. Create a GitHub repository to collaborate in a distributed environment
3. Create a .gitignore file to ignore python executables and dependencies
4. Create a .dockerignore file to optimize the production level Docker build
5. Create a pipeline to generate a machine learning model using pytorch which is serialized with pickle.
6. Create a metric evaluation code to find accuracy, specificity and Area under ROC of the model.
7. Create a predictor function that takes URL/local image as input and generates prediction of Covid19
8. Migrate the predictor function into Seldon framework
9. Create an optimized docker image of the Seldon workflow of the model
10. Expose REST Endpoints of the model with Docker and Kubernetes
11. Host a Model storage server with Verta AI's ModelDB
12. Log and Register Models in Mlflow
13. Serve gRPC endpoints in MLflow

Increment of Sprint I (Sprint Review)

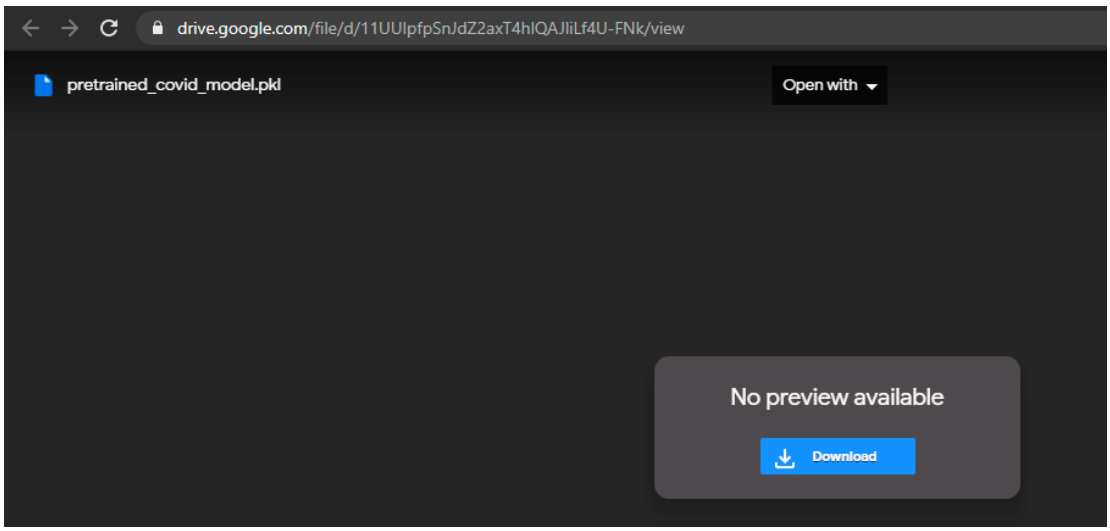
1. Host the project on GitHub -
   https://github.com/stormsinbrewing/Covid-19-Predictor



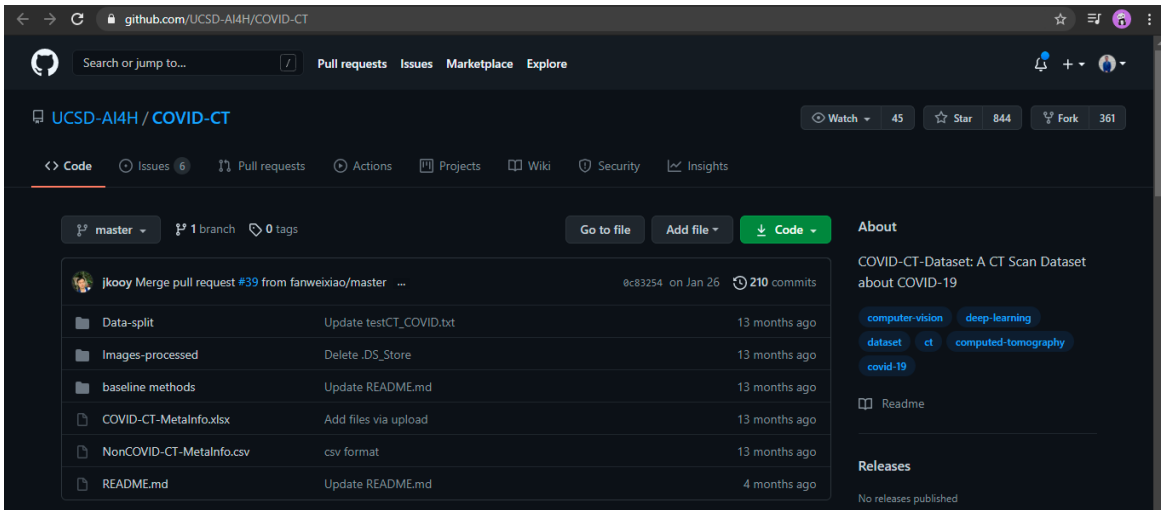
2. Create Pipeline in Jupyter Notebook to create a machine learning model with Pytorch library and pickle serialization



Since the model size is large and takes a lot of time to build, we store it safely on Google Drive for backup and sharing.



3. Download the dataset from https://github.com/UCSD-AI4H/COVID-CT



4. Write a model evaluation function in python to derive accuracy, specificity and AREA under ROC

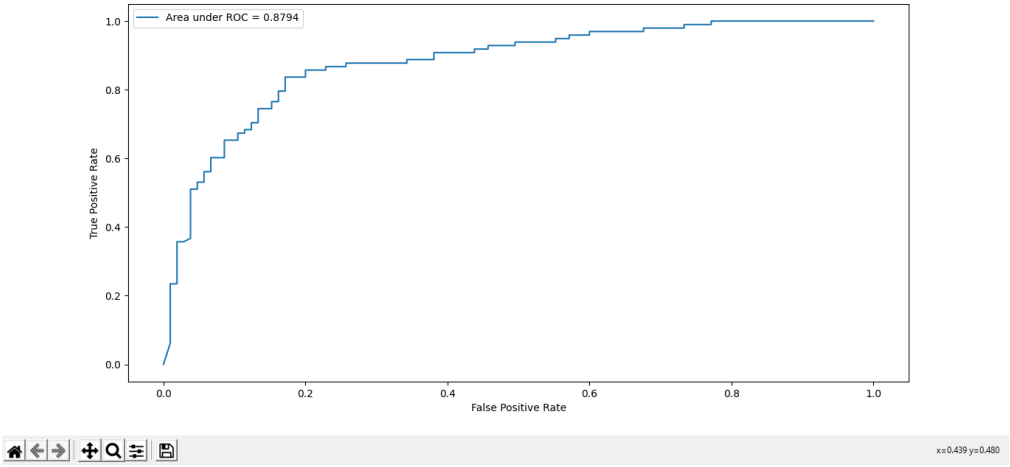
Notebook.ipynb | Metric.py | Predict.py

2_ModelMetrics > Metric.py > ...

```
1   ## Import Libraries
2   import torch
3   import torch.nn as nn
4   from torch.utils.data import Dataset, DataLoader
5   from torchvision import transforms as transforms
6   from skimage.util import montage
7   import os
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
 you can retrieve the original source code by accessing the object's source att
h tool to revert the changes.
  warnings.warn(msg, SourceChangeWarning)
C:\Python37\lib\site-packages\torch\serialization.py:656: SourceChangeWarning:
d. you can retrieve the original source code by accessing the object's source a
tch tool to revert the changes.
  warnings.warn(msg, SourceChangeWarning)
C:\Python37\lib\site-packages\torch\serialization.py:656: SourceChangeWarning:
s changed. you can retrieve the original source code by accessing the object's
e the patch tool to revert the changes.
  warnings.warn(msg, SourceChangeWarning)
C:\Python37\lib\site-packages\torch\serialization.py:656: SourceChangeWarning:
ou can retrieve the original source code by accessing the object's source attri
tool to revert the changes.
  warnings.warn(msg, SourceChangeWarning)
C:\Python37\lib\site-packages\torch\serialization.py:656: SourceChangeWarning:
 you can retrieve the original source code by accessing the object's source att
h tool to revert the changes.
  warnings.warn(msg, SourceChangeWarning)
------------------- Test Performance -------------------------------------
Accuracy         0.828
Sensitivity      0.837
Specificity      0.819
Area Under ROC   0.879
--------------------------------------------------------------------------------
PS D:\Projects\Major2\Covid-19_Prediction\2_ModelMetrics>
```



5. Write a Predictor function that takes an image of CT scan of the chest and predicts whether you are Covid positive or not based on the model created.

Notebook.ipynb | Metric.py | Predict.py

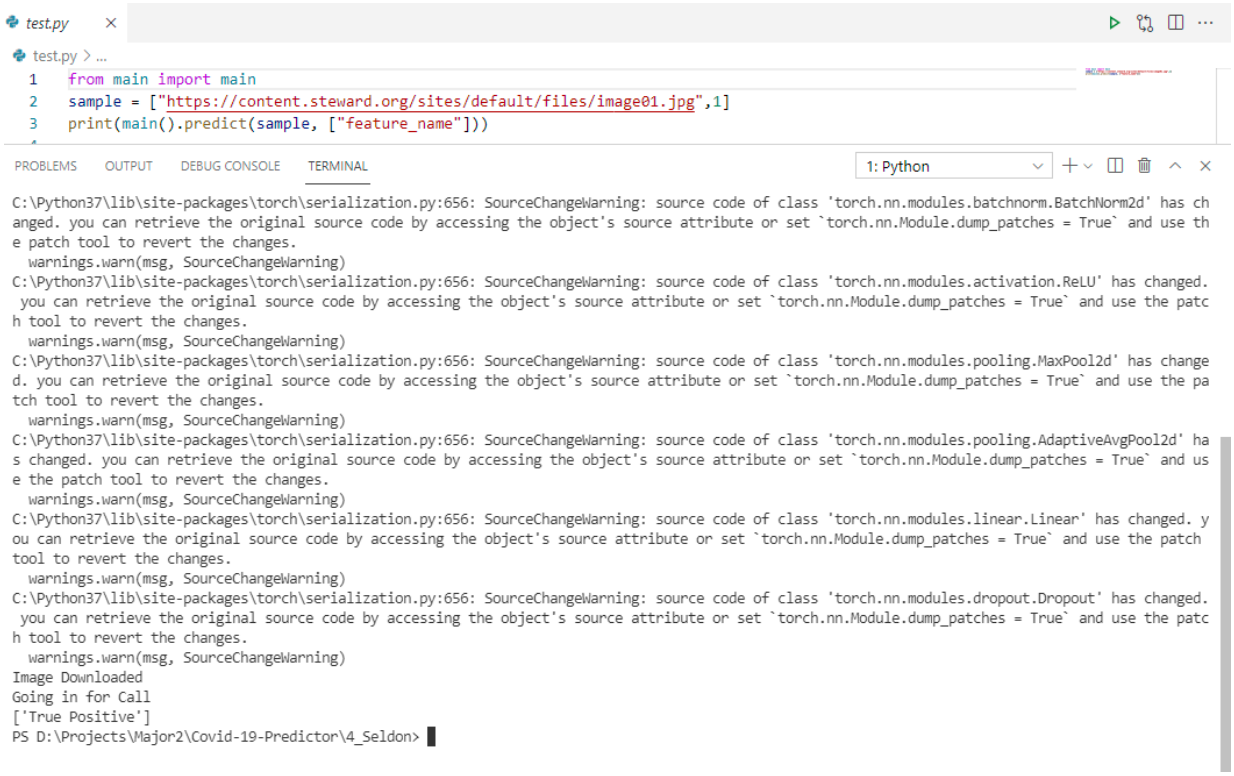
3_Predictor > Predict.py > CovidPredictor

```
1   import torch
2   from PIL import Image
3   from torchvision import transforms
4   import cv2
5   import numpy as np
6
7   def CovidPredictor(imgPath,label):
8       model = torch.load("model.pkl",map_location=torch.device('cpu'))
9       img = Image.open(imgPath)
10      img = cv2.cvtColor(np.float32(img), cv2.COLOR_BGRA2BGR)
11      img = transforms.ToTensor()(img).unsqueeze_(0)
12      output = model(img)
13      pred = output.argmax(dim=1, keepdim=True)
14      if pred.item() == 1 and label == 1:
15          return "True Positive"
```

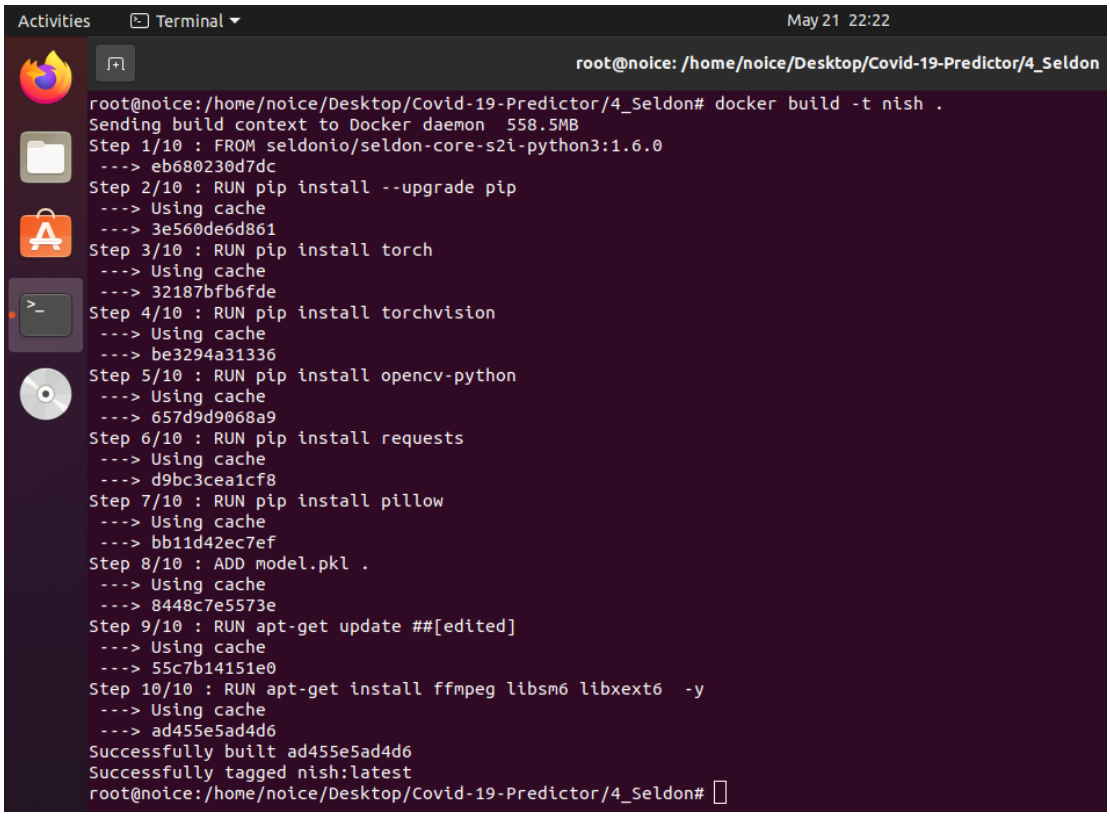
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
tool to revert the changes.
  warnings.warn(msg, SourceChangeWarning)
C:\Python37\lib\site-packages\torch\serialization.py:656: SourceChangeWarning: sou
 you can retrieve the original source code by accessing the object's source attrib
h tool to revert the changes.
  warnings.warn(msg, SourceChangeWarning)
True Positive
PS D:\Projects\Major2\Covid-19-Predictor\3_Predictor>
```

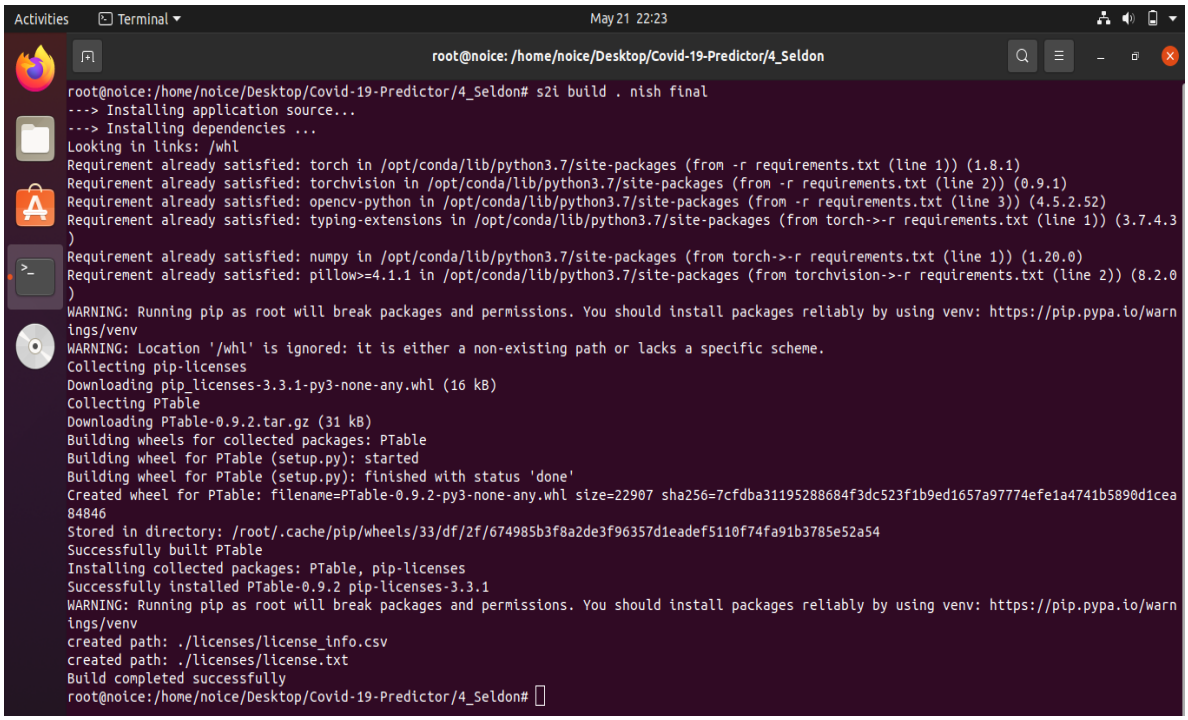
6. Migrate the predictor code to Seldon Framework to support REST and gRPC calls.



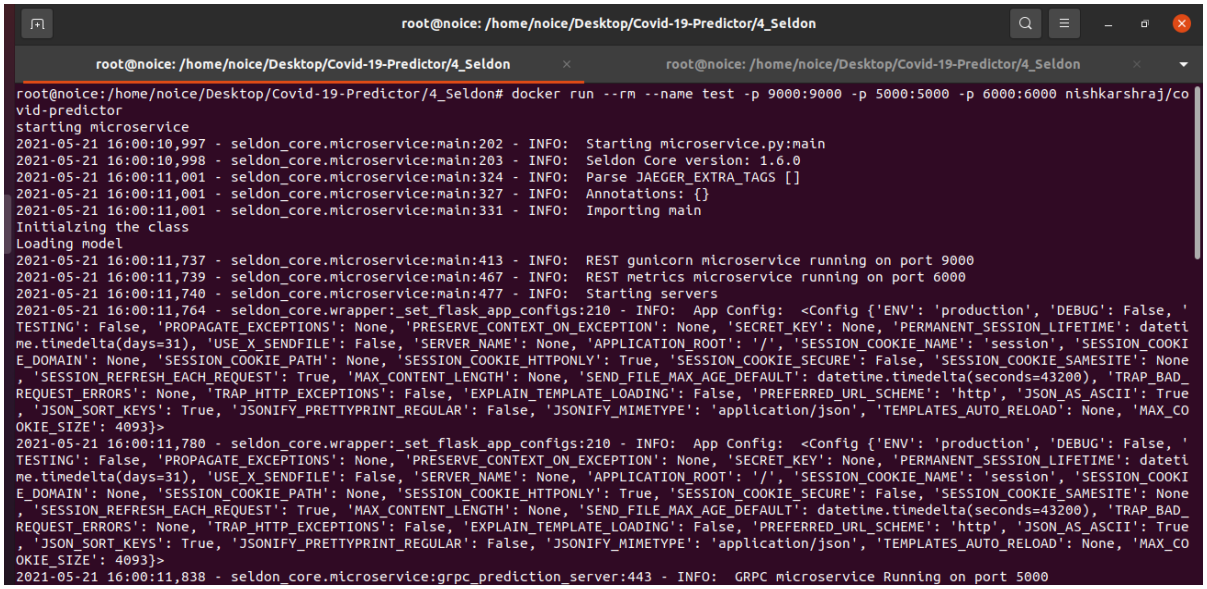
7. Build a base docker image for Seldon functionality



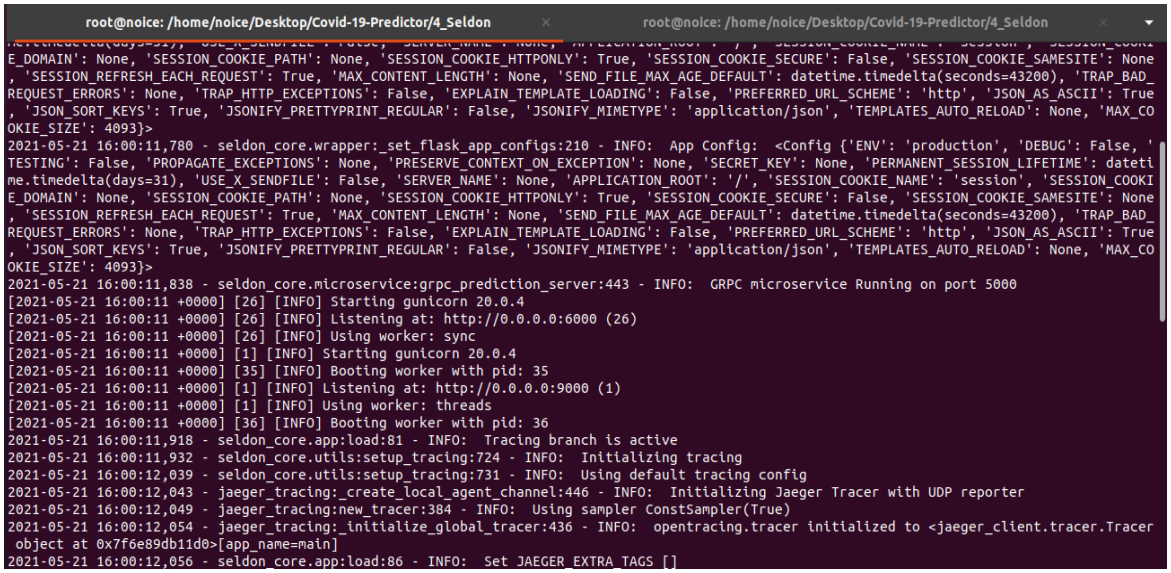
8. Create a seldon image using S2I functionality

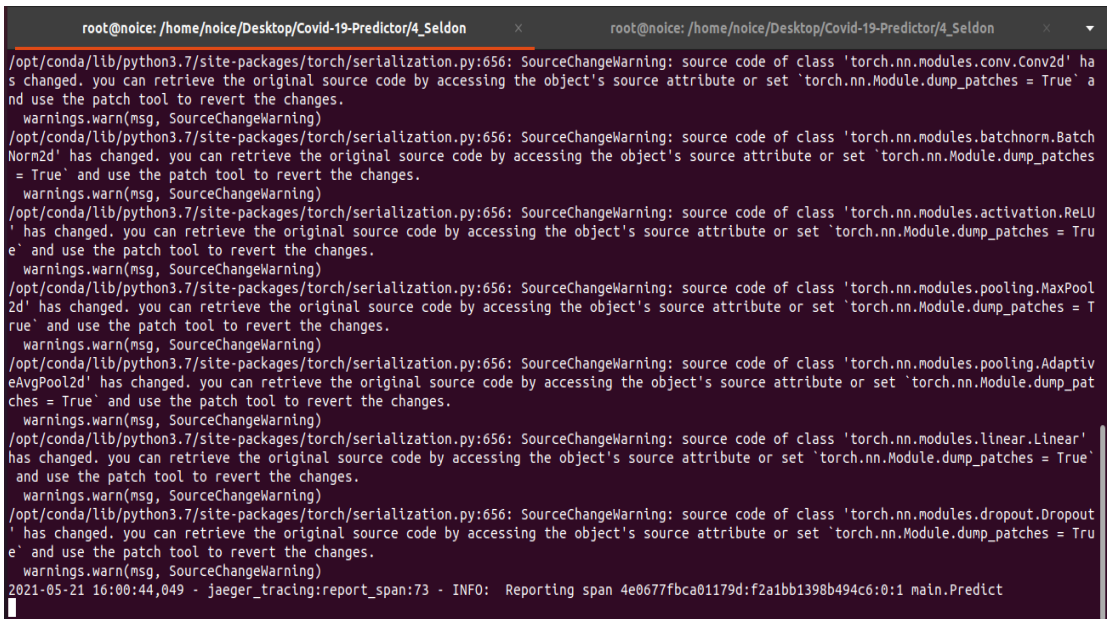


9. Expose REST Endpoints using Docker on local server with Seldon
   The REST Calls are routed to port 9000
   The gRPC Calls are routed to port 5000



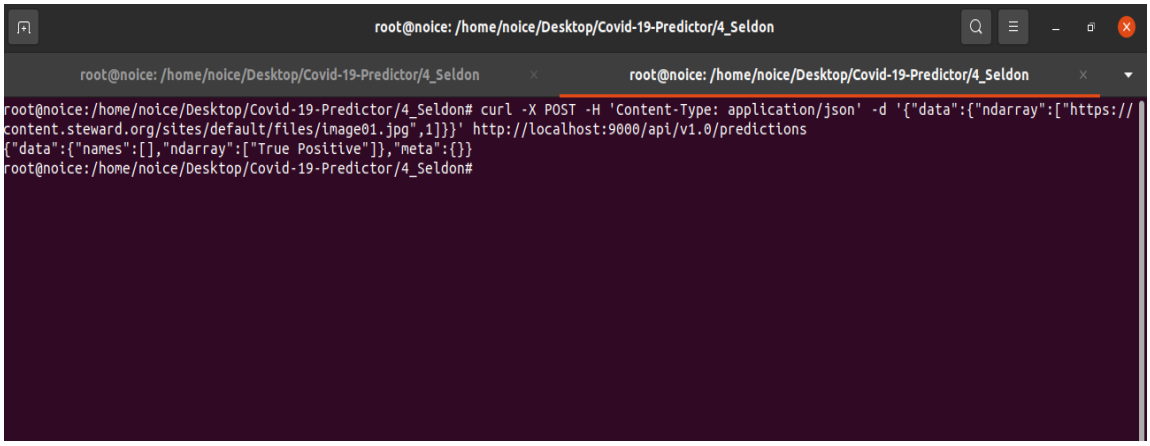
Set Jaeger Tags message defines the successful hosting of Seldon application



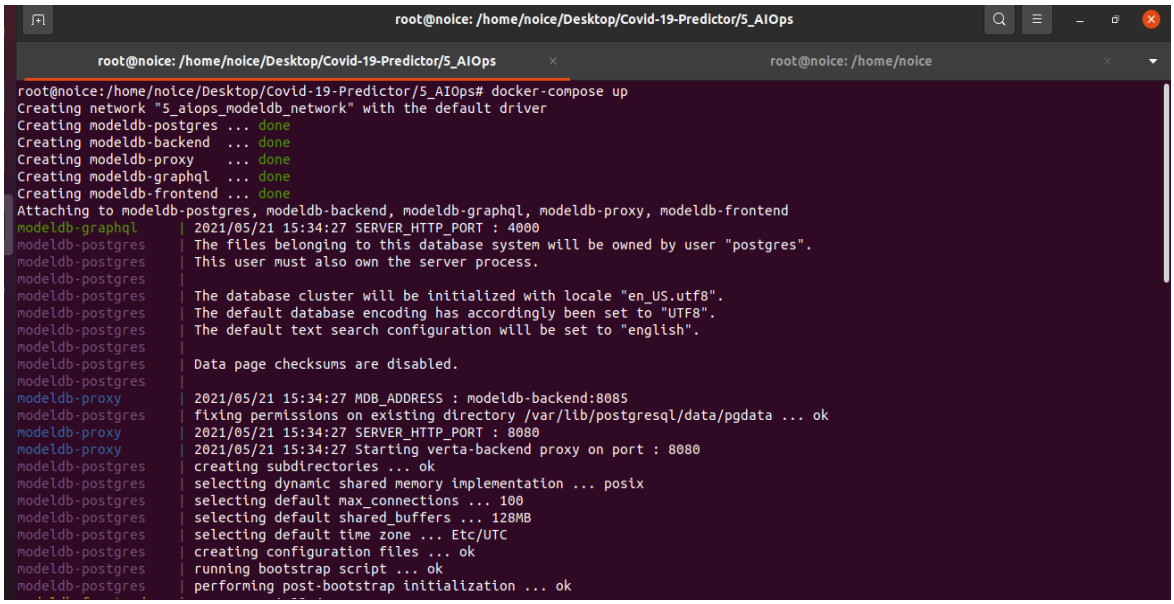


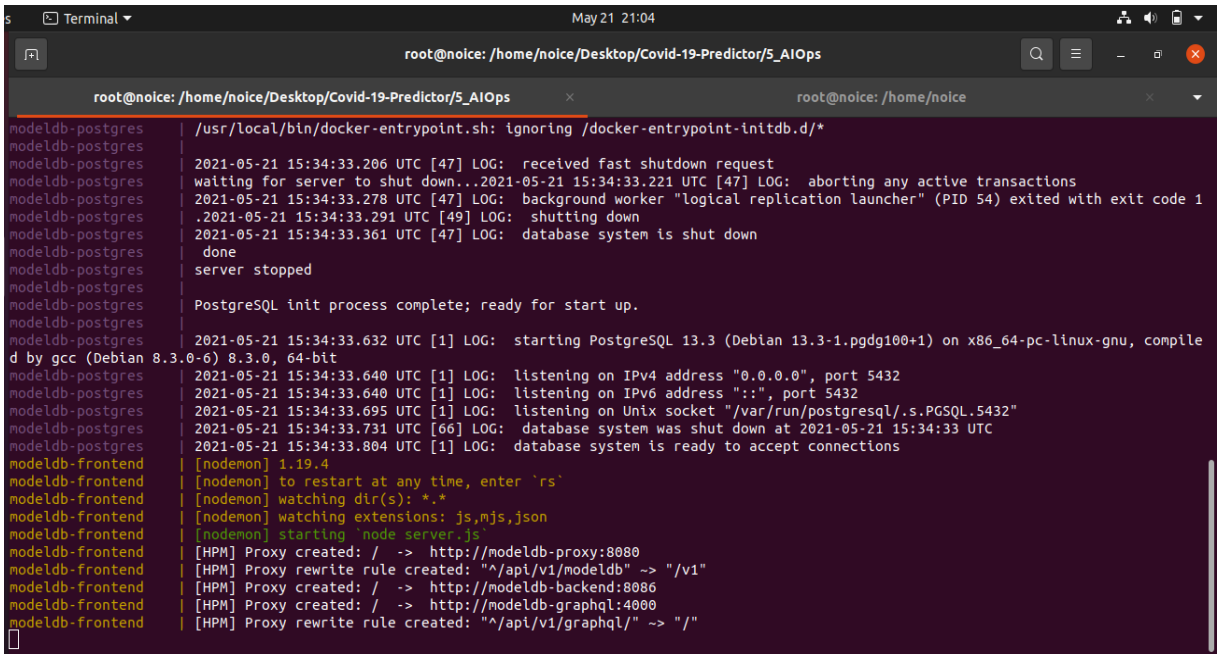
Whenever the endpoint is hit, the host shows a log of the invocation of the predict function.

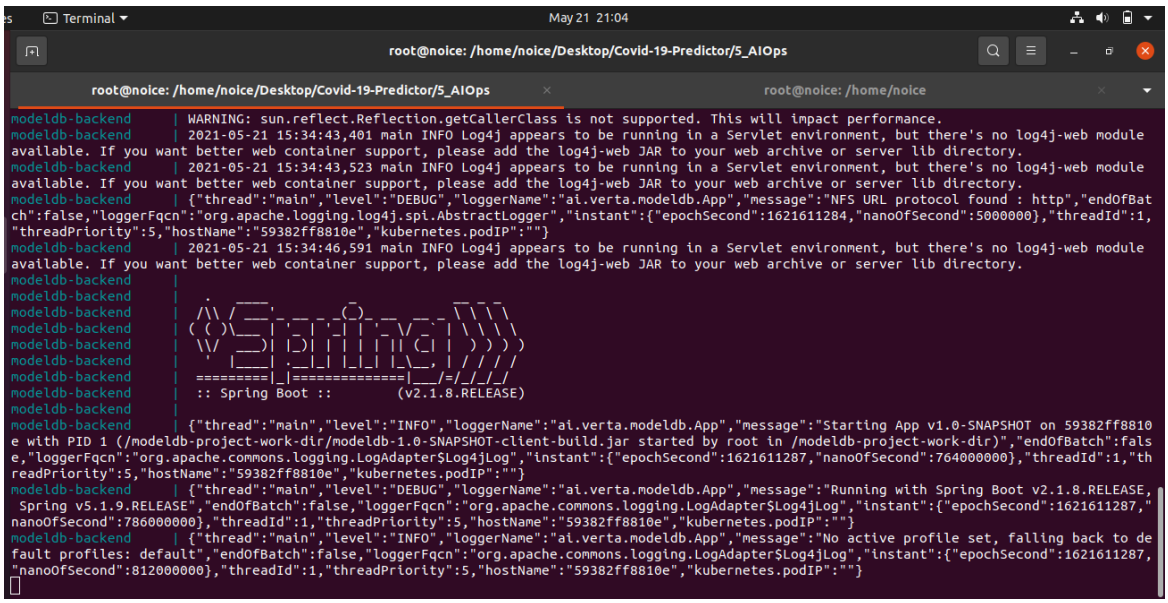
Hit the endpoint with a cURL POST call with bash or POSTMAN client.

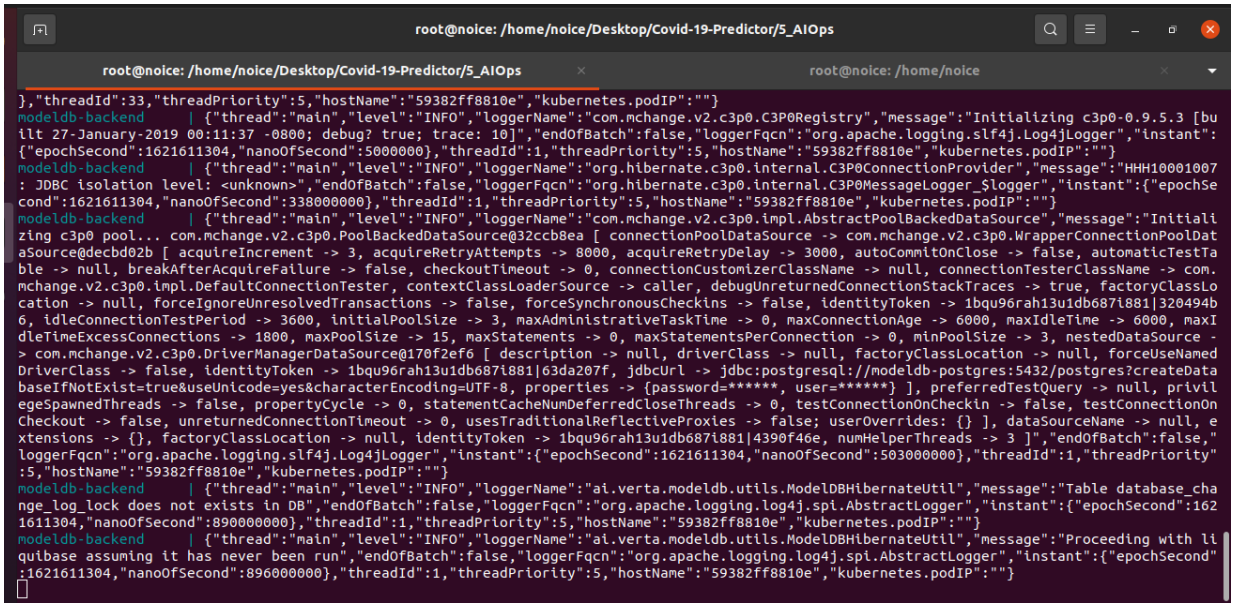


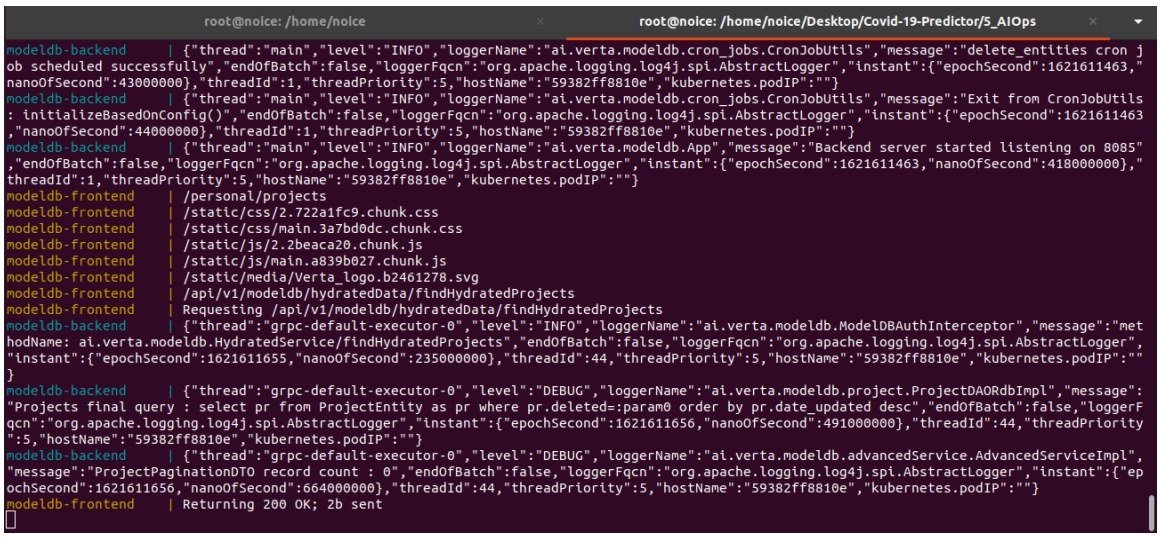
10. Host ModelDB on a local server for Model storage.
    Docker-compose is written to launch the entire stack of Model DB including frontend, backend, postgres and proxy server.



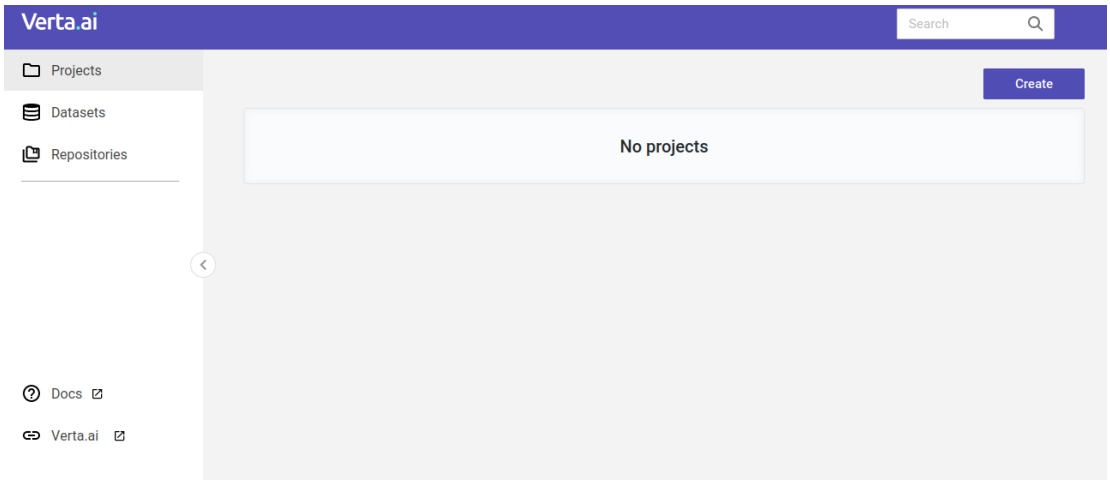



Terminal
May 21 21:04
root@noice: /home/noice/Desktop/Covid-19-Predictor/5_AIOps
root@noice: /home/noice/Desktop/Covid-19-Predictor/5_AIOps
root@noice: /home/noice
modeldb-backend | WARNING: sun.reflect.Reflection.getCallerClass is not supported. This will impact performance.
modeldb-backend | 2021-05-21 15:34:43,401 main INFO Log4j appears to be running in a Servlet environment, but there's no log4j-web module available. If you want better web container support, please add the log4j-web JAR to your web archive or server lib directory.
modeldb-backend | 2021-05-21 15:34:43,523 main INFO Log4j appears to be running in a Servlet environment, but there's no log4j-web module available. If you want better web container support, please add the log4j-web JAR to your web archive or server lib directory.
modeldb-backend | {"thread":"main","level":"DEBUG","loggerName":"ai.verta.modeldb.App","message":"NFS URL protocol found : http","endOfBatch":false,"loggerFqcn":"org.apache.logging.log4j.spi.AbstractLogger","instant":{"epochSecond":1621611284,"nanoOfSecond":5000000},"threadId":1,"threadPriority":5,"hostName":"59382ff8810e","kubernetes.podIP":""}
modeldb-backend | 2021-05-21 15:34:46,591 main INFO Log4j appears to be running in a Servlet environment, but there's no log4j-web module available. If you want better web container support, please add the log4j-web JAR to your web archive or server lib directory.
modeldb-backend |
modeldb-backend | .   ____   _   __ _ _
modeldb-backend | /\\ / ___'_ __ _ _(_)_ __ __ _ \ \ \ \
modeldb-backend | (( )___ | '_ | '_| | '_ \/ _` | \ \ \ \
modeldb-backend | \\/ ___)| |_)| | | | | || (_| |))))
modeldb-backend | ' |____| .__|_| |_|_| |___, | / / / /
modeldb-backend | =========|_|==============|___/=/_/_/_/
modeldb-backend | :: Spring Boot :: (v2.1.8.RELEASE)
modeldb-backend |
modeldb-backend | {"thread":"main","level":"INFO","loggerName":"ai.verta.modeldb.App","message":"Starting App v1.0-SNAPSHOT on 59382ff8810e with PID 1 (/modeldb-project-work-dir/modeldb-1.0-SNAPSHOT-client-build.jar started by root in /modeldb-project-work-dir)","endOfBatch":false,"loggerFqcn":"org.apache.commons.logging.LogAdapter$Log4jLog","instant":{"epochSecond":1621611287,"nanoOfSecond":764000000},"threadId":1,"threadPriority":5,"hostName":"59382ff8810e","kubernetes.podIP":""}
modeldb-backend | {"thread":"main","level":"DEBUG","loggerName":"ai.verta.modeldb.App","message":"Running with Spring Boot v2.1.8.RELEASE, Spring v5.1.9.RELEASE","endOfBatch":false,"loggerFqcn":"org.apache.commons.logging.LogAdapter$Log4jLog","instant":{"epochSecond":1621611287,"nanoOfSecond":786000000},"threadId":1,"threadPriority":5,"hostName":"59382ff8810e","kubernetes.podIP":""}
modeldb-backend | {"thread":"main","level":"INFO","loggerName":"ai.verta.modeldb.App","message":"No active profile set, falling back to default profiles: default","endOfBatch":false,"loggerFqcn":"org.apache.commons.logging.LogAdapter$Log4jLog","instant":{"epochSecond":1621611287,"nanoOfSecond":812000000},"threadId":1,"threadPriority":5,"hostName":"59382ff8810e","kubernetes.podIP":""}

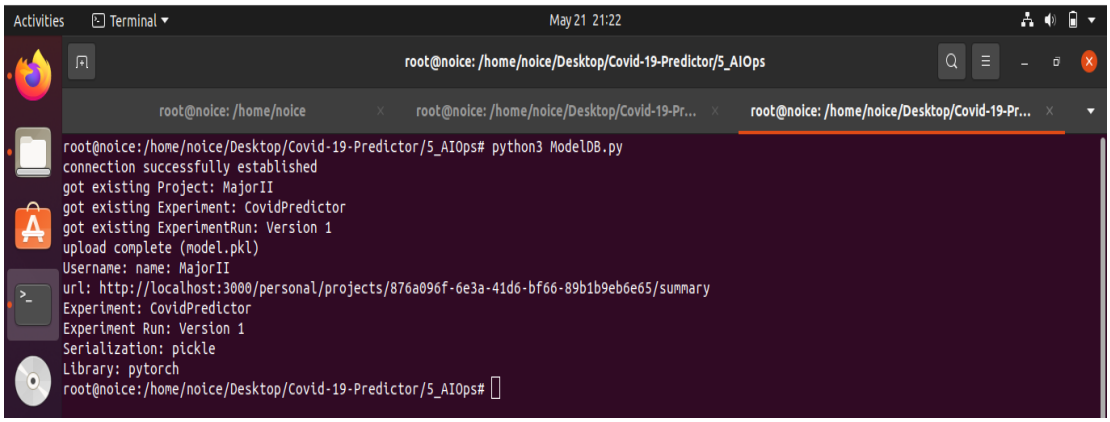

root@noice: /home/noice/Desktop/Covid-19-Predictor/5_AIOps
root@noice: /home/noice/Desktop/Covid-19-Predictor/5_AIOps
root@noice: /home/noice
},"threadId":33,"threadPriority":5,"hostName":"59382ff8810e","kubernetes.podIP":""}
modeldb-backend | {"thread":"main","level":"INFO","loggerName":"com.mchange.v2.c3p0.C3P0Registry","message":"Initializing c3p0-0.9.5.3 [built 27-January-2019 00:11:37 -0800; debug? true; trace: 10]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","instant":{"epochSecond":1621611304,"nanoOfSecond":5000000},"threadId":1,"threadPriority":5,"hostName":"59382ff8810e","kubernetes.podIP":""}
modeldb-backend | {"thread":"main","level":"INFO","loggerName":"org.hibernate.c3p0.internal.C3P0ConnectionProvider","message":"HHH10001007: JDBC isolation level: <unknown>","endOfBatch":false,"loggerFqcn":"org.hibernate.c3p0.internal.C3P0MessageLogger_$logger","instant":{"epochSecond":1621611304,"nanoOfSecond":338000000},"threadId":1,"threadPriority":5,"hostName":"59382ff8810e","kubernetes.podIP":""}
modeldb-backend | {"thread":"main","level":"INFO","loggerName":"com.mchange.v2.c3p0.impl.AbstractPoolBackedDataSource","message":"Initializing c3p0 pool... com.mchange.v2.c3p0.PoolBackedDataSource@32ccb8ea [connectionPoolDataSource -> com.mchange.v2.c3p0.WrapperConnectionPoolDataSource@decbd02b [acquireIncrement -> 3, acquireRetryAttempts -> 8000, acquireRetryDelay -> 3000, autoCommitOnClose -> false, automaticTestTable -> null, breakAfterAcquireFailure -> false, checkoutTimeout -> 0, connectionCustomizerClassName -> null, connectionTesterClassName -> com.mchange.v2.c3p0.impl.DefaultConnectionTester, contextClassLoaderSource -> caller, debugUnreturnedConnectionStackTraces -> true, factoryClassLocation -> null, forceIgnoreUnresolvedTransactions -> false, forceSynchronousCheckins -> false, identityToken -> 1bqu96rah13u1db687i881|320494b6, idleConnectionTestPeriod -> 3600, initialPoolSize -> 3, maxAdministrativeTaskTime -> 0, maxConnectionAge -> 6000, maxIdleTime -> 6000, maxIdleTimeExcessConnections -> 1800, maxPoolSize -> 15, maxStatements -> 0, maxStatementsPerConnection -> 0, minPoolSize -> 3, nestedDataSource -> com.mchange.v2.c3p0.DriverManagerDataSource@170f2ef6 [description -> null, driverClass -> null, factoryClassLocation -> null, forceUseNamedDriverClass -> false, identityToken -> 1bqu96rah13u1db687i881|63da207f, jdbcUrl -> jdbc:postgresql://modeldb-postgres:5432/postgres?createDatabaseIfNotExist=true&useUnicode=yes&characterEncoding=UTF-8, properties -> {password=******, user=******}], preferredTestQuery -> null, privilegeSpawnedThreads -> false, propertyCycle -> 0, statementCacheNumDeferredCloseThreads -> 0, testConnectionOnCheckin -> false, testConnectionOnCheckout -> false, unreturnedConnectionTimeout -> 0, usesTraditionalReflectiveProxies -> false; userOverrides: {}], dataSourceName -> null, extensions -> {}, factoryClassLocation -> null, identityToken -> 1bqu96rah13u1db687i881|4390f46e, numHelperThreads -> 3]","endOfBatch":false,"loggerFqcn":"org.apache.logging.slf4j.Log4jLogger","instant":{"epochSecond":1621611304,"nanoOfSecond":503000000},"threadId":1,"threadPriority":5,"hostName":"59382ff8810e","kubernetes.podIP":""}
modeldb-backend | {"thread":"main","level":"INFO","loggerName":"ai.verta.modeldb.utils.ModelDBHibernateUtil","message":"Table database_change_log_lock does not exists in DB","endOfBatch":false,"loggerFqcn":"org.apache.logging.log4j.spi.AbstractLogger","instant":{"epochSecond":1621611304,"nanoOfSecond":890000000},"threadId":1,"threadPriority":5,"hostName":"59382ff8810e","kubernetes.podIP":""}
modeldb-backend | {"thread":"main","level":"INFO","loggerName":"ai.verta.modeldb.utils.ModelDBHibernateUtil","message":"Proceeding with liquibase assuming it has never been run","endOfBatch":false,"loggerFqcn":"org.apache.logging.log4j.spi.AbstractLogger","instant":{"epochSecond":1621611304,"nanoOfSecond":896000000},"threadId":1,"threadPriority":5,"hostName":"59382ff8810e","kubernetes.podIP":""}

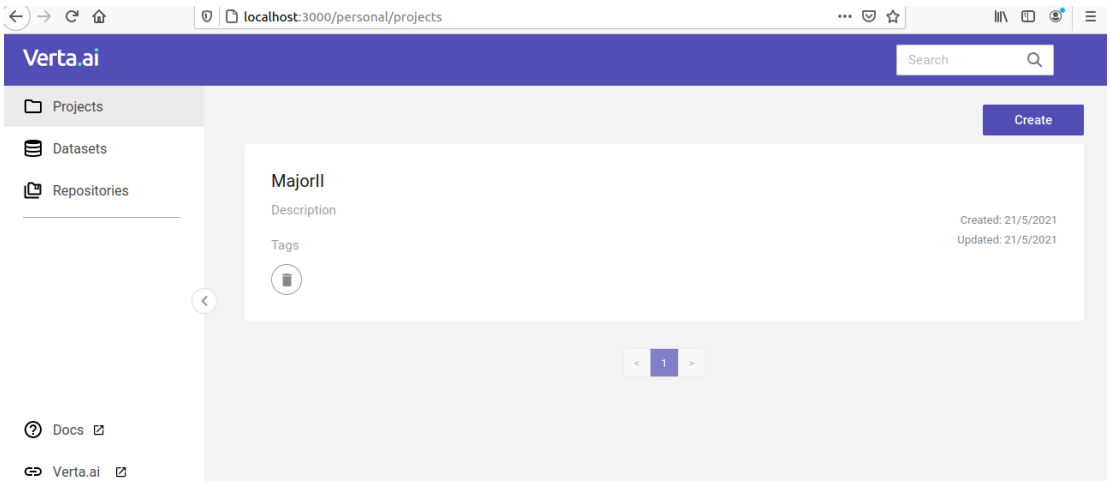


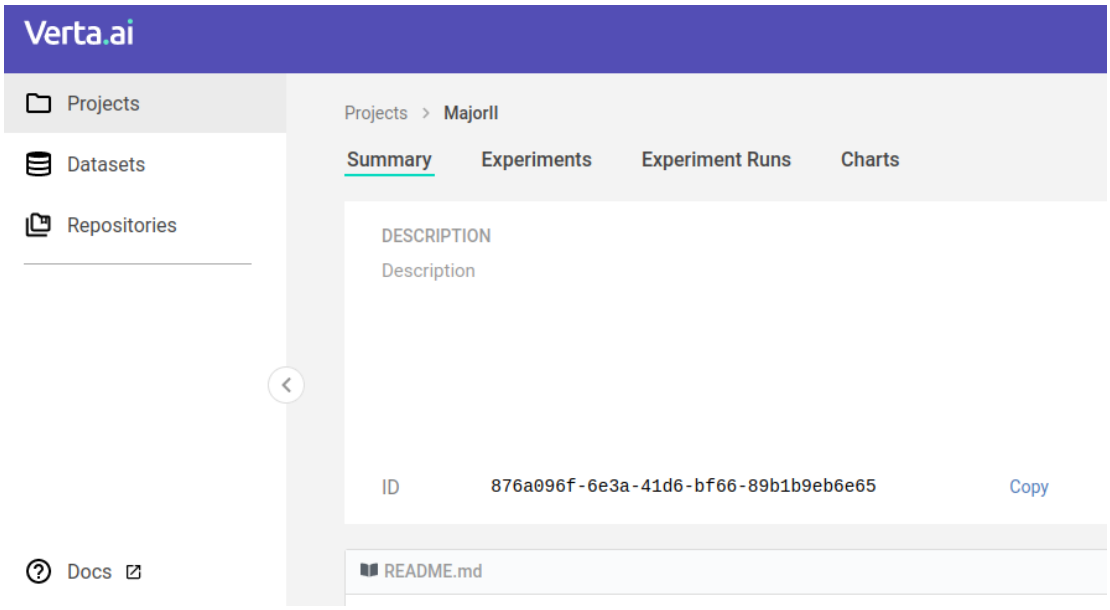
When everything is up and running, the frontend returns HTTP 200 status code on the logging server.

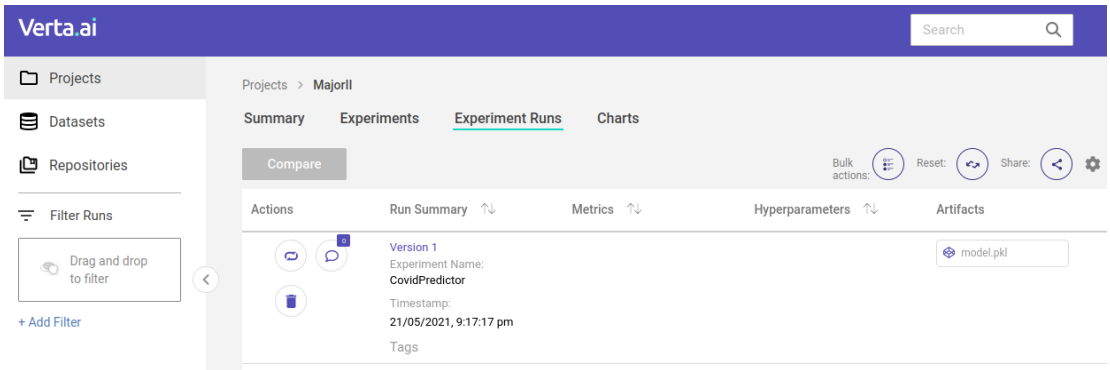


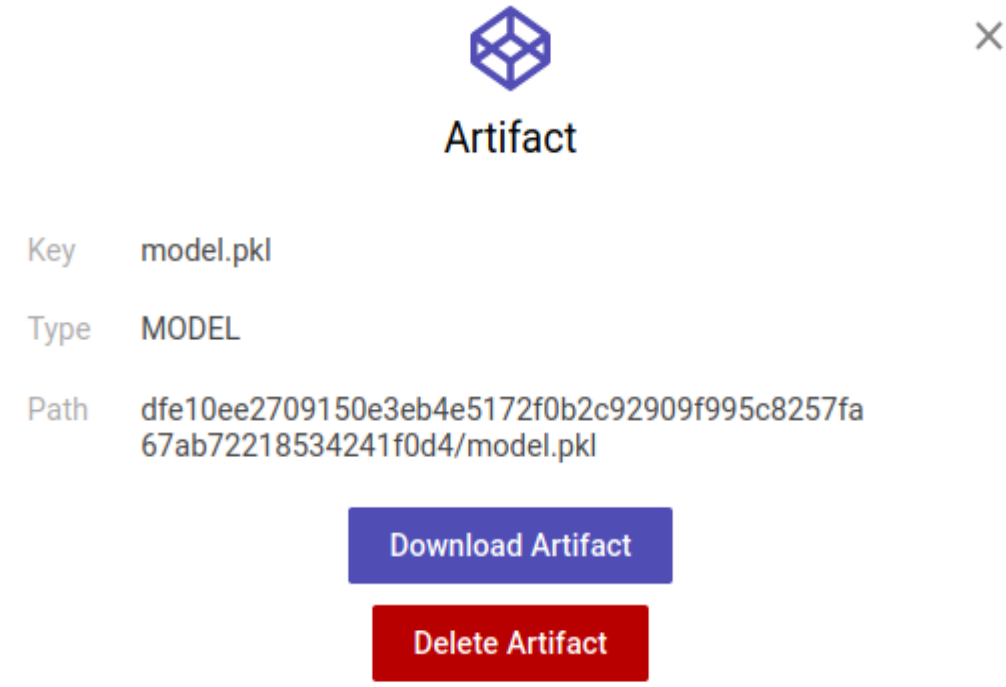
11. Log Model in ModelDB with Python SDK

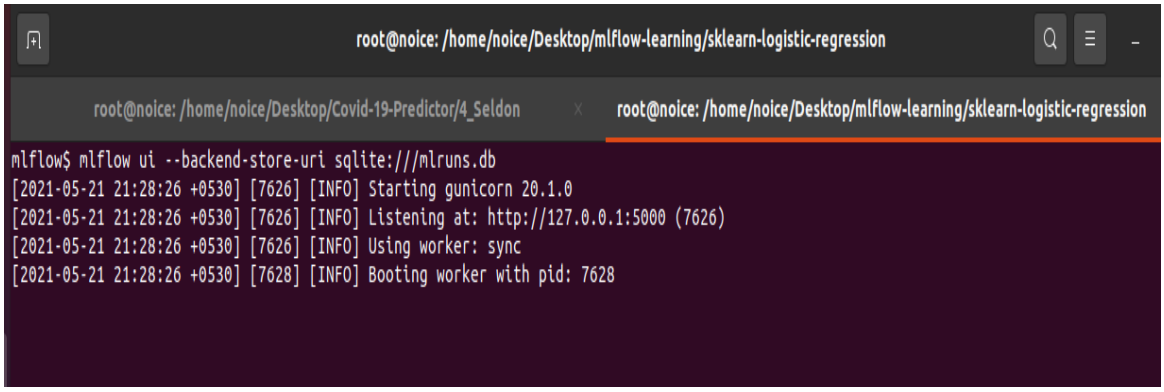


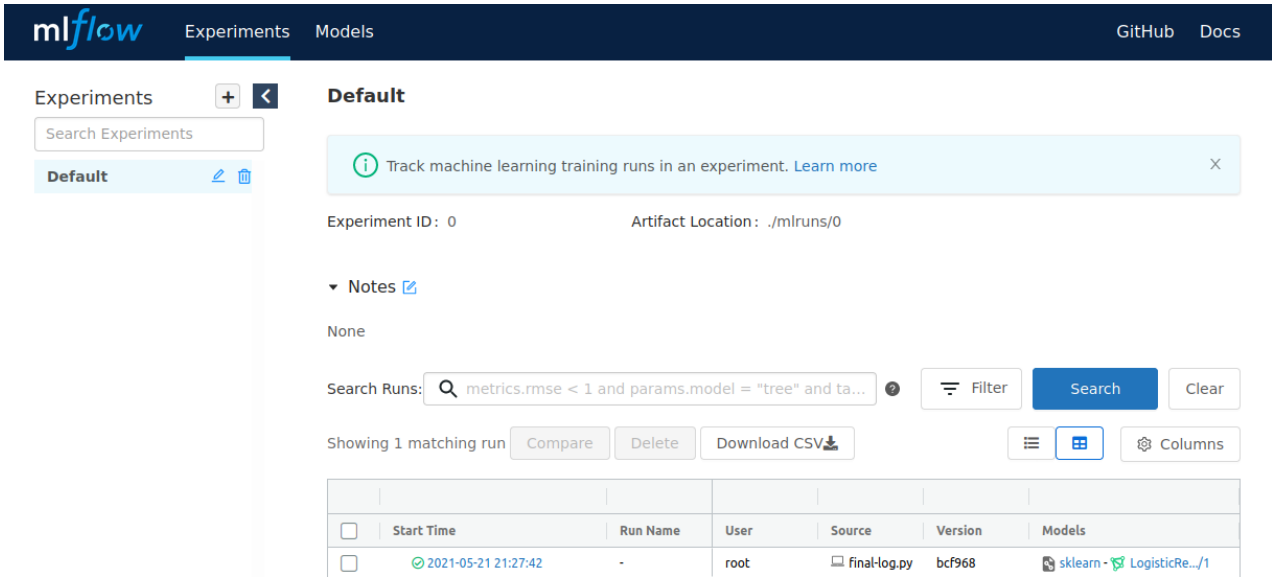

localhost:3000/personal/projects
Verta.ai
Search
Projects
Datasets
Repositories
Create
MajorII
Description
Tags
Created: 21/5/2021
Updated: 21/5/2021
1
Docs
Verta.ai

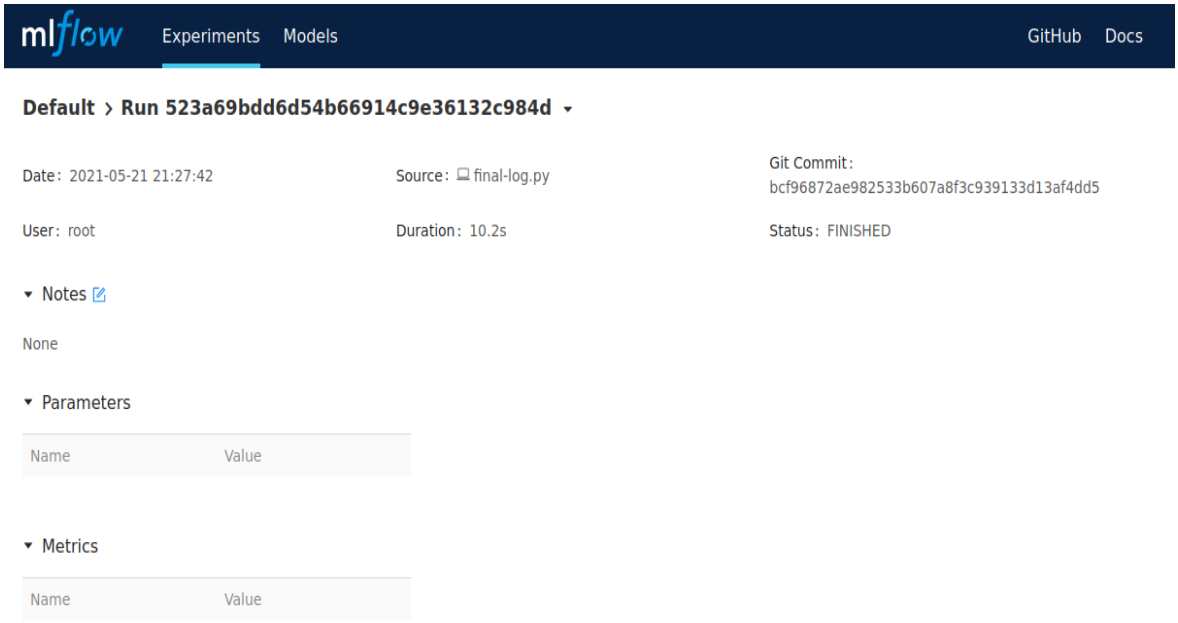

Verta.ai
Projects
Datasets
Repositories
Projects > MajorII
Summary
Experiments
Experiment Runs
Charts
DESCRIPTION
Description
ID
876a096f-6e3a-41d6-bf66-89b1b9eb6e65
Copy
README.md
Docs


Verta.ai
Search
Projects
Datasets
Repositories
Filter Runs
Drag and drop to filter
+ Add Filter
Projects > MajorII
Summary
Experiments
Experiment Runs
Charts
Compare
Bulk actions:
Reset:
Share:
Actions
Run Summary
Metrics
Hyperparameters
Artifacts
Version 1
Experiment Name:
CovidPredictor
Timestamp:
21/05/2021, 9:17:17 pm
Tags
model.pkl


Artifact
Key
model.pkl
Type
MODEL
Path
dfe10ee2709150e3eb4e5172f0b2c92909f995c8257fa
67ab72218534241f0d4/model.pkl
Download Artifact
Delete Artifact


root@noice: /home/noice/Desktop/mlflow-learning/sklearn-logistic-regression
root@noice: /home/noice/Desktop/Covid-19-Predictor/4_Seldon
root@noice: /home/noice/Desktop/mlflow-learning/sklearn-logistic-regression
mlflow$ mlflow ui --backend-store-uri sqlite:///mlruns.db
[2021-05-21 21:28:26 +0530] [7626] [INFO] Starting gunicorn 20.1.0
[2021-05-21 21:28:26 +0530] [7626] [INFO] Listening at: http://127.0.0.1:5000 (7626)
[2021-05-21 21:28:26 +0530] [7626] [INFO] Using worker: sync
[2021-05-21 21:28:26 +0530] [7628] [INFO] Booting worker with pid: 7628


mlflow
Experiments
Models
GitHub
Docs
Experiments
Search Experiments
Default
Default
Track machine learning training runs in an experiment. Learn more
Experiment ID: 0
Artifact Location: ./mlruns/0
Notes
None
Search Runs:
metrics.rmse < 1 and params.model = "tree" and ta...
Filter
Search
Clear
Showing 1 matching run
Compare
Delete
Download CSV
Columns
Start Time
Run Name
User
Source
Version
Models
2021-05-21 21:27:42
-
root
final-log.py
bcf968
sklearn - LogisticRe.../1


mlflow
Experiments
Models
GitHub
Docs
Default › Run 523a69bdd6d54b66914c9e36132c984d
Date: 2021-05-21 21:27:42
Source: final-log.py
Git Commit: bcf96872ae982533b607a8f3c939133d13af4dd5
User: root
Duration: 10.2s
Status: FINISHED
Notes
None
Parameters
Name
Value
Metrics
Name
Value

References

1. Dong, E., Du, H. & Gardner, L. An interactive web-based dashboard to track COVID-19 in real time. Lancet Infect. Dis. https://doi.org/10.1016/S1473-3099(20)30120-1 (2020).

2. Gozes, O. et al. Rapid AI development cycle for the coronavirus (COVID-19) pandemic: initial results for automated detection & patient monitoring using deep learning CT image analysis. arXiv e-prints 2003, arXiv:2003.05037 (2020).

3. Song, Y. et al. Deep learning Enables Accurate Diagnosis of Novel Coronavirus (COVID-19) with CT images. medRxiv https://doi.org/10.1101/2020.02.23.20026930 (2020).

4. Wang, S. et al. A deep learning algorithm using CT images to screen for CoronaVirus Disease (COVID-19). medRxiv, https://doi.org/10.1101/2020.02.14.20023028 (2020).

5. Jin, C. et al. Development and evaluation of an AI system for COVID-19 diagnosis. medRxiv, https://doi.org/10.1101/2020.03.20.20039834 (2020).

6. Punn, N. S. & Agarwal, S. Automated diagnosis of COVID-19 with limited posteroanterior chest X-ray images using fine-tuned deep neural networks. arXiv:2004.11676 [cs, eess] (2020).

7. Tostmann, A. et al. Strong associations and moderate predictive value of early symptoms for SARS-CoV-2 test positivity among healthcare workers, the Netherlands, March 2020. Eurosurveillance 25, 2000508 (2020).

8. Feng, C. et al. A novel triage tool of artificial intelligence assisted diagnosis aid system for suspected COVID-19 pneumonia in fever clinics. medRxiv, https://doi.org/10.1101/2020.03.19.20039099 (2020).
9. Punn, N. S., Sonbhadra, S. K. & Agarwal, S. COVID-19 Epidemic Analysis using Machine Learning and Deep Learning Algorithms. medRxiv, https://doi.org/10.1101/2020.04.08.20057679 (2020).

10. Mei, X. et al. Artificial intelligence–enabled rapid diagnosis of patients with COVID-19. Nat. Med. 26, 1224–1228 (2020).

11. COVID-19-Government Data. https://data.gov.il/dataset/covid-19 (2020).

12. The Novel CoronavirusIsrael Ministry of Health. https://govextra.gov.il/ministry-of-health/corona/corona-virus-en/ (2020).

13. COVID-19-Government Data Information. https://data.gov.il/dataset/covid-19/resource/3f5c975e-7196-454b-8c5b-ef85881f78db/download/-readme.pdf (2020).

14. Struyf, T. et al. Signs and symptoms to determine if a patient presenting in primary care or hospital outpatient settings has COVID‐19 disease. Cochrane Database Syst. Rev., https://doi.org/10.1002/14651858.CD013665 (2020).

15. Liu, Y., Gayle, A. A., Wilder-Smith, A. & Rocklöv, J. The reproductive number of COVID-19 is higher compared to SARS coronavirus. J. Travel Med. 27 (2020).

16. Jin, J.-M. et al. Gender Differences in Patients With COVID-19: Focus on Severity and Mortality. Front. Public Health 8 (2020).

17. BMJ GH Blogs. Sex, gender and COVID-19: Disaggregated data and health disparities. BMJ Global Health blog https://blogs.bmj.com/bmjgh/2020/03/24/sex-gender-and-covid-19-disaggregated-data-and-health-disparities/ (2020).

18. Whittington, A. M. et al. Coronavirus: rolling out community testing for COVID-19 in the NHS. BMJ Opinion https://blogs.bmj.com/bmj/2020/02/17/coronavirus-rolling-out-community-testing-for-covid-19-in-the-nhs/ (2020).

19. Menni, C. et al. Real-time tracking of self-reported symptoms to predict potential COVID-19. Nat. Med. 26, 1037–1040 (2020).

20. Hastie, T., Tibshirani, R. & Friedman, J. In The Elements of Statistical Learning: Data Mining, Inference, and Prediction (eds. Hastie, T., Tibshirani, R. & Friedman, J.) 337–387 (Springer, 2009).

21. Fernández-Delgado, M., Cernadas, E., Barro, S. & Amorim, D. Do we need hundreds of classifiers to solve real world classification problems? J. Mach. Learn. Res. 15, 3133–3181 (2014).

22. Omar, K. B. A. XGBoost and LGBM for Porto Seguro's Kaggle challenge: A comparison Semester Project (ETH Zurich, 2018).

23. Josse, J., Prost, N., Scornet, E. & Varoquaux, G. On the consistency of supervised learning with missing values. arXiv:1902.06931 [cs, math, stat] (2019).

24. Chen, T. & Guestrin, C. XGBoost: A scalable tree boosting system. in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 785–794 (Association for Computing Machinery, 2016).

25. Ke, G. et al. In Advances in Neural Information Processing Systems 30 (eds. Guyon, I. et al.) 3146–3154 (Curran Associates, Inc., 2017).

26. Raskutti, G., Wainwright, M. J. & Yu, B. Early stopping for nonparametric regression: An optimal data-dependent stopping rule. in 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton) 1318–1325 (2011).

27. Lundberg, S. & Lee, S.-I. A Unified Approach to Interpreting Model Predictions. arXiv:1705.07874 [cs, stat] (2017).

28. Lundberg, S. M. et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. Nat. Biomed. Eng. 2, 749–760 (2018).

29. Efron, B. & Tibshirani, R. J. An Introduction to the Bootstrap. (CRC press, 1994).

---------- ----------

Signature of HoD Signature of Mentor