Liceria & Co.

END-TO-END FRAUD DETECTION WITH MLOPS

BY: NAUFAL FAIZ N.

Liceria & Co.

Fraud Detection

Tujuan Proyek

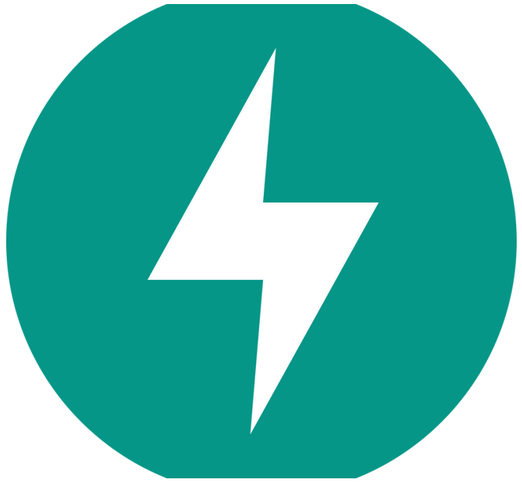
- Membangun sistem pendeteksi kecurangan (fraud detection) yang efisien dengan menggunakan teknik machine learning.
- Implementasi MLOps untuk pipeline yang mencakup pengolahan data, model training, dan deployment dengan integrasi Docker.

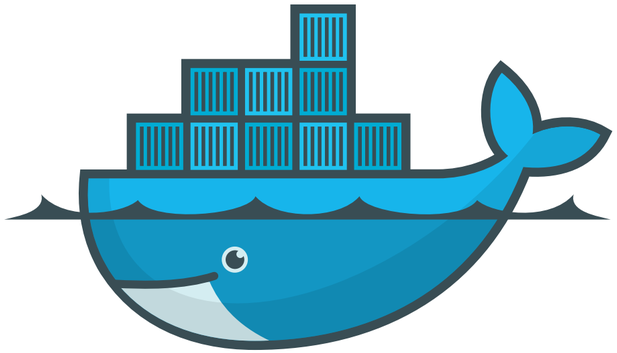
The Ideas

Proyek ini bertujuan untuk membangun sistem yang dapat mendeteksi kecurangan atau penipuan dengan menggunakan data teks dari dataset SMS Spam. Dataset yang digunakan dapat diakses di id-nlp-resource.

Liceria & Co.

Tools yang Digunakan





PostgreSQL

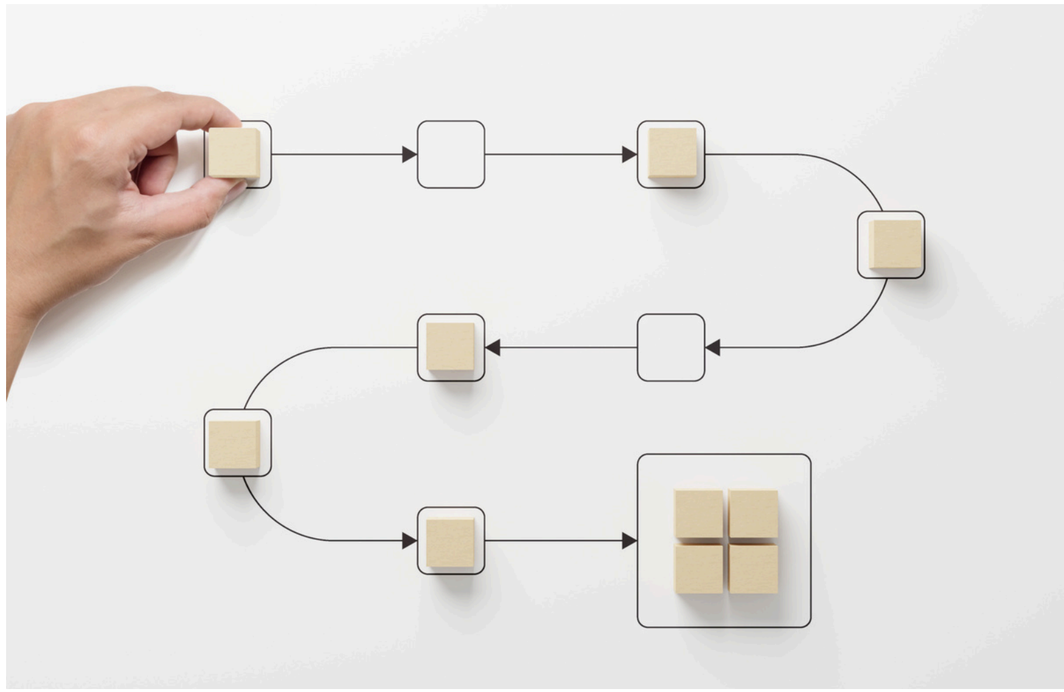
SQLAlchemy

Liceria & Co.

Step by Step

Workflow Proyek

1. Data Collection (BigQuery): Data transaksi yang digunakan untuk deteksi kecurangan diambil dari BigQuery.
2. ETL Process (PostgreSQL): Data diproses dan dipindahkan ke PostgreSQL untuk disimpan dalam bentuk data mart.
3. Data Preprocessing: Melakukan pembersihan teks dan ekstraksi fitur menggunakan TF-IDF.
4. Model Training: Menggunakan Random Forest untuk membangun model deteksi kecurangan.
5. Model Deployment: Model dilatih dan dideploy menggunakan FastAPI di dalam Docker container.



Liceria & Co.

Data Collection In BigQuery

query_transaction RUN SAVE QUERY (CLASSIC) SHARE

```
1  # show data
2  SELECT *
3  FROM `fraud-detection-project-449112.fraud_data.transactions `
4  LIMIT 10
5
6  # cek jumlah fraud dan Tidak fraud
7  SELECT label, COUNT(*) as count
8  FROM `fraud-detection-project-449112.fraud_data.transactions `
9  GROUP BY label
10
11 # cek nilai null
12 SELECT COUNT(*)
13 FROM `fraud-detection-project-449112.fraud_data.transactions `
14 WHERE Label IS NULL
15
16 # seleksi data duplikat
17 SELECT DISTINCT *
18 FROM `fraud-detection-project-449112.fraud_data.transactions `
19
20 # query untuk ekstraksi data
21 SELECT
22   Teks,
23   label
24 FROM `fraud-detection-project-449112.fraud_data.transactions `
```

Liceria & Co.

```python
from google.cloud import bigquery
from sqlalchemy import create_engine
import pandas as pd

# set up BigQuery client
client = bigquery.Client.from_service_account_json('C:\\Users\\NAUFAL FAIZ\\Documents\\Fraud ETL\\data\\fraud-detection-project-449112-d58ffa4a2213.json')

# query untuk menarik data
query = """
SELECT
  Teks,
  label
FROM `fraud-detection-project-449112.fraud_data.transactions `;
"""
df = client.query(query).to_dataframe()

# menampilkan beberapa baris pertama
print(df.head())

# set up kredensial PostgreSQL Anda
db_url = 'postgresql://postgres:postgresql123@localhost:5432/fraud_warehouse'

# buat koneksi ke PostgreSQL
engine = create_engine(db_url)

# save dataframe ke PostgreSQL
df.to_sql('transactions_mart', engine, if_exists='replace', index=False)

print("Data berhasil disimpan ke PostgreSQL")
```

ETL Proses

Liceria & Co.

Data Stored PostgreSQL

Query | Query History

```
1 SELECT *
2 FROM transactions_mart
```

Data Output | Messages | Notifications

	Teks text	label bigint
1	Jika anda bermasalah dgn CC/KT@, stres dgn bunga, pelunasan disc s/d 75%...	1
2	Lelah byr min payment? Kami Solusinya, bantu secara LEGAL penutupan CC/...	1
3	Bisa Dgn BPKB rate 0.99% 3 hari CAIR . hub AYU 081584650877 (WA).mhn m...	1
4	"ROXI CELL" Hanya dengan Rp.100rb Anda bisa jadi agen pulsa elektrik ke se...	1
5	3 RAMADHAN Selamat Anda Pemenang Rp.100jt. PIN CODE 7Y7R8K9Z Info: ...	1
6	Anda brminat cash&kredit mtor scond brg istmwa tipe&merk apa sj.dsini mny...	1
7	ANDA MAU MENANG TOGEL 100% Tembus pasang shio 7 tunggal angka 07....	1
8	Anda mempunyai 3 pesan suara dari 083139195872; untuk mendengarkan pe...	1
9	Anda pmain togel sring kalah.mau bagi hasil. MBAH JOKO bisah mbntu anda ...	1
10	ANEKA SHOP Berbagi Promo Type Blackberry DAKOTA 2,4 jt, TOURCH 1,7 jt, ...	1
11	Anyonghaseyo! Ngaku KPOPers Sejati?? Ayo lengkapi koleksi Video Kpopmu ...	1
12	Artha Cell diskon Brg elektronik BB Dakota 3 juta BB Onyx3 2,8 jt Camera DLL ...	1
13	Ass, mengenai rumah dan tanah yg kemarin, sy sdah survei, sy mrasa cocok, ...	1
14	Ass, Sy Randy. Mengenai mobil yg sdh sy liat kondisi'y , kebetulan kami bermi...	1
15	Ass,maaf sy WAHYU.mengenai Mobil bpk yg mau di jual sy sudah lihat dan co...	1
16	Ass. Maaf sy by hj.suri yang kemarin survey rumah anda. Mengenai harga bis...	1
17	Ass.sy SUGIYONO yg tempo hari mngenai mobil TOYOTANYA.sy brminat.coco...	1

Total rows: 904 Query complete 00:00:00.128

Liceria & Co.

Preprocessing

```python
import pandas as pd
from sqlalchemy import create_engine
import string
import re
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import nltk
nltk.download('punkt')
nltk.download('punkt_tab')
from nltk.tokenize import word_tokenize
from sklearn.feature_extraction.text import TfidfVectorizer

# buat koneksi ke PostgreSQL
db_url = 'postgresql://postgres:postgresql123@localhost:5432/fraud_warehouse'
engine = create_engine(db_url)

# ambil data dari PostgreSQL
query = "SELECT * FROM transactions_mart"
df = pd.read_sql(query, engine)

print(df.head())

# fungsi cleaning
def cleaning(text):
    # stopword
    stop_factory = StopWordRemoverFactory().get_stop_words()
    # stemmer
    stem_factory = StemmerFactory()
    stemmer = stem_factory.create_stemmer()

    text = text.lower() # ubah teks menjadi lower case
    text = text.strip(' ') # menghapus spasi di awal dan di akhir
    text = re.sub(r'\d+', '', text) # menghapus angka
    text = text.translate(str.maketrans('','', string.punctuation)) # menghapus punctuation
    text = re.sub(r'\b[a-zA-Z]\b', '', text) # menghapus kata yang hanya terdiri dari satu huruf
    text = re.sub(r'\s+', ' ', text) # menghapus spasi berlebih
    text = word_tokenize(text) # tokenisasi
    text = [word for word in text if word not in stop_factory] # menghapus stopword
    text = [stemmer.stem(word) for word in text] # stemming
    text = ' '.join(text)
    return text

# fungsi TF-IDF
def tfidf(df):
    tfidf = TfidfVectorizer(max_features=1000)
    X = tfidf.fit_transform(df['Teks_bersih']).toarray()
    y = df['label']

    return X, y, tfidf
```

Liceria & Co.

```python
from preprocessing.preprocessing import *
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sqlalchemy import create_engine
import pandas as pd
import pickle

# buat koneksi ke PostgreSQL
db_url = 'postgresql://postgres:postgresql123@localhost:5432/fraud_warehouse'
engine = create_engine(db_url)

# ambil data dari PostgreSQL
query = "SELECT * FROM transactions_mart"
df = pd.read_sql(query, engine)

# cleaning dataframe
df['Teks_bersih'] = df['Teks'].apply(cleaning)
# ekstraksi fitur
X, y, tfidf = tfidf(df)

# save dataset hasil cleaning ke PostgreSQL dan csv
df.to_sql('transactions_mart_clean', engine, if_exists='replace', index=False)
print("Data berhasil disimpan ke PostgreSQL")

df.to_csv('transactions_mart_clean.csv', index=False)
print("Data berhasil disimpan ke CSV")

# train test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# inisialisasi model
model = RandomForestClassifier()
# training model
model.fit(X_train, y_train)

# evaluasi model
y_pred = model.predict(X_test)

print(classification_report(y_test, y_pred))

# save model
with open('model.pkl', 'wb') as f:
    pickle.dump(model, f)
print("Model berhasil disimpan")

# save tfidf vectorizer
with open('tfidf_vectorizer.pkl', 'wb') as f:
    pickle.dump(tfidf, f)
print("TF-IDF Vectorizer berhasil disimpan")
```

Data Modelling

Liceria & Co.

Model Deployment

```python
from fastapi import FastAPI
from pydantic import BaseModel
import pickle
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer

# load model
with open('C:\\Users\\NAUFAL FAIZ\\Documents\\Fraud ETL\\model\\model.pkl', 'rb') as f:
    model = pickle.load(f)

# load tfidf vectorizer
with open('C:\\Users\\NAUFAL FAIZ\\Documents\\Fraud ETL\\model\\tfidf_vectorizer.pkl', 'rb') as f:
    tfidf = pickle.load(f)

# inisialisasi fastapi
app = FastAPI()

# pydantic untuk input data
class TransactionRequest(BaseModel):
    text: str

# endpoint untuk prediksi fraud
@app.post('/predict')
def predict(transaction: TransactionRequest):
    X_new = tfidf.transform([transaction.text])

    prediction = model.predict(X_new)

    result = "Fraud" if prediction[0] == 1 else "Not Fraud"
    return {"Text": transaction.text, "Prediction": result}

# menjalankan aplikasi fastapi
if __name__ == '__main__':
    import uvicorn
    uvicorn.run(app, host="0.0.0.0", port=8000)
```

Liceria & Co.

Docker Containerization

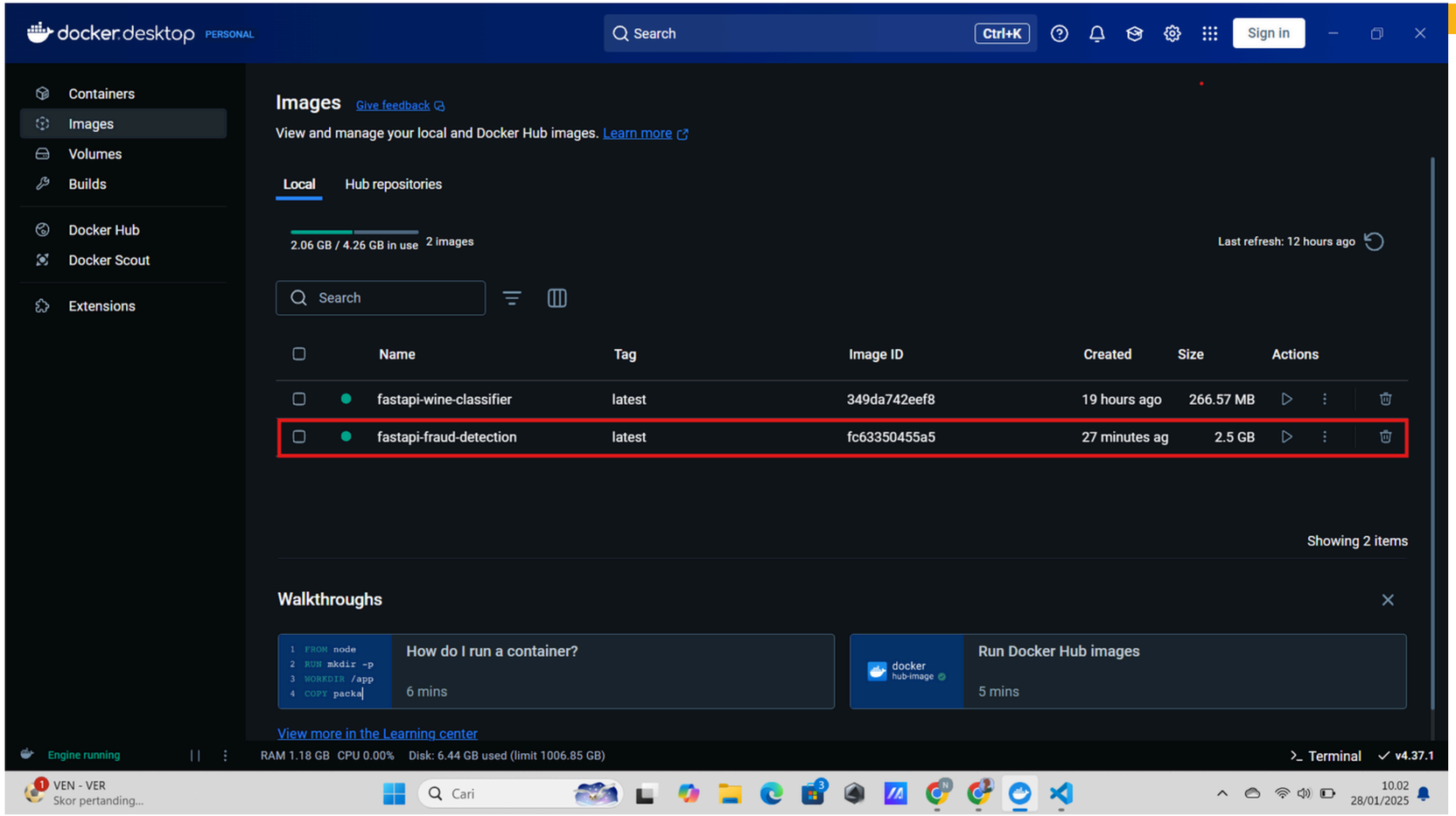
```dockerfile
# menggunakan image Python yang ringan sebagai base image
FROM python:3.8-slim

# install dependencies yang diperlukan
RUN apt-get update && apt-get install -y \
    build-essential \
    libpq-dev \
    postgresql-server-dev-all

# set working directory di dalam container
WORKDIR /app

# salin requirements.txt ke dalam container
COPY requirements.txt .

# salin model ke dalam container
COPY ./model /app/model

# install dependensi yang ada di requirements.txt
RUN pip install -r requirements.txt

# salin seluruh kode aplikasi ke dalam container
COPY . .

# Tentukan perintah untuk menjalankan aplikasi FastAPI dengan Uvicorn
CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "8000"]
```

```
pandas
scikit-learn
nltk
sqlalchemy
psycopg2
psycopg2-binary
fastapi
uvicorn
```

Requirements

Liceria & Co.

Check Docker



THANK YOU