Alexis Alejandro Martínez Suárez

alexis.martinez.6584@gmail.com | +56 9 6847 9046 | linkedIn/alexismartinezs | github/alex-msu

PROFILE

Computer Engineering student with a strong focus on **Data Science**, **Machine Learning**, and the development of **Python-based solutions**. Passionate about building efficient models and solving **real-world problems** through **technology**.

EDUCATION

Bachelor's Degree in Computer Engineering, major in Data Science   Cerrillos, RM | Mar 2022 – Present
DUOC UC – Plaza Oeste

Technician in Business Administration, major in HR   Peñaflor, RM | Graduated 2020
Colegio Comercial de Peñaflor

CERTIFICATIONS

TOEIC | 980/990 – English Level: C1 (MCER)   ETS | Dec 2024

PROJECTS

CUSTOMER CHURN PREDICTION WITH MLOPS ⬈

Python, scikit-learn, MLflow, Pandas, Matplotlib, argparse

Developed a modular and reproducible machine learning pipeline to predict banking customer churn, structured according to MLOps best practices.
Implemented data cleaning, feature engineering, scaling, categorical encoding, model training and evaluation (Random Forest).
Fully integrated MLflow for tracking metrics, artifacts, parameters, and model versioning.
Emphasized recall and F1-score to reduce false negatives.
Result: F1-score and recall ≥ 0.96.

RAINFALL PREDICTION USING ML + WEB APP ⬈

Python, Pandas, scikit-learn, Flask, HTML, pickle

Developed a classification model to predict rainfall using meteorological data from Australia. Built a Flask-based web application to serve the model.
Result: Achieved 88% accuracy on test data.

SENTIMENT ANALYSIS WITH RNN, LSTM & TRANSFORMER ⬈

Python, Keras, TensorFlow, scikit-learn, PyTorch, NLP

Classified tweet sentiments using the Sentiment140 dataset. Compared RNN and LSTM architectures based on precision, F1-score, and loss, ultimately selecting LSTM as the best performer. Also implemented a basic Transformer model in PyTorch as a conceptual experiment.
Result: LSTM outperformed RNN (77% vs 72% accuracy) — the Transformer was included as an exploratory prototype.

IMAGE CLASSIFICATION USING TRANSFER LEARNING (CIFAR-10) ⬈

Python, TensorFlow, Keras, Matplotlib

Designed and trained a CNN using transfer learning to classify CIFAR-10 images into 10 categories. Applied regularization techniques (Dropout, L2, Data Augmentation) to improve generalization and reduce overfitting.
Result: 89.04% validation accuracy.

TECHNICAL SKILLS

Languages

Python, SQL, Java, JavaScript, HTML, CSS.

Data Science and ML

Pandas, NumPy, scikit-learn, Matplotlib, Seaborn. Feature engineering, variable selection, and classification metrics (precision, recall, F1-score).

MLOps

Reproducible model training and evaluation. Use of **MLflow** to track metrics, parameters, and artifacts. Modular organization in Python scripts, use of **argparse** for CLI pipelines, virtual environments (`venv`), and version control with Git/GitHub.

Deep Learning

TensorFlow, Keras, PyTorch.

Tools and Environments

Git, GitHub, Jupyter Notebook, Google Colab (cloud environment), basic knowledge of Docker, Power BI, Notion.

Databases

MySQL, SQLite.

Others

Microsoft Excel, Word, PowerPoint.

SOFT SKILLS

Communication

Clear and effective communication in collaborative environments.

Problem Solving

Analytical thinking focused on solving complex problems.

Continuous Learning

Strong focus on continuous improvement and self-directed learning.

Motivation

Highly motivated to learn and contribute to interdisciplinary projects.

Adaptability

Ability to adapt to agile methodologies and emerging technologies.

CURRENT AREAS OF INTEREST

Prompt Engineering

Design and optimization of instructions to control and improve the output of language models.

Retrieval-Augmented Generation (RAG)

Integration of information retrieval systems with text generation to enhance contextual relevance and accuracy.

Pipeline Automation (CI/CD)

Implementation of automated workflows for model training, evaluation, and deployment using continuous integration and delivery practices.

Model Monitoring in Production

Monitoring the behavior of deployed models to detect performance drift, latency issues, and ensure sustained model quality.