10 Academy: Artificial Intelligence Mastery

Week 8&9 Challenge Document

Date: 05 Feb - 18 Feb 2025

Improved detection of fraud cases for e-commerce and bank transactions

Overview

Business Need

You are a data scientist at **Adey Innovations Inc**., a top company in the financial technology sector. Your company focuses on solutions for e-commerce and banking. Your task is to improve the detection of fraud cases for e-commerce transactions and bank credit transactions. This project aims to create accurate and strong fraud detection models that handle the unique challenges of both types of transaction data. It also includes using geolocation analysis and transaction pattern recognition to improve detection.

Good fraud detection greatly improves transaction security. By using advanced machine learning models and detailed data analysis, **Adey Innovations Inc**. can spot fraudulent activities more accurately. This helps prevent financial losses and builds trust with customers and financial institutions. A well-designed fraud detection system also makes real-time monitoring and reporting more efficient, allowing businesses to act quickly and reduce risks.

This project will involve:

- Analyzing and preprocessing transaction data.
- Creating and engineering features that help identify fraud patterns.
- Building and training machine learning models to detect fraud.
- Evaluating model performance and making necessary improvements.
- Deploying the models for real-time fraud detection and setting up monitoring for continuous improvement.

Data and Features

You will be using the following datasets:

1. Fraud_Data.csv

   Includes e-commerce transaction data aimed at identifying fraudulent activities.

   - **user_id**: A unique identifier for the user who made the transaction.
   - **signup_time**: The timestamp when the user signed up.
   - **purchase_time**: The timestamp when the purchase was made.
   - **purchase_value**: The value of the purchase in dollars.
   - **device_id**: A unique identifier for the device used to make the transaction.
   - **source**: The source through which the user came to the site (e.g., SEO, Ads).
   - **browser**: The browser used to make the transaction (e.g., Chrome, Safari).
   - **sex:** The gender of the user (M for male, F for female).
   - **age**: The age of the user.
   - **ip_address**: The IP address from which the transaction was made.
   - **class**: The target variable where 1 indicates a fraudulent transaction and 0 indicates a non-fraudulent transaction.

2. IpAddress_to_Country.csv

   Maps IP addresses to countries

   - **lower_bound_ip_address**: The lower bound of the IP address range.
   - **upper_bound_ip_address**: The upper bound of the IP address range.
   - **country**: The country corresponding to the IP address range.

3. creditcard.csv

   Contains bank transaction data specifically curated for fraud detection analysis.

   - **Time**: The number of seconds elapsed between this transaction and the first transaction in the dataset.
   - **V1 to V28**: These are anonymized features resulting from a PCA transformation. Their exact nature is not disclosed for privacy reasons, but they represent the underlying patterns in the data.
   - **Amount**: The transaction amount in dollars.
   - **Class**: The target variable where 1 indicates a fraudulent transaction and 0 indicates a non-fraudulent transaction.

Learning Outcomes

Skills:

- Deploying machine learning models using Flask
- Containerizing applications using Docker
- Creating REST APIs for machine learning models
- Testing and validating APIs
- Developing end-to-end deployment pipelines
- Implementing scalable and portable machine-learning solutions
- Developing a dashboard using Dash

Knowledge:

- Principles of model deployment and serving
- Best practices for creating REST APIs
- Understanding of containerization and its benefits
- Techniques for real-time prediction serving
- Security considerations in API development
- Methods for monitoring and maintaining deployed models

Communication:

- Reporting on statistically complex issues

Competency Mapping

The tasks you will carry out in this week's challenge will contribute differently to the 11 competencies 10 Academy identified as essential for job preparedness in the field of Data Engineering, and Machine Learning engineering. The mapping below shows the change (lift) one can obtain through delivering the highest performance in these tasks.

Competency	**Potential contributions from this week**
Professionalism for a global-level job	Articulating business values
Collaboration and Communicating	Reporting to stakeholders
Software Development Frameworks	Using Github for CI/CD, writing modular codes, and packaging
Python Programming	Advanced use of python modules such as Pandas, Matplotlib, Numpy, Scikit-learn, Prophet and other relevant python packages
SQL programming	MySQL db create, read, and write
Data & Analytics Engineering	data filtering, data transformation, and data warehouse management
SHAP and LIME	Model Explainability
Flask	Create a Python API

Team

Tutors:

- Mahlet
- Elias
- Rediet
- Kerod
- Emitinan
- Rehmet

Key Dates

- Discussion on the case - 09:00 UTC on Wednesday 05 Feb 2025. Use #all-week8 to pre-ask questions.
- Interim-1 Submission - 20:00 UTC on Friday 07 Feb 2025.
- Interim-2 Submission - 20:00 UTC on Tuesday 11 Feb 2025.
- Final Submission - 20:00 UTC on Tuesday 18 Feb 2025

Deliverables

Task 1 - Data Analysis and Preprocessing

1. Handle Missing Values
    - Impute or drop missing values
2. Data Cleaning
    - Remove duplicates
    - Correct data types
3. Exploratory Data Analysis (EDA)
    - Univariate analysis
    - Bivariate analysis
4. Merge Datasets for Geolocation Analysis
    - Convert IP addresses to integer format
    - Merge Fraud_Data.csv with IpAddress_to_Country.csv
5. Feature Engineering
    - Transaction frequency and velocity for Fraud_Data.csv
    - Time-Based features for Fraud_Data.csv
        i. hour_of _day
        ii. day_of_week
6. Normalization and Scaling
7. Encode Categorical Features

Task 2 - Model Building and Training

- Data Preparation:
    - Feature and Target Separation [‘**Class**’(creditcard), ‘**class**’(Fraud_Data)]
    - Train-Test Split
- Model Selection
    - Use several models to compare performance, including:
        - Logistic Regression
        - Decision Tree
        - Random Forest
        - Gradient Boosting
        - Multi-Layer Perceptron (MLP)
        - Convolutional Neural Network (CNN)
        - Recurrent Neural Network (RNN)
        - Long Short-Term Memory (LSTM)
- Model Training and Evaluation

    - Training models for both credit card and fraud-data datasets.
- MLOps Steps
    - Versioning and Experiment Tracking
        - Use tools like MLflow to track experiments, log parameters, metrics, and version models.

Task 3 - Model Explainability

Model explainability is crucial for understanding, trust, and debugging in machine learning models. You will use SHAP (Shapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) to interpret the models you built for fraud detection.

- Using SHAP for Explainability
  SHAP values provide a unified measure of feature importance, explaining the contribution of each feature to the prediction.
    - Installing SHAP
      `pip install shap`
    - Explaining a Model with SHAP
    - SHAP Plots
        - **Summary Plot**: Provides an overview of the most important features.
        - **Force Plot**: Visualizes the contribution of features for a single prediction.
        - **Dependence Plot**: This shows the relationship between a feature and the model output.
- Using LIME for Explainability
  LIME explains individual predictions by approximating the model locally with an interpretable model.
    - Installing LIME
      `pip install lime`
    - Explaining a Model with LIME
    - LIME Plots
        - **Feature Importance Plot**: Shows the most influential features for a specific prediction.

Task 4 - Model Deployment and API Development

- Setting Up the Flask API
    - Create the Flask Application
        - Create a new directory for your project:
        - Create a Python script serve_model.py to serve the model using Flask
        - Create a requirements.txt file to list dependencies

- API Development
  - Define API Endpoints
  - Test the API
- Dockerizing the Flask Application
  - Create a Dockerfile in the same directory

```
# Use an official Python runtime as a parent image
FROM python:3.8-slim

# Set the working directory in the container
WORKDIR /app

# Copy the current directory contents into the
container at /app
COPY . .

# Install any needed packages specified in
requirements.txt
RUN pip install -r requirements.txt

# Make port 5000 available to the world outside this
container
EXPOSE 5000

# Run serve_model.py when the container launches
CMD ["python", "serve_model.py"]
```

  - Build and Run the Docker Container
    - Build the Docker image
      `docker build -t fraud-detection-model.`
    - Run the Docker container
      `docker run -p 5000:5000 fraud-detection-model`
- Integrate logging Flask-Logging to track incoming requests, errors, and fraud predictions for continuous monitoring.

Task 5 - Build a Dashboard with Flask and Dash

Create an interactive dashboard using **Dash** for visualizing fraud Insights from your data. The Flask backend will serve data from the datasets, while Dash will be used to visualize insights.

- Add a Flask Endpoint that reads fraud data from a CSV file and serves summary statistics and fraud trends through API endpoints.
- Use Dash to handle the frontend visualizations.
- Dashboard Insights can be:
    - Display total transactions, fraud cases, and fraud percentages in simple summary boxes.
    - Displays a **line chart** showing the number of detected fraud cases over time, using the data served by the Flask backend.
    - Analyze where fraud is occurring geographically
    - Show a bar chart comparing the number of fraud cases across different devices and browsers.
    - Show a chart comparing the number of fraud cases across different devices and browsers.

Tutorials Schedule

Overview

In the following, the colour **purple** indicates morning sessions, and **blue** indicates afternoon sessions.

Wednesday (Feb 05):

- Introduction to the challenge (Mahlet)
- Fraud Detection in e-commerce and credit card transactions(Elias).

Thursday (Feb 06):

- Model Building and Training Including Neural Networks, MLOps using MLflow(Rediet)

Friday (Feb 07):

- How to communicate insight from data (Kerod)

Monday (Feb 10):

- Model Deployment and API Development Using Flask(Emitinan)

Tuesday(Feb 11) :

- How to use Dash for dashboard (Rehmet)

Thursday (Feb 13)

- Q&A (Kerod & Elias)

Monday (Feb 17)

- Q&A (Emitinan & Rediet)

Interim - 1 Submission Friday 07 Feb, 2025

- Interim report - Covering task-1
- Link to your GitHub.

Interim - 2 Submission Tuesday 11 Feb, 2025

- Link to your GitHub.

Feedback

You may not receive detailed comments on your interim submission but will receive a grade.

Final Submission Tuesday 18 Feb 2025

- A blog post entry (which you can submit for example to Medium publishing) or a pdf report.
- Link to your Github code, and make sure to include screenshots demonstrating anything else you have done.

Feedback

You will receive comments/feedback in addition to a grade.

References

Fraud Detection

1. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
2. https://www.kaggle.com/c/ieee-fraud-detection/code
3. https://www.kaggle.com/datasets/vbinh002/fraud-ecommerce/code
4. Fraud Detection
5. https://complyadvantage.com/insights/what-is-fraud-detection/
6. https://www.spiceworks.com/it-security/vulnerability-management/articles/what-is-fraud-detection/

Modeling

1. https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/
2. https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/
3. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/
4. https://www.datacamp.com/tutorial/guide-to-the-gradient-boosting-algorithm
5. https://www.datacamp.com/tutorial/multilayer-perceptrons-in-machine-learning
6. https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns
7. https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939
8. https://www.ibm.com/topics/recurrent-neural-networks
9. https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/
10. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/
11. https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/

Model Explainability

1. https://www.larksuite.com/en_us/topics/ai-glossary/model-explainability-in-ai
2. https://www.analyticsvidhya.com/blog/2021/11/model-explainability/
3. https://www.ibm.com/topics/explainable-ai

4. https://www.datacamp.com/tutorial/explainable-ai-understanding-and-trusting-machine-learning-models

Flask and dash

1. https://flask.palletsprojects.com/en/3.0.x/
2. https://www.geeksforgeeks.org/flask-tutorial/
3. https://realpython.com/python-dash/
4. https://dash.plotly.com/layout