

SUYOG KARKI

Data Science Intern Candidate | ML & Analytics

✉ suyogkarki2@gmail.com ☎ +977 9860103707 📍 Tokha, Kathmandu

🌐 [linkedin.com/in/suyog-karki-8544a033a](https://www.linkedin.com/in/suyog-karki-8544a033a) 🐙 github.com/suyogkarki1

PROFILE

Data Science student experienced in building end-to-end machine learning projects using Python, SQL, and Scikit-learn — covering data cleaning, EDA, feature engineering, preprocessing pipeline, cross-validation, hyperparameter tuning, and model evaluation. Applied this across projects in weather forecasting, customer churn, and healthcare risk. I'm genuinely curious about AI and machine learning, and I want to keep getting better at it — looking for a data science/ML or analytics internship to learn and grow.

EDUCATION

Higher Secondary Education (+2) - Science, 2020 – 2022
Kathmandu Model College (KMC), Bagbazar
GPA: 3.35 / 4.0

Bachelor of Science in Information Technology (BSIT) - Data Science, 2023 – 2027
Kings College, Kathmandu (Affiliated with Westcliff University, California)
Currently in 4th Year
GPA: 3.54 / 4.0

TECHNICAL SKILLS

- Languages : Python · SQL
- Data Analysis : Pandas · NumPy · Excel
- Data Visualization: Matplotlib · Seaborn · Power BI
- Tools: Git & GitHub
- Machine Learning : Scikit-learn
- Learning: Object Detection (OpenCV, YOLO , OCR)

CERTIFICATIONS

- Supervised Learning with Scikit-learn | DataCamp | 2026
- Associate Data Scientist Certification | DataCamp | 2026
- SQL Associate | DataCamp | 2026

PROJECTS

Diabetes Risk Detection Using Gradient Boosting

Python · Pandas · NumPy · Matplotlib · Seaborn · Scikit-learn · Streamlit

Github: <https://github.com/suyogkarki1/Diabetes-Detection-Using-Gradient-Boosting.git> 

- Built an end-to-end diabetes screening pipeline on 15,000+ patient records with a severe **98:2** class imbalance, applying **stratified splitting** and **sample-weighted training** across five classifiers to avoid the accuracy trap common in imbalanced medical datasets.
- Enforced train-only statistical analysis (**Chi-square, Cramér's V, Welch's t-test, Cohen's d**) to eliminate test-set leakage during EDA, then selected Gradient Boosting via cross-validated PR-AUC and tuned it with GridSearchCV — reaching **0.99** balanced accuracy on a held-out test set.

- Engineered clinical-domain features (pulse pressure, mean arterial pressure, symptom count, risk flags) using fixed medical thresholds, and validated via feature importance that no single input was driving predictions before trusting the results.
- Trained a second model excluding the direct blood glucose reading to test whether vitals and symptoms alone — the signals available *before* a lab test — could still screen for risk; it retained **99%** balanced accuracy, and deployed both as an interactive Streamlit app for real-time prediction.

Kathmandu Rain Prediction

Python · Pandas · NumPy · Scikit-learn · Matplotlib · Seaborn · Streamlit

GitHub: <https://github.com/suyogkarki1/Rain-Prediction-> 

- Sourced a 10-year Kathmandu weather dataset (2015–2025, 3,468 records) from Kaggle, and performed data cleaning, unit conversion, and feature engineering across 14 meteorological variables for supervised classification.
- Built and evaluated four classification models — Logistic Regression, Decision Tree, K-Nearest Neighbors, and Random Forest — using stratified 6-fold cross-validation; selected Random Forest as the final model, achieving **84% accuracy and 0.90 ROC-AUC** on held-out test data.
- Conducted feature importance analysis identifying **humidity, temperature, and cloud cover** as the strongest rainfall predictors, validating the model against real-world Kathmandu Valley monsoon patterns.
- Applied **bootstrap resampling** to estimate 2026 monthly rain probabilities with lower and upper uncertainty bounds, and developed a separate **SARIMA** seasonal time-series model to forecast monthly rainy-day percentages for comparison.
- Deployed the trained model as an interactive Streamlit web application (**SkyCast**) enabling real-time rain classification from user-defined weather inputs and visualizing seasonal forecast trends.

Customer Churn Prediction

Python · Pandas · Scikit-learn · Matplotlib · Seaborn

GitHub : <https://github.com/suyogkarki1/customer-churn-prediction> 

- Analyzed a Kaggle dataset of 10,000 European bank customers to identify behavioral patterns linked to churn — including credit score, account balance, and product usage habits.
- Engineered new features including **BalancePerProduct** and **AgeGroup**; conducted full EDA to uncover class imbalance and key churn indicators before applying preprocessing steps.
- Achieved **86.35% accuracy and 0.87 ROC-AUC** using Gradient Boosting Classifier, selected after benchmarking against simpler models including Logistic Regression and Decision Tree.
- Feature importance analysis revealed **Age and NumOfProducts** as the strongest churn predictors — suggesting retention teams should prioritize older single-product customers as the highest risk segment

Smart Healthcare and Lifestyle Prediction

Python · Pandas · Scikit-learn · Matplotlib · Seaborn

GitHub : <https://github.com/suyogkarki1/Smart-Healthcare-and-Lifestyle-Prediction> 

- Worked with a Kaggle healthcare dataset of 5,000 patient records containing clinical indicators, including blood pressure, cholesterol, BMI, smoking habits, and chest pain severity.
- Performed end-to-end pipeline — data cleaning, EDA, feature analysis, model training, and evaluation — comparing multiple classifiers before finalizing Logistic Regression.
- Achieved **99.4% accuracy and 0.999 ROC-AUC**; identified that well-separated class distributions in the dataset contributed to high performance and validated results using cross-validation to rule out data leakage.

- Feature importance analysis revealed **Age, BMI, and Smoking** as the strongest heart disease predictors — insights that can help clinicians prioritize high-risk patients for early screening and preventive intervention.

UAE ID Card Number Extraction Pipeline

Python · YOLOv11 (Ultralytics) · OpenCV · Tesseract OCR · PostgreSQL

Github: https://github.com/suyogkarki1/object_detection.git 

- Annotated a custom dataset of 424 UAE ID card images on Ultralytics HUB, labeling the ID-number field as a dedicated detection class, split into 405 train / 10 validation / 9 test images.
- Trained and deployed a custom YOLOv8 model via Ultralytics' hosted endpoint, achieving **99.5% mAP50** and **95.0% recall** on ID-number localization.
- Built an OCR pipeline (OpenCV preprocessing + Tesseract) to extract and clean detected ID numbers via regex, then persisted records to PostgreSQL with conflict-safe deduplication.

LANGUAGES

Nepali

Native

English

Fluent