

**Fraud Detector**

**DLBDSMTP01**

**32009177**

**Machine Learning Project:**

**From Model To Production**

## Table of Contents

<b>Title Page</b>	1	
<b>Table of Contents</b>	2	
<b>Introduction</b>	3	
<b>Main Part</b>		
<b>1. Fraud detection Machine Learning Model</b>		
1.1 Isolation Forest	4	
1.2 Conceptual architecture	5	
<b>2. Flask API and Docker</b>	6	
2.1 User Input, Fraud Detection with Flask API	7	
2.2 Flask Codes	8	
2.3 Create Docker Image and Container	9	
<b>3. Challenge and Constraints</b>	12	
<b>4. MLFlow tracking, monitoring, automate</b>	6	
4.1 Model tracking with MLFlow	14	
4.2 MLFlow codes	15	
4.3 Automate scheduled/triggered training	15	
<b>5. MLOps with Azure DevOps</b>	18	
5.1 Invite members to Azure devOps	19	
5.2 Azure DevOps Repository linked with Github and VS Code	20	
5.3 Azure DevOps permissions and Access List	22	
<b>6. Deploy Experiments and Model into Azure ML</b>	22	
6.1 Azure ML resource and workspace creation and access control	24	
6.2 Azure ML Data store authentication	26	
6.3 Azure ML Experiments	27	
6.4 Azure ML Codes	28	
6.5 Azure ML Model Registered	29	
6.5 Azure ML Model Deployed	30	
<b>Conclusion</b>	31	
<b>Appendix</b>	32	
List of Figures	32	2
List of Literatures	33	

# Introduction

## The Goal :

- ▶ 1. To build simple Fraud Detection Model using Python.
- ▶ 2. To train, test, check basic statistical measure of the Model.
- ▶ 3. To package the model in API such as Flask or MLFlow.
- ▶ 4. To monitor performance of predictive model using MLFlow.
- ▶ 5. To manage security of data and access control list.
- ▶ 6. To automate, trigger the training once per-month, or if the datasets exceed certain quantity.
- ▶ 7. To describe challenges and constraints during the project.

Data source (Fictional sample dataset):

- ▶ fraud\_detector.csv

Code link:

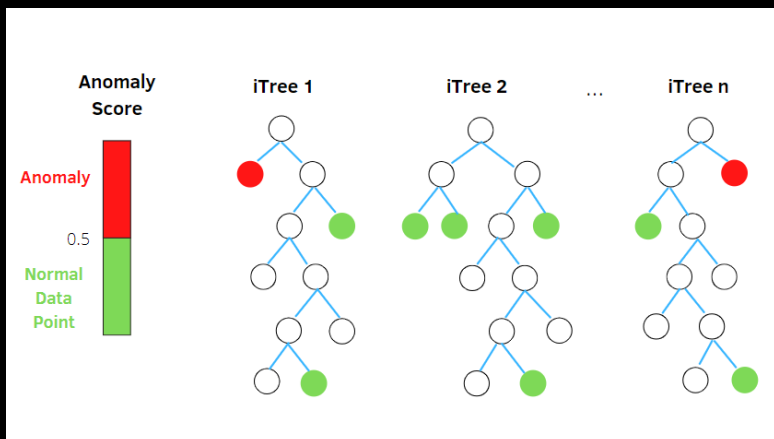
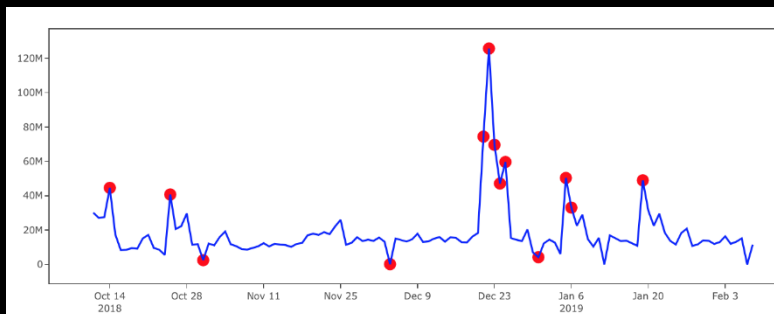
- ▶ [https://github.com/hennypurwadi/mlflow\\_mlop](https://github.com/hennypurwadi/mlflow_mlop)

# Fraud Detection Machine Learning Model

To detect Fraud is to identify anomaly.

Anomaly is set of data that differ significantly from the rest of others.

Anomaly pattern detection is identified automatically by algorithm of Machine Learning.

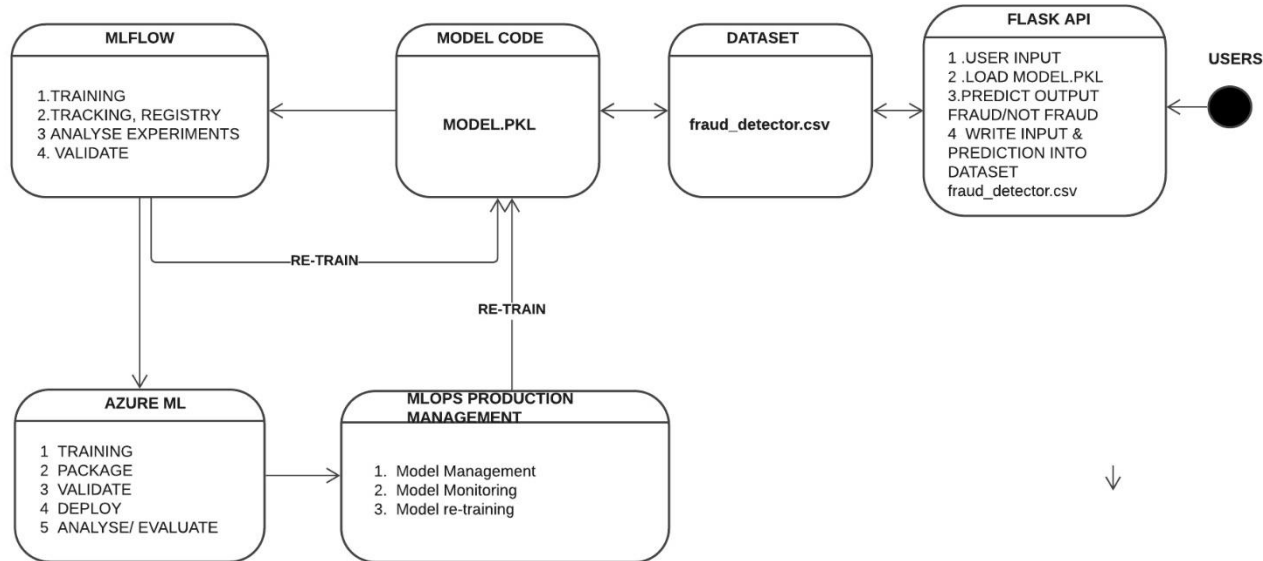


## ISOLATION FOREST

We are identifying anomalies using sklearn's **Isolation Forest**. Isolation forest builds an ensemble of Isolation Trees for dataset.

Anomalies are the point which have shorter average path lengths on the trees, compared to normal ones.

## MACHINE LEARNING MODEL WORKFLOW



Conceptual architecture of Data input, storage processing, prediction model as a service



# FLASK API AND

- ▶ DOCKER

► USER INPUT,  
► FRAUD PREDICTION  
WITH FLASK API

```
(my_env_mlops) (base) PS C:\Users\HENNY\Documents\PYTHON\mlops_proj> python app.py
* Serving Flask app "app" (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 all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://192.168.100.6:5000/ (Press CTRL+C to quit)
```

## Flask API

A1 input:  A2 input:  A1 & A2 input for Predict Fraud or Not

y\_new:

### User Input and model predict

A1 input:  A2 input:  A1 & A2 input for Predict Fraud or Not

y\_new:

A1 input:  A2 input:  A1 & A2 input for Predict Fraud or Not

y\_new:

A1 input:  A2 input:  A1 & A2 input for Predict Fraud or Not

y\_new:

A1 input:  A2 input:  A1 & A2 input for Predict Fraud or Not

y\_new:

## Dataset

	A	B
996	0,0,0	
997	0,0,0	
998	0,0,0	
999	0,0,0	
1000	0,1,1	
1001		
1002		
1003		
1004		

## Write input into Dataset

	A	B
996	0,0,0	
997	0,0,0	
998	0,0,0	
999	0,0,0	
1000	0,1,1	
1001	1,1,1	
1002	0,0,0	
1003	1,0,1	
1004	0,1,1	
1005		

```

%%writefile app.py
import sklearn
import scipy
import numpy as np
import pandas as pd
import csv
from sklearn.metrics import classification_report, accuracy_score
from sklearn.ensemble import IsolationForest
from flask import Flask, flash, request, redirect, url_for
from werkzeug.utils import secure_filename
import json
import joblib

app = Flask(__name__)

@app.route("/status")
def status():
    return "success"

@app.route("/", methods=['GET', 'POST'])
def index():
    A1 = request.args.get("A1", None)
    A2 = request.args.get("A2", None)

    if A1 != None:
        y_new = predict(A1, A2)
    else:
        y_new = ""

    write(A1, A2, y_new)
    return (
        """<form action="" method="get">
            A1 input: <input type="text" name="A1">
            A2 input: <input type="text" name="A2">
            <input type="submit" value="A1 & A2 input for Predict Fraud or Not">
        </form>"""
        + "y_new: "
        + str(y_new)
    )

@app.route("/json", methods=['GET', 'POST'])
def jsonify():
    request_value = request.get_json()
    return request_value

def write(A1, A2, y_new):
    filedf = "fraud_detector.csv"
    # write new data into csv
    with open(filedf, 'a', newline='') as f:
        writer = csv.writer(f)
        writer.writerow([A1, A2, y_new])
    print("file written")

def predict(A1, A2):
    """Predict Fraud or Not Fraud."""
    print("predicting")

    model = joblib.load(open("model.pkl", 'rb'))
    X_new = np.array([A1, A2]).reshape(1, -1)
    y_new = model.predict(X_new)
    y_new[y_new == 1] = 0 # normal
    y_new[y_new == -1] = 1 # possibly fraud
    y_new = (int(y_new))
    return y_new

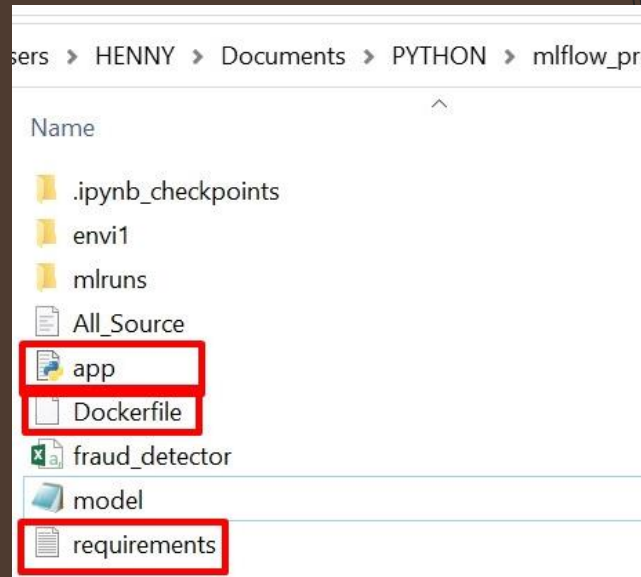
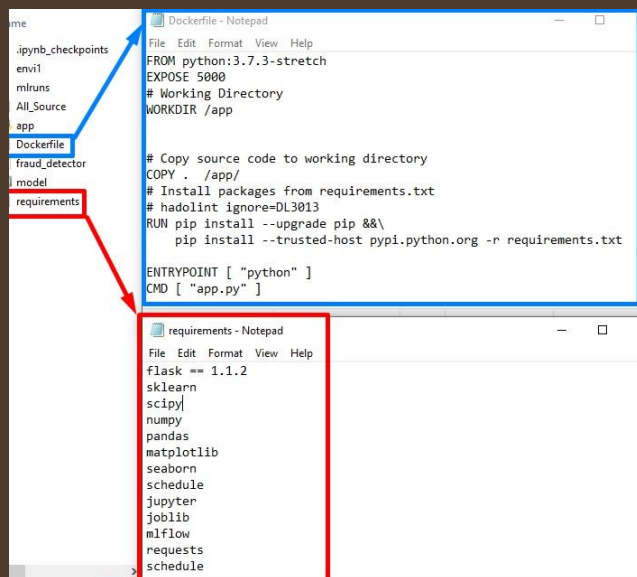
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=int("5000"), debug=True, use_reloader=False)

```

Writing app.py

# Flask codes





```
Anaconda Powershell Prompt (Anaconda3)

(base) PS C:\Users\HENNY\Documents\PYTHON\mlflow_project> python -m venv envi1
(base) PS C:\Users\HENNY\Documents\PYTHON\mlflow_project> .\envi1\Scripts\activate
(envi1) (base) PS C:\Users\HENNY\Documents\PYTHON\mlflow_project> pip install -r requirements.txt
```

## Create Docker Image and Container

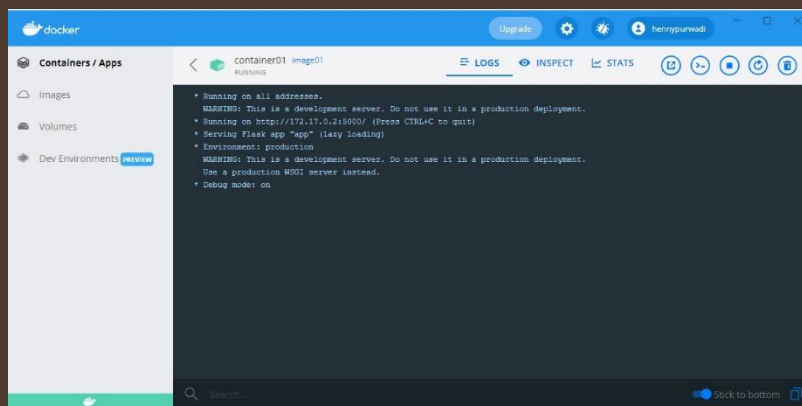
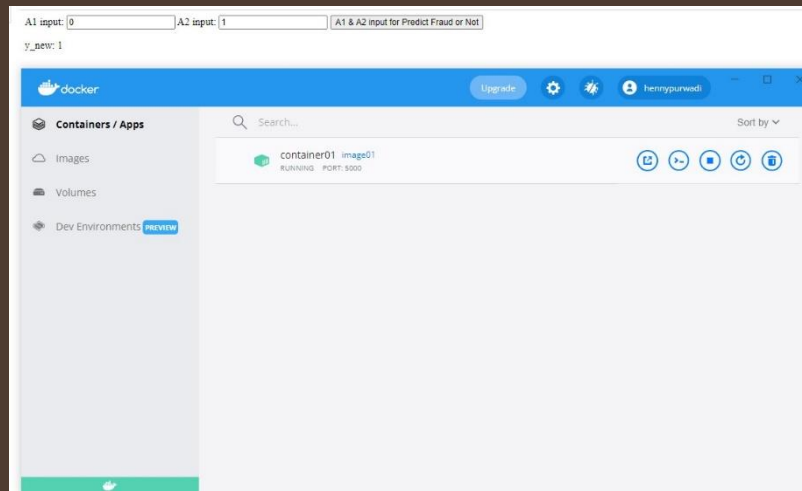
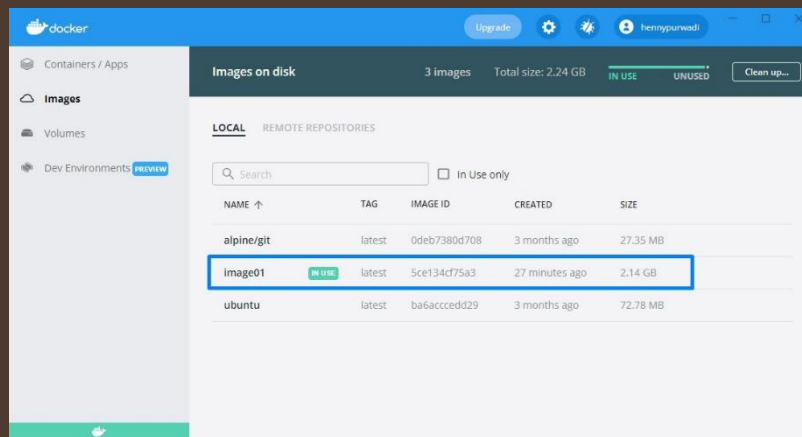
```
Anaconda Powershell Prompt (Anaconda3)

(envi1) (base) PS C:\Users\HENNY\Documents\PYTHON\mlflow_project> docker build -t image01 .
[+] Building 160.5s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 32B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/python:3.7.3-stretch
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load build context
=> => transferring context: 2.30MB
=> [1/4] FROM docker.io/library/python:3.7.3-stretch@sha256:205cafadb7f0c48f2a6d655eb100a9675aa85e6e73958860f4884
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . /app/
=> [4/4] RUN pip install --upgrade pip && pip install --trusted-host pypi.python.org -r requirements.txt
=> exporting to image
=> => exporting layers
=> => writing image sha256:c20c0f1d46742646b15e8e4f3c2eb70a585dc7c882436dbcf99295ca3d20ebe
=> => naming to docker.io/library/image01

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
(envi1) (base) PS C:\Users\HENNY\Documents\PYTHON\mlflow_project> docker run --name container01 -p 5000:5000 image01
* Serving Flask app "app" (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 all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
* Running on http://172.17.0.2:5000/ (Press CTRL+C to quit)
```

# Build image and container from Terminal/ Cmd

# Docker apps



## Challenge and constraint, errors when building Docker image and Docker run:

996	0,0,0	
997	0,0,0	
998	0,0,0	
999	0,0,0	
1000	„	
1001	0,0,0	
1002		

1. Make sure dataset is not opening manually when the code trying to open and write automatically.
2. NaN empty row can cause error in overwrite dataset.
3. Make sure all dependencies in the code are listed in requirements.txt.

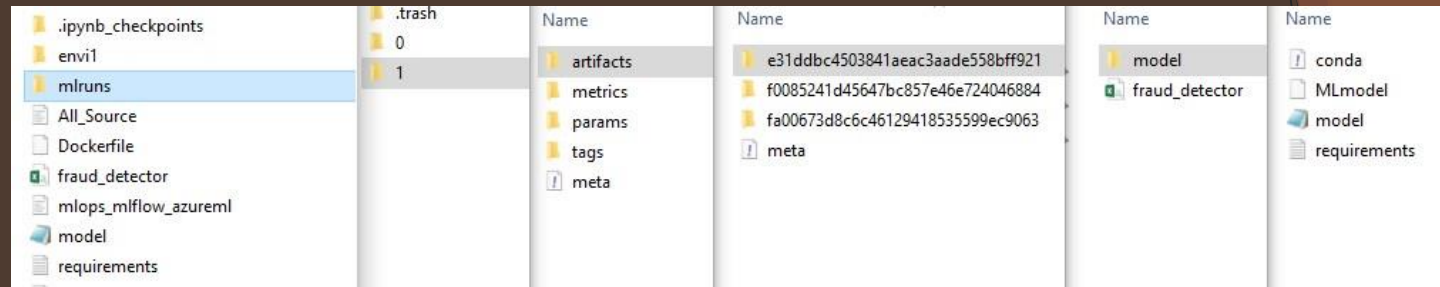
```
app.py x
C: > Users > HENNY > Documents > PYTHON > mlops_proj > app.py > ...
1  import sklearn
2  import scipy
3  import numpy as np
4  import pandas as pd
5  import csv
6  from sklearn.metrics import classification_report, accuracy_score
7  from sklearn.ensemble import IsolationForest
8  from flask import Flask, flash, request, redirect, url_for
9  from werkzeug.utils import secure_filename
10 import json
11 import pickle
12
```

```
requirements - Notepad
File Edit Format View Help
flask == 1.1.2
sklearn
scipy
numpy
pandas
matplotlib
seaborn
schedule
jupyter
```



# MLFlow tracking, monitoring, automate

- ▶ triggered training



```

Anaconda Powershell Prompt (Anaconda3)

(base) PS C:\Users\HENNY\Documents\PYTHON\mlflow_project> .\envi1\Scripts\activate
(envi1) (base) PS C:\Users\HENNY\Documents\PYTHON\mlflow_project> mlflow ui
INFO:waitress:erving on http://127.0.0.1:5000
  
```

mlflow

Experiments Models

Experiments

Search Experiments

Default

evaluate\_metric

evaluate\_metric

Experiment ID: 1

Description Edit

Refresh

Compare

Delete

Download CSV

Start Time

All time

Columns Only show differences

metrics.rose < 1 and params.model = "tree"

Search Filter Clear

Showing 4 matching runs

								Metrics	Parameters	Tags
	Start Time	Duration	Run Name	User	Source	Version	Models	accuracy score quality	behaviour bootstrap contamination estimator_class estimator_name deployed	
	4 minutes ago	8.7s	-	HENNY	C:\Program	-	sklearn	- -	deprecated False auto sklearnense...	IsolationFor...
	18 minutes ago	4.5s	-	HENNY	C:\Program	-	sklearn	0.977 -	- - - -	15-01-2022 ...
	18 minutes ago	6.8s	-	HENNY	C:\Program	-	sklearn	0.977 -	- - - -	-
	18 minutes ago	31ms	-	HENNY	C:\Program	-	-	- 4	- - - -	-

# Model tracking with mlflow



## MLFlow

```
import mlflow
import numpy as np
import pandas as pd
import joblib
import csv
import json
import os
import sklearn
import mlflow.sklearn
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, accuracy_score
from sklearn.ensemble import IsolationForest
from datetime import datetime
from mlflow.tracking import MlflowClient
from flask import Flask, flash, request, redirect, url_for
from werkzeug.utils import secure_filename

mlflow.tracking.get_tracking_uri()
exp_name = "evaluate_metric"
mlflow.set_experiment(exp_name)

filedf = "fraud_detector.csv"
df = pd.read_csv(filedf)

#training and testing dataset
X = df.drop("Category", axis=1)
y = df.Category
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

#load model
model = joblib.load(open("model.pkl", "rb"))
#model= IsolationForest(n_estimators=100, max_samples=len(X_train), random_state=0, verbose=0)
#model.fit(X_train, y_train)
ypred= model.predict(X_test)

ypred[ypred == 1] = 0 #normal
ypred[ypred == -1] = 1 #possibly fraud

def eval_metrics(actual, pred):
    # compute relevant metrics
    acc_score = accuracy_score(y_test, ypred)
    return acc_score

def load_data(filedf):
    df = pd.read_csv(filedf)
    X = df.drop("Category", axis=1)
    y = df.Category
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
    return X_train, y_train, X_test, y_test
```

```
def main(n_estimators=40, max_samples=len(X_train)):
    # train a model with given parameters
    np.random.seed(42)

    # Read csv file
    filedf = "fraud_detector.csv"
    train_X, train_Y, test_X, test_Y = load_data(filedf)

    # Useful for multiple runs
    with mlflow.start_run():
        # Load model
        model = joblib.load(open("model.pkl", "rb"))

        ypred[ypred == 1] = 0 #normal
        ypred[ypred == -1] = 1 #possibly fraud

        #Freeze Model with joblib
        filename_pkl = "model.pkl"
        joblib.dump(model, open(filename_pkl, "wb"))

        # Evaluate Metrics
        predicted_qualities = model.predict(X_test)
        (acc_score) = eval_metrics(y_test, predicted_qualities)

        # Print out metrics
        print("evaluate_metric (n_estimators=%f, max_samples=%f):" % (n_estimators, max_samples))
        print(" ACCURACY SCORE: %s" % acc_score)

        # Log parameter, metrics, and model to MLflow
        mlflow.log_param(key="n_estimators", value=n_estimators)
        mlflow.log_param(key="max_samples", value=max_samples)
        mlflow.log_metrics({"accuracy score": acc_score})
        mlflow.log_artifact(filedf)
        print("Save to: {}".format(mlflow.get_artifact_uri()))

        mlflow.sklearn.log_model(model, "model")

    with mlflow.start_run():
        for epoch in range(0, 3):
            mlflow.log_metric(key="quality", value=2*epoch, step=epoch)

main(100,600)
print('\n')
main(110,630)

2022/01/15 16:35:37 INFO mlflow.tracking.fluent: Experiment with name 'evaluate_metric' does not exist. Creating a new experiment.

evaluate_metric (n_estimators=100.000000, max_samples=600.000000):
ACCURACY SCORE: 0.9766666666666667
Save to: file:///C:/Users/HENNY/Documents/PYTHON/mlflow_project/mlruns/1/f0885241d45647bc897e46e724046884/artifacts

evaluate_metric (n_estimators=110.000000, max_samples=630.000000):
ACCURACY SCORE: 0.9766666666666667
Save to: file:///C:/Users/HENNY/Documents/PYTHON/mlflow_project/mlruns/1/e31ddbc4503841aeac3ade558bf921/artifacts
```

```
from datetime import datetime
from mlflow.tracking import MlflowClient

client = MlflowClient()
experiments = client.list_experiments() # returns a list of mlflow.entities.Experiment
print(experiments)

[Experiment: artifact_location='file:///C:/Users/HENNY/Documents/PYTHON/mlflow_project/mlruns/0', experiment_id='0', lifecycle_stage='active', name='default', tags={}, <Experiment: artifact_location='file:///C:/Users/HENNY/Documents/PYTHON/mlflow_project/mlruns/1', experiment_id='1', lifecycle_stage='active', name='evaluate_metric', tags={}]]

# get the run
_run = client.get_run(run_id="e31ddbc4503841aeac3ade558bf921")
print(_run)

<Run: data=<RunData: metrics={'accuracy score': 0.9766666666666667}, params={'max_samples': '630', 'n_estimators': '110'}, tags={'mlflow.log_model.history': [{"run_id": "e31ddbc4503841aeac3ade558bf921", 'artifact_path': "model", 'utc_time_created': '2022-01-15 09:35:45.018400', 'flavors': {'python_function': ('model_path': "model.pkl", 'loader_module': "mlflow.sklearn", 'python_version': "3.8.8", "env": "conda.yaml", 'sklearn': {'pickled_model': "model.pkl", 'sklearn_version': "0.23.2", 'serialization_format': "cloudpickle"}}}], 'mlflow.source.name': 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\ipykernel_launcher.py', 'mlflow.source.type': 'LOCAL', 'mlflow.user': 'HENNY'}, info=(RunInfo: artifact_uri='file:///C:/Users/HENNY/Documents/PYTHON/mlflow_project/mlruns/1/e31ddbc4503841aeac3ade558bf921/artifacts', end_time=164220949412, experiment_id='1', lifecycle_stage='active', run_id='e31ddbc4503841aeac3ade558bf921', run_uid='e31ddbc4503841aeac3ade558bf921', start_time=164220944917, status='FINISHED', user_id='HENNY')>>

# add a tag to the run
dt = datetime.now().strftime("%d-%m-%Y (%H:%M:%S.%f)")
client.set_tag(_run.info.run_id, "deployed", dt)
print(client.set_tag)
print('\n')
print(dt)

<bound method MlflowClient.set_tag of <mlflow.tracking.client.MlflowClient object at 0x00000208AC227CAB>>
```

# Mlflow codes

# Automate scheduled/ triggered monthly training

## Automate scheduled training

```
#!/usr/bin/env python
# %%writefile trigger_train.py

def trigger_train():

    import sklearn
    from sklearn.metrics import confusion_matrix
    import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import classification_report, accuracy_score
    from sklearn.ensemble import IsolationForest
    import joblib
    import datetime
    import requests
    import warnings
    warnings.filterwarnings('ignore')

    filedf = 'fraud_detector.csv'
    df = pd.read_csv(filedf)
    X = df.drop("Category", axis=1)
    y = df.Category
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=40)

    model = IsolationForest(n_estimators=100, max_samples=len(X_train), random_state=0, verbose=0)
    model.fit(X_train, y_train)
    #model = joblib.load(open("model.pkl", 'rb'))

    ypred = model.predict(X_test)
    ypred[ypred == 1] = 0 #normal
    ypred[ypred == -1] = 1 #possibly fraud

    #Freeze Model with joblib
    filename_pkl = 'model.pkl'
    joblib.dump(model, open(filename_pkl, 'wb'))
    print("model.pkl saved")

#Automate scheduled training
#mlflow.autolog({"run_id": "749eb2eaf2a84e1992110481c707a7a9"})
trigger_train()

import schedule
schedule.every(720).hours.do(trigger_train)

model.pkl saved

Every 720 hours do trigger_train() (last run: [never], next run: 2022-02-15 12:55:03)
```

Automate mlflow Run\_id  
scheduled/ triggered  
monthly training

```
#Automate scheduled training mlflow Run_id
mlflow.autolog({"run_id": "e31ddbc4503841aeac3aade558bff921"})
trigger_train()

import schedule
schedule.every(720).hours.do(trigger_train)

2022/01/16 13:31:49 INFO mlflow.tracking.fluent: Autologging successfully enabled for sklearn.
2022/01/16 13:31:49 INFO mlflow.utils.autologging_utils: Created MLflow autologging run with ID '7613e491506842e2aca36b471acaeb48', which will track hyperparameters, performance metrics, model artifacts, and lineage information for the current sklearn workflow

model.pkl saved

Every 720 hours do trigger_train() (last run: [never], next run: 2022-02-15 13:31:54)
```



# Automate training when quantity of datasets exceed certain numbers

```
def exceed_train():

    import sklearn
    from sklearn.metrics import confusion_matrix
    import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import classification_report, accuracy_score
    from sklearn.ensemble import IsolationForest
    import joblib
    import datetime
    import requests
    import warnings
    warnings.filterwarnings('ignore')

    filedf = 'fraud_detector.csv'
    df = pd.read_csv(filedf)
    len(df)

    limit = 2000
    if len(df) > limit:
        trigger_train()
        print('training done')

#Automate scheduled training
#mlflow.autolog({"run_id": "749eb2eaf2a84e1992110481c7a7a9"})
trigger_train()
exceed_train()

import schedule
schedule.every(240).hours.do(exceed_train)
schedule.every(720).hours.do(trigger_train)

model.pkl saved

Every 720 hours do trigger_train() (last run: [never], next run: 2022-02-15 21:13:16)
```



# MLOps with

- ▶ Azure DevOps

henrypuriadi / mlop / Settings / Teams / mlop Team

### Project Settings

mlop

General

- Overview
- Teams
- Permissions
- Notifications
- Service hooks
- Dashboards

Boards

- Project configuration
- Team configuration
- GitHub connections

Pipelines

- Agent pools
- Parallel jobs
- Settings
- Test management
- Release retention
- Service connections
- XAML build services

Repos

- Repositories

Artifacts

- Storage

Test

- Retention

### mlop Team

The default project team.  
Relevant links: Notifications | Dashboards | Iterations and Area Paths

Members Settings

Total 1

<input type="checkbox"/>	Name	Type	Username or scope
<input type="checkbox"/>	Henry Purwadi henry.puriadi@lubb.de	add user	henry.puriadi@lubb.de

#### Invite members to mlop Team

Search and add users and/or groups to your team

Add users and/or groups

- Abels, Nadine  
nadine.abels@lu-study.org
- Ackermann, Nadine  
nadine.ackermann@lubb-fernstudium.de
- Adrian, Nadine  
nadine.adrian@lu-study.org
- Apreiter, Nadine  
nadine.agreiter@lubb-fernstudium.de
- Agulgan, Nadine-Goussi  
nadine-goussi.agulgan@lubb-fernstudium.de
- Akmeztanli, Nadine  
nadine.akmeztanli@lubb-fernstudium.de

henrypuriadi / mlop / Settings / Teams / mlop Team

Search

### Project Settings

mlop

General

- Overview
- Teams
- Permissions
- Notifications
- Service hooks
- Dashboards

### mlop Team

The default project team.  
Relevant links: Notifications | Dashboards | Iterations and Area Paths

Members Settings

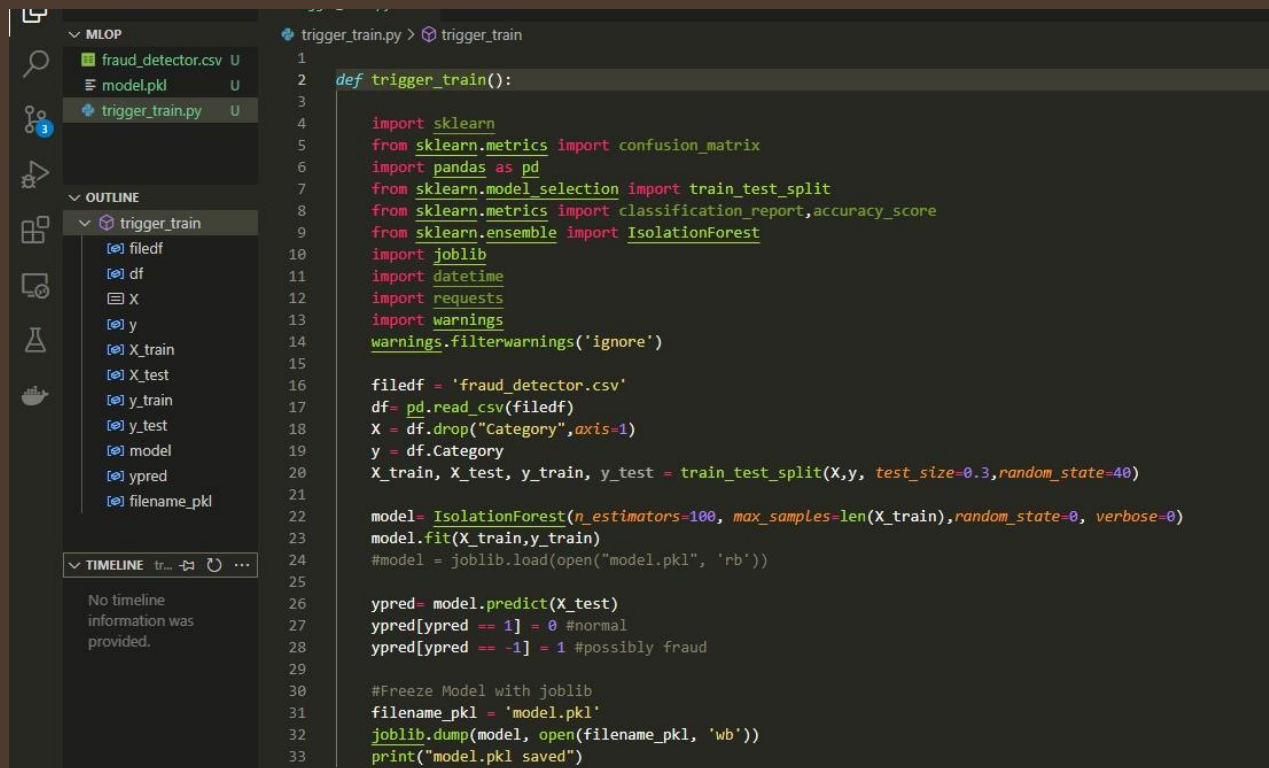
Total 1

<input type="checkbox"/>	Name	Type	Username or scope
<input type="checkbox"/>	Henry Purwadi henry.puriadi@lubb.de	add user	henry.puriadi@lubb.de

Search users and groups

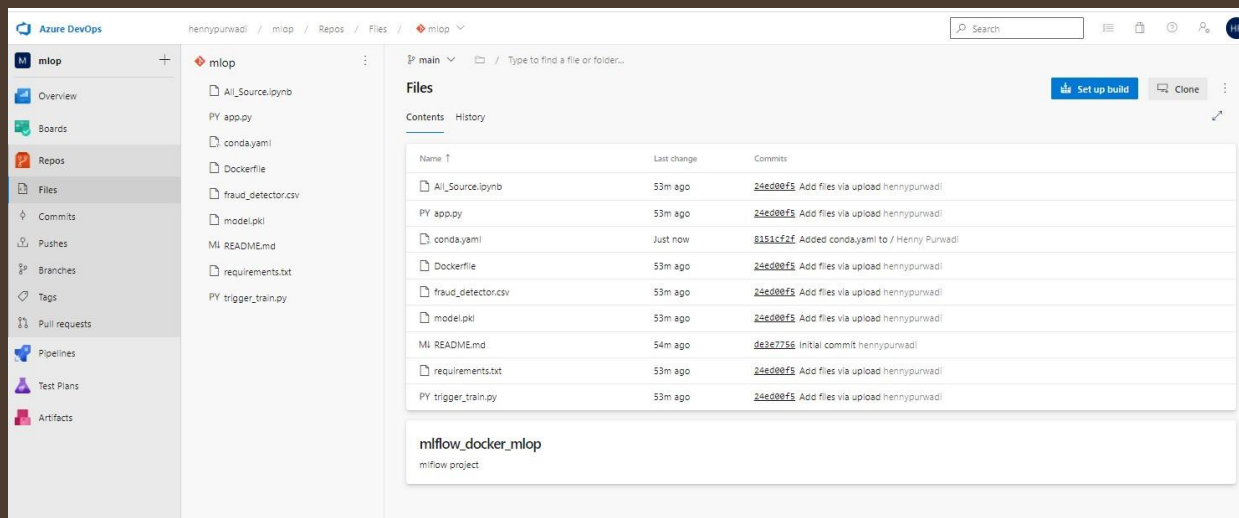
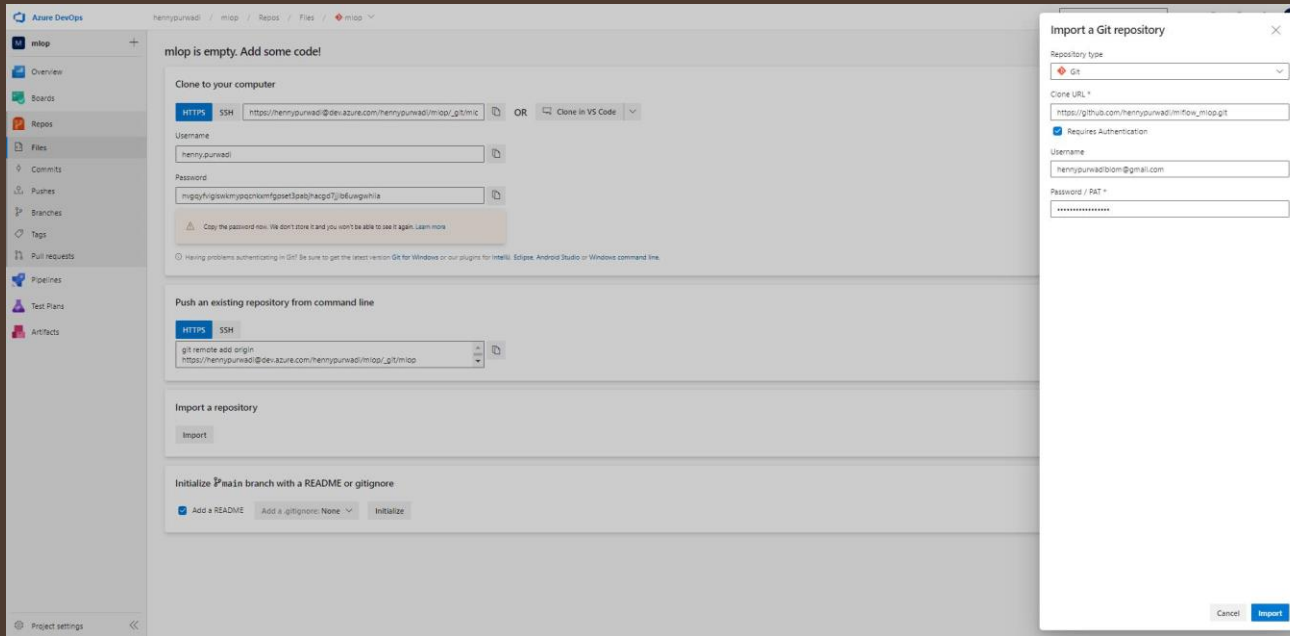
Expanded Members

# Invite members to Project in Azure DevOps



```
1 trigger_train.py > trigger_train
2 def trigger_train():
3
4     import sklearn
5     from sklearn.metrics import confusion_matrix
6     import pandas as pd
7     from sklearn.model_selection import train_test_split
8     from sklearn.metrics import classification_report, accuracy_score
9     from sklearn.ensemble import IsolationForest
10    import joblib
11    import datetime
12    import requests
13    import warnings
14    warnings.filterwarnings('ignore')
15
16    filedf = 'fraud_detector.csv'
17    df = pd.read_csv(filedf)
18    X = df.drop("Category", axis=1)
19    y = df.Category
20    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=40)
21
22    model = IsolationForest(n_estimators=100, max_samples=len(X_train), random_state=0, verbose=0)
23    model.fit(X_train, y_train)
24    #model = joblib.load(open("model.pkl", 'rb'))
25
26    ypred = model.predict(X_test)
27    ypred[ypred == 1] = 0 #normal
28    ypred[ypred == -1] = 1 #possibly fraud
29
30    #Freeze Model with joblib
31    filename_pkl = 'model.pkl'
32    joblib.dump(model, open(filename_pkl, 'wb'))
33    print("model.pkl saved")
```

# Azure DevOps linked with VS Code



Azure DevOps linked with Github repository

henrypunwadi / mlop / Settings / Repositories

Project Settings  
mlop

General

- Overview
- Teams
- Permissions
- Notifications
- Service hooks
- Dashboards

Boards

- Project configuration
- Team configuration
- GitHub connections

Pipelines

- Agent pools
- Parallel jobs
- Settings
- Test management
- Release retention
- Service connections
- XAML build services

Repos

- Repositories
- Artifacts
- Storage
- Test
- Retention

All Repositories

Filter by keywords

mlop

Settings Policies Security

User permissions

Download detailed report

Inheritance

Search for users or groups

Azure DevOps Groups

- Project Collection Administrators
- Project Collection Build Service Accounts
- Project Collection Service Accounts
- Build Administrators
- Contributors
- Project Administrators
- Readers
- Users
- Mlop Build Service (Henrypunwadi)

[henrypunwadi]Project Collection Administrators

Bypass policies when completing pull requests

Bypass policies when pushing

Contribute

Contribute to pull requests

Create branch

Create tag

Delete or disable repository

Edit policies

Force push (rewrite history, delete branches and tags)

Manage notes

Manage permissions

Read

Remove others' locks

Rename repository

Not set

Not set

Allow (inherited)

Allow (inherited)

Allow (inherited)

Allow (inherited)

Allow (inherited)

Allow (inherited)

Allow (inherited)

Allow (inherited)

Allow (inherited)

Allow (inherited)

Allow (inherited)

Allow (inherited)

Allow (inherited)

Allow (inherited)

Pipeline permissions

No permitted pipelines

Only YAML pipelines defined in this repository are allowed access. YAML pipelines from other projects are not shown in this list. All classic pipelines can use this repository. [Learn more](#)

henrypunwadi / mlop / Settings / Permissions

Project Settings  
mlop

General

- Overview
- Teams
- Permissions
- Notifications
- Service hooks
- Dashboards

Boards

- Project configuration
- Team configuration
- GitHub connections

Pipelines

- Agent pools
- Parallel jobs

Permissions

Groups Users

Search groups

New Group

Total 6

Name	Description	Type	Members
Build Administrators	Members of this group can create, modify and delete build definitions and manage queued and completed builds.	Group	0
Contributors	Members of this group can add, modify, and delete items within the team project.	Group	1
Project Administrators	Members of this group can perform all operations in the team project.	Group	1
Project Valid Users	Members of this group have access to the team project.	Group	5
Readers	Members of this group have access to the team project.	Group	0
mlop Team	The default project team.	Team	1

# Azure DevOps Permissions



# Deploy Model

- ▶ into Azure ML

```
AZ_SUBSCRIPTION_ID='b667c281-58f0-47a7-899b-63ea6cd8b7e8' #azure-subscription-id
ws = Workspace.create(name='ws1',subscription_id=AZ_SUBSCRIPTION_ID,
                      resource_group='res1',create_resource_group=True,location='southeastasia')
```

Deploying KeyVault with name ws1keyvault5baf6f8ab8d64.  
 Deploying StorageAccount with name ws1storage4dbf6fc102d446.  
 Deploying AppInsights with name ws1insightsde9d189167484.  
 Deployed AppInsights with name ws1insightsde9d189167484. Took 4.48 seconds.  
 Deploying Workspace with name ws1.  
 Deployed KeyVault with name ws1keyvault5baf6f8ab8d64. Took 22.98 seconds.  
 Deployed StorageAccount with name ws1storage4dbf6fc102d446. Took 29.27 seconds.  
 Deployed Workspace with name ws1. Took 38.88 seconds.

Name	Name
.azureml	ws_config
apihub_checkpoints	
_pycache_	
env1	
mlup	
mlruns	
All_Source	
app	
Dockerfile	
fraud_detector	
model	
myenv	
README	
requirements	
score	
trigger_train	

The screenshot displays the Microsoft Azure portal interface. At the top, the header shows 'Microsoft Azure' and a search bar. The main navigation pane on the left lists various services and settings. The central pane is titled 'res1 | Access control (IAM)' and shows the 'Check access' tab. It includes a search bar for finding users, groups, or service principals. The right-hand side of the page contains four panels: 'Grant access to this resource' with an 'Add role assignment' button, 'View access to this resource' with a 'View' button, 'View deny assignments' with a 'View' button, and 'Create a custom role' with an 'Add' button.

# AzureML resource and workspace creation and access control



Microsoft Azure Search resources, services, and docs (G+)

Home > res1 >

## Add role assignment

Got feedback?

Role **Members** Review + assign

**Assign access to**

☒ User, group, or service principal  
☐ Managed identity

**Members** + Select members

Name	Object ID	Type
No members selected		

**Description**

Optional

Select members

Select

Nadine

Bohler, Nadine  
nadine.bohler@iubh-fernstudium.de

Bohne, Nadine  
nadine.bohne@iubh-fernstudium.de

Bohne, Nadine  
nadine.bohne@iubh.de

Bollmann, Nadine  
nadine.bollmann@iubh-fernstudium.de

Boor, Nadine  
nadine.boor@iu-study.org

Borchardt, Nadine  
nadine.borchardt@iubh-fernstudium.de

Selected members:  
No members selected. Search for and add one or more members you want to assign to the role for this resource.

[Learn more about RBAC](#)

Microsoft Azure Search resources, services, and docs (G+)

Home > res1

## res1 | Access control (IAM)

Resource group

Search (Ctrl+J)

+ Add Download role assignments Edit columns Refresh Remove Got feedback?

Check access **Role assignments** Roles Deny assignments Classic administrators

Number of role assignments for this subscription

23 2000

Search by name or email

Type: All Role: All Scope: All scopes Group by: Role

5 items (4 Users, 1 Managed Identities)

Name	Type	Role	Scope	Condition
Contributor				
ws1	App	Contributor	This resource	None
Owner				
Purwadi, Henry henry.purwadi@iubh.de	User	Owner	Subscription (inherited)	None
Security Admin				
Simon Stitzl a.stitzl@iu.org	User	Security Admin	Management group (inherited)	None
User Access Administrator				
Marsden, Chris (Admin) a.marsden@iu.org	User	User Access Administrator	Root (inherited)	None
Köppel, Reinhard reinhard.koeppel@iu.org	User	User Access Administrator	Root (inherited)	None

Overview  
Activity log  
Access control (IAM)  
Tags  
Resource visualizer  
Events

Settings

Deployments  
Security  
Policies  
Properties  
Locks

Cost Management

Cost analysis  
Cost alerts (preview)  
Budgets  
Advisor recommendations

Monitoring

Insights (preview)  
Alerts  
Metrics

Microsoft Azure Machine Learning Studio

Home > Datastores > azureml

azureml

Overview Browse (preview)

Create dataset Refresh Update authentication Set as default datastore Unregister

**General**

Datastore name  
azureml

Datastore type  
Azure Blob Storage

Created by  
Purwadi

Subscription ID  
---

Resource group name  
--

Protocol  
https

Endpoint  
core.windows.net

Account name  
ws1storage4dbefc102d446

**Update datastore credentials**

Subscription ID  
Select or search by name

Resource group name of the storage resource  
Select or search by name

Save credentials with the datastore for data access ⓘ

Authentication type \* ⓘ  
Account key

Account key \* ⓘ  
.....

Use workspace managed identity for data preview and profiling in Azure Machine Learning studio ⓘ

ⓘ Note: Azure Machine Learning service does not validate whether the underlying data source exists... ⓘ

Microsoft Azure Machine Learning Studio

Home > Datastores > azureml

azureml

Overview Browse (preview)

Create dataset Refresh Update authentication Set as default datastore Unregister

**General**

Datastore name  
azureml

Datastore type  
Azure Blob Storage

Created by  
Purwadi

Subscription ID  
---

Resource group name  
---

Protocol  
https

Endpoint  
core.windows.net

Account name  
ws1storage4dbefc102d446

Blob container  
azureml

Data URL  
https://ws1storage4dbefc102d446.blob.core.windows.net/azureml

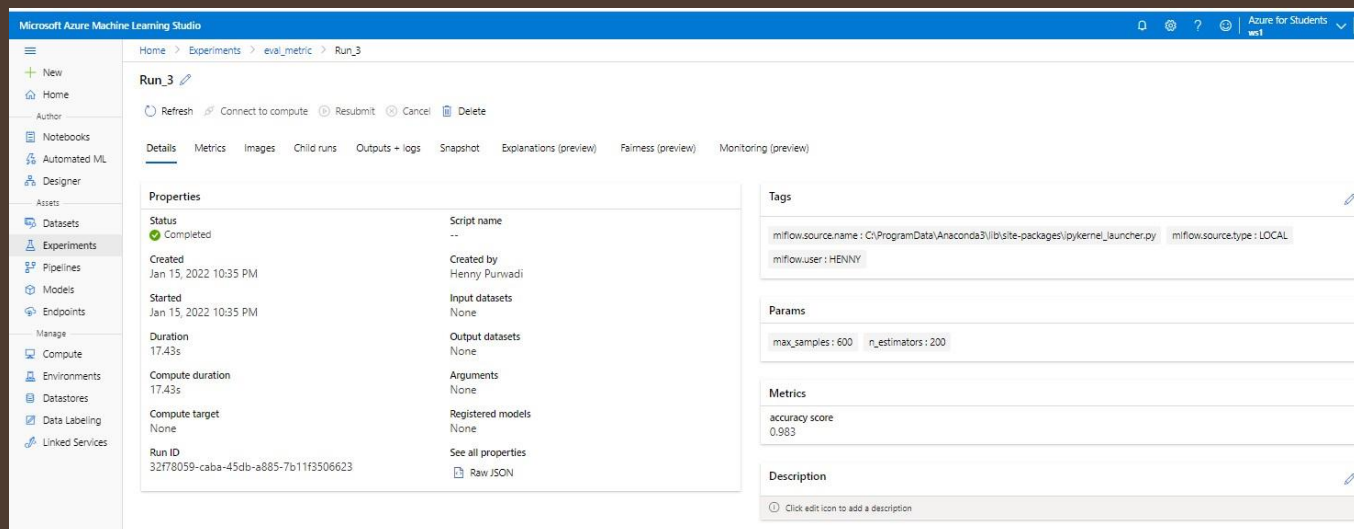
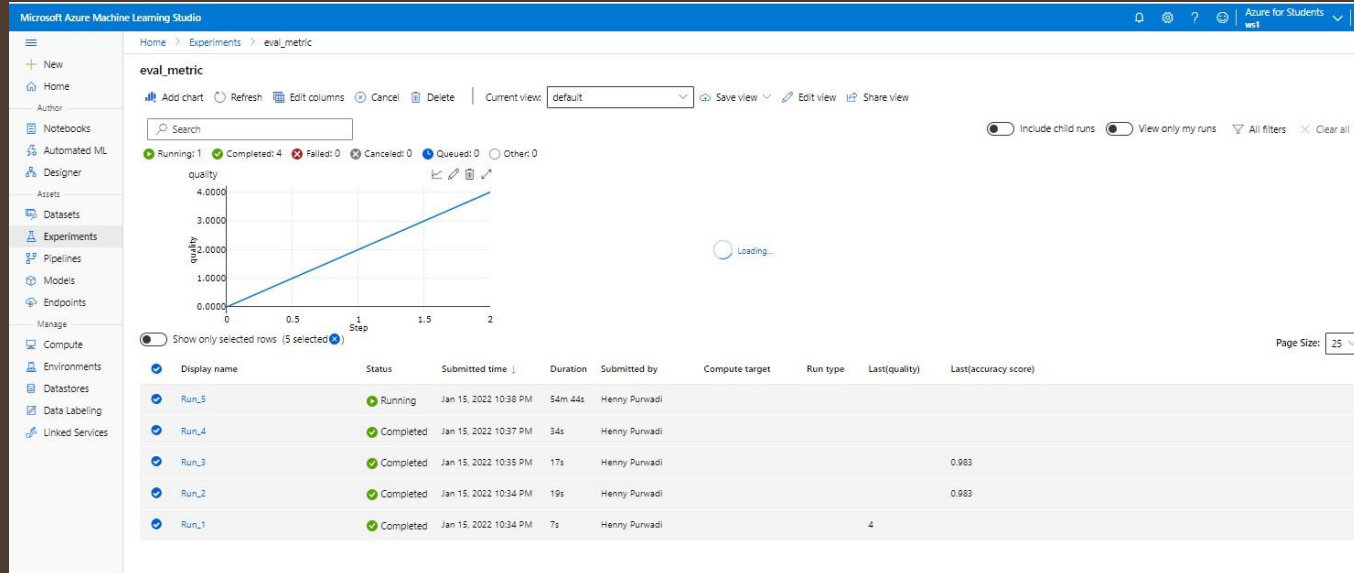
Created on  
Jan 15, 2022 11:14 PM

**Authentication**

Allowed workspace managed identity access  
No

Authentication type  
Account key

# Datastore authentication



# AzureML Experiments

```

train(200, 600)

evaluate_metric (n_estimators=200.000000, max_samples=600.000000):
  ACCURACY SCORE: 0.9033333333333333
  Save to: azureml://experiments/eval_metric/runs/32f78059-caba-45db-a885-7b11f3506623/artifacts

mlflow.end_run() #End previous Run
mlflow_run = mlflow.start_run() #start new Run

client = MlflowClient()
finished_mlflow_run = MlflowClient().get_run(mlflow_run.info.run_id)

exp = Experiment(workspace=ws, name=exp_name)
list_experiments = exp.list(ws)

list_runs = exp.get_runs()
for run in list_runs:
    print(run.id)

37b89e16-4cd3-435d-b823-d5eecd040820
bdb569bd-d092-481d-a34c-8d1b39eb6c9f
32f78059-caba-45db-a885-7b11f3506623
04709530-c2be-473d-9d95-0acd270c6cd0
8453c18a-f1fa-460c-880d-8ce1ae7ce07b

metrics = finished_mlflow_run.data.metrics
tags = finished_mlflow_run.data.tags
params = finished_mlflow_run.data.params
print(tags)

{'mlflow.user': 'HEWNY', 'mlflow.source.name': 'C:\\ProgramData\\Anaconda3\\lib\\site-packages\\ipykernel_launcher.py', 'mlflow.source.type': 'LOCAL', 'mlflow.rootRunId': '37b89e16-4cd3-435d-b823-d5eecd040820'}

from azureml.core.model import Model
model = Model.register(model_path = "model.pkl",
                        model_name = "fraud_detect",
                        tags = {"key": "1"},
                        description = "fraud Prediction",
                        workspace = ws,)

Registering model fraud_detect

#import model
model = Model(workspace=ws, name="fraud_detect")
model.download(target_dir=os.getcwd(), exist_ok=True)
#model.delete()

'C:\\Users\\HEWNY\\Documents\\PYTHON\\mlflow_project\\model.pkl'

print(model.name, model.id, model.version, sep='\\t')
azureml.core.compute.ComputeTarget
azureml.core.runconfig.RunConfiguration

fraud_detect      fraud_detect:1 1

azureml.core.runconfig.RunConfiguration

```

## AzureML codes

```

%%time
image_config = ContainerImage.image_configuration(execution_script="score.py", runtime="python")

Wall time: 1 ms

```

### Inference Config

```

env=ws.environments['AzureML-mlflow-ubuntu18.04-py37-cpu-inference']

dummy_inference_config = InferenceConfig(environment=env, source_directory='.', entry_script="score.py")
print(dummy_inference_config)
print(InferenceConfig)
print(image_config)

Warning, custom base image or base dockerfile detected without a specified 'inferencing_stack_version'. Please set environment.
inferencing_stack_version='latest'

InferenceConfig(entry_script=score.py, runtime=None, conda_file=None, extra_docker_file_steps=None, source_directory=C:\\Users\\HENNY\\Documents\\PYTHON\\mlflow_project, enable_gpu=None, base_image=None, base_image_registry=azureml.core.container_registry.ContainerRegistry object at 0x000001c23506c280)
<class 'azureml.core.model.InferenceConfig'>
<azureml.core.image.container.ContainerImageConfig object at 0x000001c235147490>

from azureml.core.webservice import AciWebservice, Webservice
# Create a deployment config
aci_config = AciWebservice.deploy_configuration(cpu_cores=1, memory_gb=1, auth_enabled=True)
aci_config

<azureml.core.webservice.aci.AciServiceDeploymentConfiguration at 0x1c235ad2580>

src = ScriptRunConfig(source_directory='.', script='score.py', environment=env)
print(src)

<azureml.core.script_run_config.ScriptRunConfig object at 0x000001c235ad2550>

```

```

from azureml.core import Environment
from azureml.core.environment import Environment
from azureml.core.conda_dependencies import CondaDependencies
from azureml.core.model import InferenceConfig

myenv = CondaDependencies()
myenv.add_conda_package('')

myenv.add_pip_package('pip==21.0.1')
myenv.add_pip_package('azureml.core')
myenv.add_conda_package('scikit-learn')
myenv.add_conda_package('mlflow')
myenv.add_conda_package('numpy')
myenv.add_conda_package('pandas')
myenv.add_conda_package('joblib')
myenv.add_conda_package('python==3.6.2')
myenv.add_conda_package('flask==2.0.1')

with open("myenv.yaml", "w") as f:
    f.write(myenv.serialize_to_string())
with open("myenv.yaml", "r") as f:
    print(f.read())

myenv = Environment.from_conda_specification(name='myenv', file_path='myenv.yaml')

# Conda environment specification. The dependencies defined in this file will
# be automatically provisioned for runs with userManagedDependencies=False.

# Details about the Conda environment file format:
# https://conda.io/docs/user-guide/tasks/manage-environments.html#create-env-file-manually

name: project_environment
dependencies:
  # The python interpreter version.

  # Currently Azure ML only supports 3.5.2 and later.

- python=3.6.2

- pip:
  # Required packages for AzureML execution, history, and data preparation.

  - azureml-defaults
  - pip==21.0.1
  - azureml.core
  - ..
  - scikit-learn
  - mlflow
  - numpy
  - pandas
  - joblib
  - flask==2.0.1
channel(s):
  - anaconda
  - conda-forge

```

Microsoft Azure Machine Learning Studio

Home > Experiments > eval\_metric > Run\_3

### Run\_3

Refresh Connect to compute Resubmit Cancel Delete Download all Enable log streaming Word wrap

Details Metrics Images Child runs **Outputs + logs** Snapshot Explanations (preview) Fairness (preview) Monitoring (preview)

model

- conda.yaml
- MLmodel
- model.pkl
- requirements.txt
- fraud\_detector.csv

Microsoft Azure Machine Learning Studio

Home > Models > fraud\_detect:1

### fraud\_detect:1

Details Versions Artifacts Endpoints Explanations (preview) Fairness (preview) Datasets

Refresh Edit columns Reset view

Search

Created on Created by Tags All filters Clear all

Showing 1-1 of 1 models

Page size: 25

Name	Version	Experiment	Run ID	Created on	Tags	Properties	Created by
fraud_detect	1			Jan 15, 2022 10:42 PM	key:1		Henny Purwadi

Microsoft Azure Machine Learning Studio

Home > Models > fraud\_detect:1

### fraud\_detect:1

Details Versions Artifacts Endpoints Explanations (preview) Fairness (preview) Datasets

Refresh Deploy Download all

Attributes

Version  
1

ID  
fraud\_detect:1

Date registered  
1/15/2022, 10:42:58 PM

Format  
CUSTOM

Experiment name  
--

Run ID  
--

Created by  
Henny Purwadi

Tags

key:1

Properties

No properties

Description

fraud Prediction

AzureML Model registered

Microsoft Azure

Search resources, services, and docs (G+)

Home > res1

## res1 | Deployments

Resource group

Search (Ctrl+/)

Refresh Cancel Redeploy Delete View template

Filter by deployment name or resources in the deployment...

Deployment name	Status	Last modified	Duration	Related events
<input type="checkbox"/> Microsoft.MachineLearningServices_30522	✓ Succeeded	1/16/2022, 9:52:55 AM	1 minute 40 seconds	<a href="#">Related events</a>
<input type="checkbox"/> Microsoft.MachineLearningServices_96381	✓ Succeeded	1/16/2022, 12:12:12 AM	1 minute 7 seconds	<a href="#">Related events</a>
<input type="checkbox"/> Microsoft.MachineLearningServices_51232	✓ Succeeded	1/15/2022, 10:29:06 PM	1 minute 9 seconds	<a href="#">Related events</a>

Overview  
Activity log  
Access control (IAM)  
Tags  
Resource visualizer  
Events

New  
Home  
Author  
Notebooks  
Automated ML  
Designer  
Assets  
Datasets  
Experiments  
Pipelines  
Models  
Endpoints  
Manage  
Compute  
Environments  
Datastores  
Data Labeling  
Linked Services

### endpoint2

Details Deployment logs

Provisioning state  
Succeeded

Compute type  
Managed

Created by  
Purwadi

Created on  
1/16/2022 10:18:07 AM

Last updated on  
1/16/2022 10:19:35 AM

Authentication type  
Key

Swagger URI  
<https://endpoint2.southeastasia.inference.ml.azure.com/swagger.json>

REST endpoint  
<https://endpoint2.southeastasia.inference.ml.azure.com/score>

Metrics  
[View metrics](#)

Tags  
No tags

Properties

azureml.onlineendpointid  
/subscriptions/b667c281-58f0-47a7-899b-63ea6cd8b7e8/resourcegroups/res1/providers/microsoft.machinelearningservices/workspaces/ws1/onlineendpoints/endpoint2

AzureAsyncOperationUri  
<https://management.azure.com/subscriptions/b667c281-58f0-47a7-899b-63ea6cd8b7e8/providers/Microsoft.MachineLearningServices/locations/southeastasia/mfeOperationsStatus/oe:30d3e896-8b3c-450c-bbce-6168b59420e3:2827d323-4eb3-43d5-a07f-08447e36fdff?api-version=2021-10-01>

AzureML Model deployed



# Conclusion

Advantage of deploying Machine Learning model into MLFlow, AzureML, and MLOPs is to enables teams to easily maintaining, monitoring, training, tuning the model, in order to help us build, testing, and generate advanced analytics based on the data, and reproduce better versions of models.

# List of Figure:

► Krishnan, Adithya. (2019). Anomaly Detection with Isolation Forest & Visualization.

<https://towardsdatascience.com/anomaly-detection-with-isolation-forest-visualization-23cd75c281e2>

► Anello, Eugenia. (2021). Anomaly Detection With Isolation Forest.

► <https://betterprogramming.pub/anomaly-detection-with-isolation-forest-e41f1f55cc6>



# Literature:

► Hansi,Chen.Hongzhan,Ma.Xuening,Chua.Deyi,Xue. (2021). Anomaly detection and critical attributes identification for products with multiple operating conditions based on isolation forest.

<https://www.sciencedirect.com/science/article/abs/pii/S1474034620301105>

► Natu Lauchande. (2021).Machine Learning Engineering with Mlflow. Packt Publishing

► Sridhar Alla, Suman Kalyan Adari . (2021). Beginning MLOps with MLFlow Deploy Models in AWS SageMaker, Google Cloud, and Microsoft Azure. Apress.