**NAME: TEMITOPE ADENIYAN**

**COURSE: DSCI-6015-01: ARTIFICIAL INTELLIGENCE AND CYBERSECURITY**

**Machine Learning Model Training and Deployment Report**

1. **Introduction**

This report outlines training a machine learning classifier to identify malicious URLs. I also cover the deployment of the model as an API endpoint on Amazon SageMaker and the development of a Streamlit Python client to interact with the deployed endpoint. My project involves using a dataset of URLs that are labeled as malicious, phishing, defacement or benign, and I utilize various technologies such as Colab, Python, Scikit-Learn, Amazon SageMaker, and Streamlit.

2**. Training a Classifier for Malicious URL Detection**

In implementing machine learning for URL classification, the initial step involves data preparation. This encompasses tokenizing URLs and transforming them into numerical features. Subsequently, the data is partitioned into training and testing sets. A Scikit-Learn Random Forest classifier is chosen as the model and trained on the training data to classify URLs as malicious or benign. The model's efficacy on the testing data is assessed using metrics like accuracy, precision, recall, and F1 score, offering insights into its capacity to distinguish between the two URL types.

The training report provides an overview of a machine learning model's performance in identifying malicious URLs. Across 13,024 training samples, the model achieved an accuracy of 53.34%, with validation on 4,070 samples yielding an improved accuracy of 56.25%. Despite the advancement from training to validation, the model's performance remains moderate, as reflected by significant negative loss values.

A screenshot of a computer

Description automatically generated

3. **Deployment on Amazon SageMaker**

After training the dataset I proceeded to the deployment

I created the S3bucket on AWS which was filled with the generated model and tar.gr my code was able to generate.

A screenshot of a computer

Description automatically generated

My code was also generated an endpoint in AWS sagemaker

A screenshot of a computer

Description automatically generated

I created an instance on AWS

A screenshot of a computer

Description automatically generated

After effectively executing all tasks, I encountered hurdles when deploying the model as an API endpoint on Amazon SageMaker. These obstacles stemmed from configuration difficulties and permission constraints. Navigating the learning curve of AWS SageMaker setup and deployment procedures presented additional challenges.

A screenshot of a computer

Description automatically generated

**Conclusion**

This project has been a rich learning experience, underscoring the intricacies of developing and deploying a machine learning model alongside crafting a client application. Despite the successes achieved, it's crucial to acknowledge the challenges encountered, notably during the deployment phase on Amazon SageMaker due to configuration issues and permissions constraints. These obstacles, however, have provided valuable lessons in navigating AWS integration and the complexities of cloud deployment. From model evaluation to addressing user experience considerations and maintenance requirements, each challenge has contributed to a deeper understanding of best practices in real-world machine learning deployments. Moving forward, incorporating suggestions like data augmentation, model optimization, enhanced UI features, continuous monitoring, and security enhancements will further fortify future projects against similar challenges. Overall, this project has not only expanded technical skills but also honed problem-solving abilities in tackling unforeseen obstacles in machine learning deployment scenarios.

Key Insights: To streamline machine learning workflows, it's imperative to prioritize certain steps. Firstly, preprocess data to prepare models effectively. Next, choose algorithms tailored to specific dataset requirements and performance metrics. Lastly, capitalize on AWS services like Amazon SageMaker for smooth deployment and cloud resource management, ensuring efficient operations.

Lessons Acquired: Thorough evaluation of machine learning models pre-deployment is crucial for their effectiveness. Challenges in deploying models as API endpoints may include resource allocation, scalability, and security considerations. A superior user experience in client applications hinges on intuitive UI design and clear result visualization. Post-deployment maintenance involves versioning, monitoring, and addressing drift to maintain model efficacy over time.

Recommendations for Future Enhancement: To elevate model performance and user experience, several strategies can be adopted. These encompass implementing data augmentation techniques for enhanced generalization, optimizing models through hyperparameter tuning, enriching UI with visualizations and feedback mechanisms, continuous performance monitoring, and robust security measures to safeguard data privacy and integrity.

References:

* Scikit-Learn Documentation: <https://scikit-learn.org/>
* Amazon SageMaker Documentation: <https://docs.aws.amazon.com/sagemaker/>
* Streamlit Documentation: <https://docs.streamlit.io/>

Github link: <https://github.com/TemiAdeniyan/SimplifiedMidTerm>