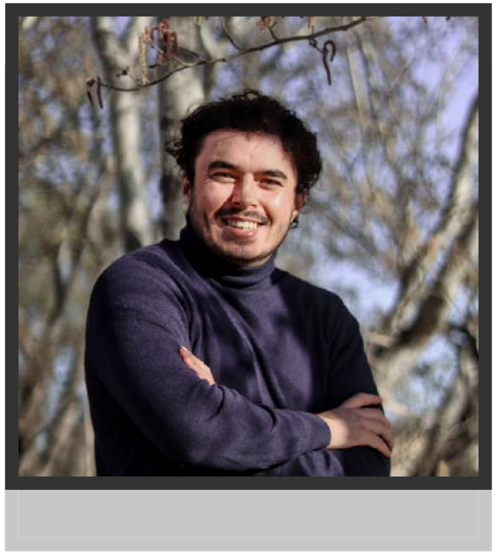
PORTFOLIO

Machine Learning & Data Science Projects

NASSER CHAOUCHI

WHO AM I?

📌 My academic journey



I'm Nasser, a **French computer science engineer**, passionate about **artificial intelligence**, **data**, and **innovation**. I **graduated from UTC** (Université de Technologie de Compiègne) in 2025 with **a major in AI and Data Science**.

I completed:

- A **dual-focused internship** at **Numberly** as a **Data Engineer and Project Manager**, combining technical and management responsibilities.
- An **exchange semester** at the **Escuela de Ingeniería y Arquitectura in Zaragoza** (Spain), as part of a **Data Science Master's program**.
- A **final-year internship** at **Ubisoft** as a **Data Scientist**, working on **real-world game data** and **predictive models**.

💪 Strengths

Curious	**Rigorous**	**Positive**	**Patient**

❤️ Interests

Artificial Intelligence	**Sports**	**Literature**	**Chess**

MY WORK EXPERIENCE

An **internship** at **Ubisoft** from **October 2024** to **March 2025** as a **Data Scientist**, with the following missions:

Audiences Understanding

Segmentation Based on Players' Profiles

Player Behavior Prediction

I worked on the game **Avatar: Frontiers of Pandora**. My role was **to understand the game's underperformance** and to **identify and target potential players** within the **Ubisoft ecosystem** who would most likely acquire the game.

The project was divided into **three main phases**:

Ad-hoc Analyses → **Clustering Development** → **Classifier with Prediction**

You can contact the team manager for a reference:

- Nicolas Tatin, Associate Director, Data & Analytics

Data Analysis

Data Science

MY PROJECTS – THE MOVIE RECOMMENDER SYSTEM

CONTEXT

- **Dataset**: MovieLens 32M
- **Goal**: Recommend movies users might like, based on behavior and content features.
- **Type**: Hybrid Recommendation System
    - Collaborative Filtering (ratings)
    - Content-Based Filtering (genres, titles)
- **Size**: 32M+ ratings, ~270k users, 62k movies.

BUILT WITH

- Scikit-learn
- Pandas, NumPy
- Seaborn
- Matplotlib
- HuggingFace datasets
- Streamlit

Open the repository

Open the interface
(with MovieLens 1M)

APPROACH

- **Data Cleaning**: Merged movies.csv and ratings.csv, extracted year, processed genres.
- **Collaborative Filtering**: Built user-item matrix, applied cosine similarity.
- **Content-Based Filtering**: Used TF-IDF/CountVectorizer on genres and titles.
- **Hybrid Strategy**: Combined top recommendations from both approaches.
- **Implemented multiple strategies**: Most rated movies, Top-rated by genre, Top-rated by year, User-user collaborative hybrid, Item-item collaborative hybrid.
- **Profile-Based Recommendation**: Built a user profile from favorite movies to generate personalized suggestions.

WHAT I LEARNED

- **Designing** and **comparing recommender strategies**.
- Using **similarity metric**s (cosine) on **sparse data**.
- **Evaluating trade-offs** between **relevance** and **diversity**.

MY PROJECTS – THE MOVIE RECOMMENDER SYSTEM

Deploy

Navigation

Select a view

- ○ Popular Picks
- ○ By Movie (Item-Item Hybrid)
- ○ By User (Item-Item Hybrid)
- ○ Manual Selection (Item-Item Hybrid)
- ○ By User (User-User Hybrid)
- ● Manual Selection (User-User Hybrid)

Top-N recommendations

10 − +

Data: MovieLens 1M (via Hugging Face)

MovieLens Hybrid Recommender

Hybrid recommendation system combining **ratings signals** and **content (genres)**. Use the navigation on the left to explore different strategies.

Manual Selection → User-User Hybrid

Search a movie to add

vita

Choose a movie

Life Is Beautiful (La Vita è bella) (1997)

Add to selection

Current selection

Clear

Godfather, The (1972) | Avengers, The (1998) | Life Is Beautiful (La Vita è bella) (1997)

Genre weight

0.30

Recommend from selection

Title	Year	Genres
Almost Famous	2000	Comedy|Drama
American Beauty	1999	Comedy|Drama
Being John Malkovich	1999	Comedy
Gladiator	2000	Action|Drama
GoodFellas	1990	Crime|Drama
Matrix, The	1999	Action|Sci-Fi|Thriller
Pulp Fiction	1994	Crime|Drama
Saving Private Ryan	1998	Action|Drama|War
Schindler's List	1993	Drama|War
Silence of the Lambs, The	1991	Drama|Thriller

MY PROJECTS – THE TWITTER SENTIMENT ANALYSIS

CONTEXT

- **Dataset**: Sentiment140
- **Goal**: Predict sentiment (positive or negative) from tweets
- **Type**: Supervised, NLP
    - Classical Machine Learning
    - Deep Learning
- **Size**: 1.6M labeled tweets (short, noisy, informal text).

BUILT WITH

- Python, scikit-learn, joblib
- Pandas, NumPy, Matplotlib, Seaborn
- Hugging Face Transformers (BERT)
- PyTorch
- Streamlit

Open the repository

Open the interface

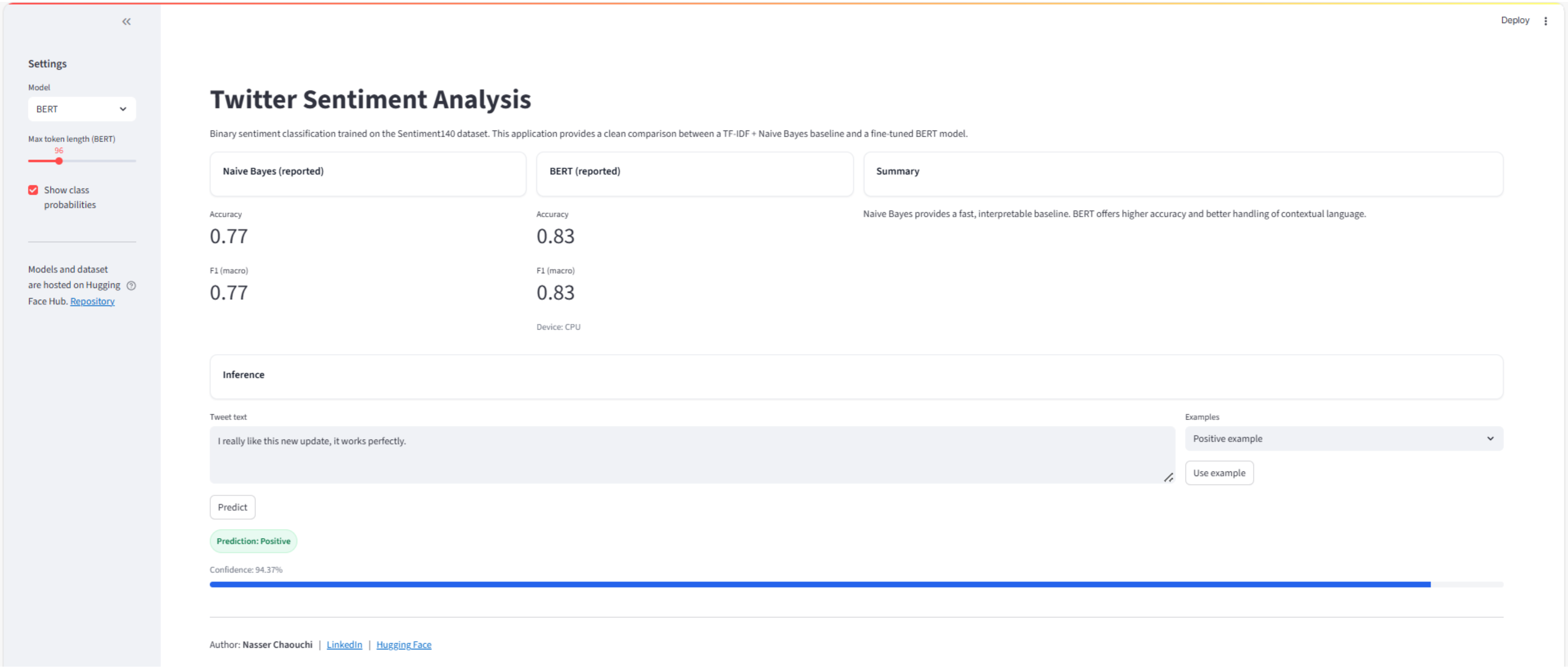
APPROACH

- **Preprocessing**: Cleaned and normalized tweets (tokenization, lowercasing, removal of emojis, URLs, hashtags).
- **Baseline Model**: **TF-IDF and Naive Bayes**, a lightweight and interpretable solution for fast text classification.
- **Advanced Model**: **BERT fine-tuning with Hugging Face Transformers**, leveraging contextual embeddings for higher accuracy.
- **Evaluation**: Accuracy, Macro F1-score, Confusion Matrix, with additional analysis of ambiguous tweets.
- **Deployment**: **Real-time sentiment prediction** through an interactive Streamlit application.
- **Resources:** Dataset and trained models hosted and shared on Hugging Face Hub.

WHAT I LEARNED

- The **value of preprocessing** when working with noisy Twitter data.
- How to **contrast classical ML approaches with state-of-the-art NLP** methods.
- The **trade-off** between a simple, fast model (Naive Bayes) and a more complex, accurate one (BERT).
- Best practices for **hosting and sharing datasets and models** on Hugging Face Hub.

MY PROJECTS – THE TWITTER SENTIMENT ANALYSIS



CONTEXT

- **Dataset**: Stanford Dogs Dataset (subset, ~2000 images, 15 breeds).
- **Goal**: Build a convolutional neural network to classify dog breeds from images.
- **Type**: Supervised, Image Classification (Deep Learning).
- **Size**: ~2000 labeled images, high intra-class variability.

BUILT WITH

- Python – PyTorch (ResNet18, Grad-CAM)
- Torchvision (pretrained models, transforms)
- Pandas, NumPy, Matplotlib, Seaborn
- Streamlit (interactive deployment)

Open the repository

Open the interface

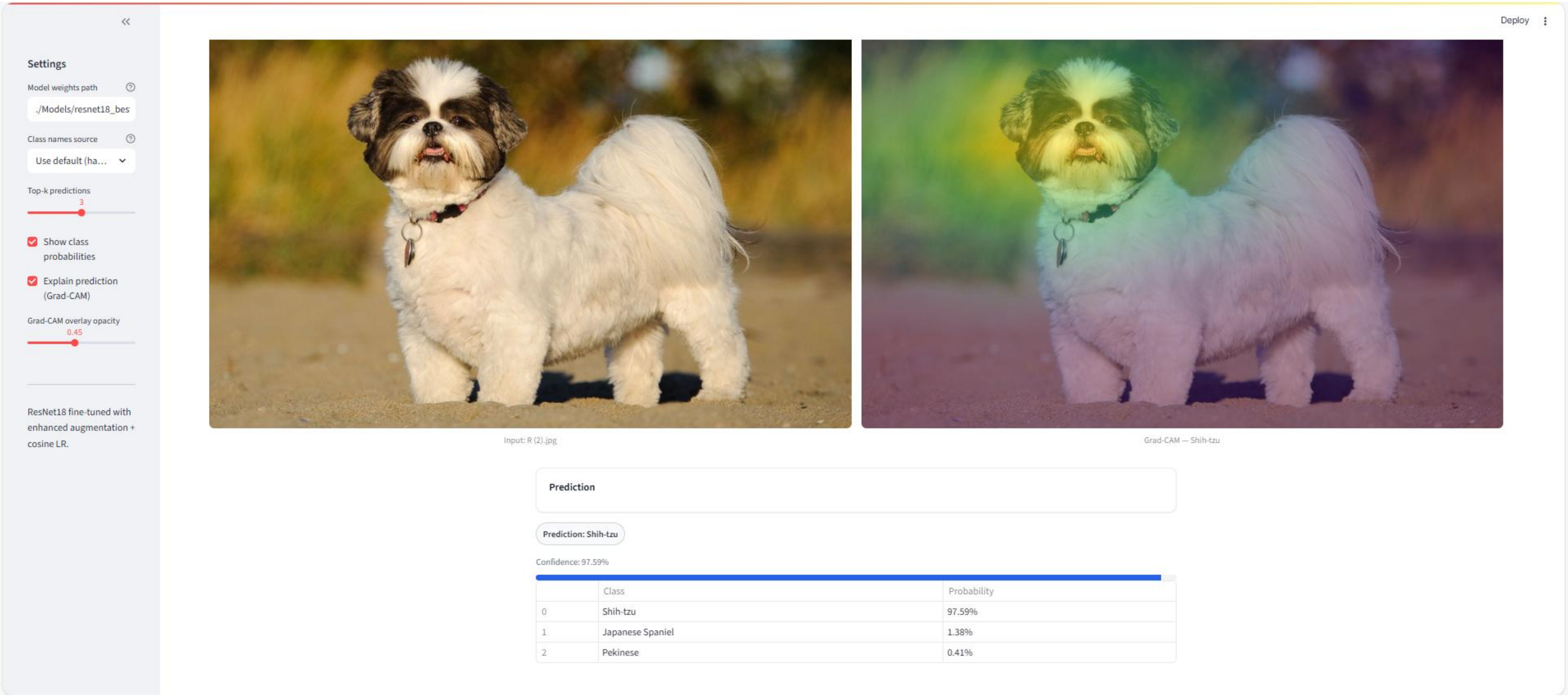
APPROACH

- **Data Preprocessing**: Resized images, normalization, data augmentation (rotation, flip, zoom, shift).
- **Model Architecture**: Fine-tuned **ResNet18** with transfer learning.
- **Training Strategy**: **Baseline** (moderate augmentation) vs **Enhanced Augmentation + Cosine LR scheduler**.
- **Evaluation**: Accuracy, Macro F1-score, Confusion Matrix, per-class results.
- **Explainability**: Integrated **Grad-CAM heatmaps** to visualize model decisions.
- **Deployment**: Built an interactive **Streamlit application** for predictions and explainability.

WHAT I LEARNED

- Leveraging **transfer learning** for limited datasets.
- Improving generalization with **data augmentation and LR scheduling**.
- Combining **performance metrics with interpretability (Grad-CAM)**.
- Deploying an **end-to-end Computer Vision application**.

MY PROJECTS – THE DOG BREED CLASSIFICATION (CNN)



MY PROJECTS – MULTICLASS CLASSIFICATION FOR DIABETES

CONTEXT

- **Dataset**: Multiclass Diabetes Dataset
- **Goal**: Classify patients into several diabetes stages
- **Type**: Supervised, Multiclass classification
- **Size**: 264 patients, 12 features

BUILT WITH

- Scikit-learn
- Pandas, NumPy
- Seaborn
- Matplotlib
- Streamlit

Open the repository

Open the interface

APPROACH

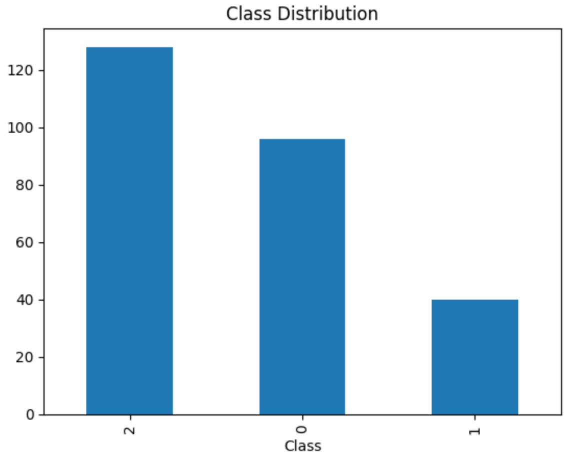
- **EDA & Preprocessing**: Analyzed feature distributions, handled missing values, balanced classes, and scaled data.
- **Model tested**: Logistic Regression, Random Forest and K-Nearest Neighbour.
- **Cross Validation**: Ensured robust performance and avoided overfitting.
- **Evaluation**: Confusion Matrix, Classification report (F1-Score, Accuracy, Recall).
- **Final model (Random Forest)**
    - **Accuracy**: 97%
    - **Macro F1-score** (better suited to class imbalance): 0.98

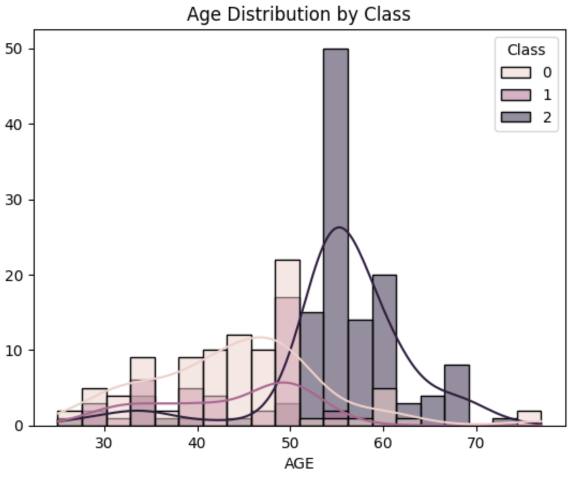
WHAT I LEARNED

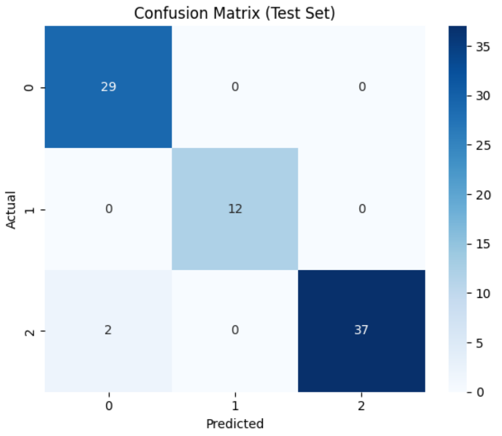
- **How to handle imbalanced multiclass data**.
- The **importance** of **feature engineering** and **model tuning**.
- Model explainability with **SHAP** or **feature importance**.

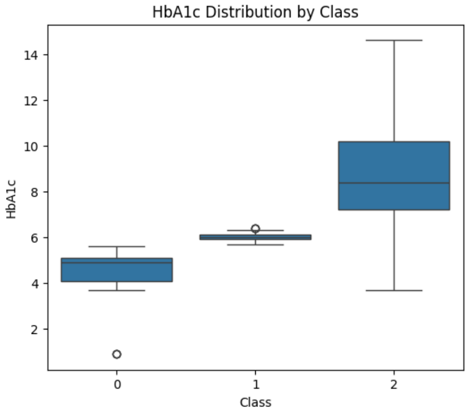
MY PROJECTS – MULTICLASS CLASSIFICATION FOR DIABETES

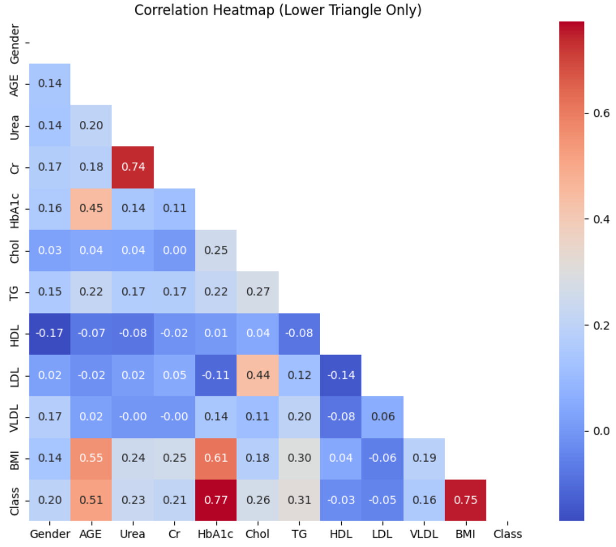
Variable	F-statistic	p-value
AGE	60.368	0.00000
BMI	190.565	0.00000
TG	14.218	0.00000
HbA1c	200.415	0.00000
Chol	9.881	0.00007
Urea	9.115	0.00015
Cr	6.466	0.00182
VLDL	3.712	0.02573
LDL	0.957	0.38526
HDL	0.488	0.61457

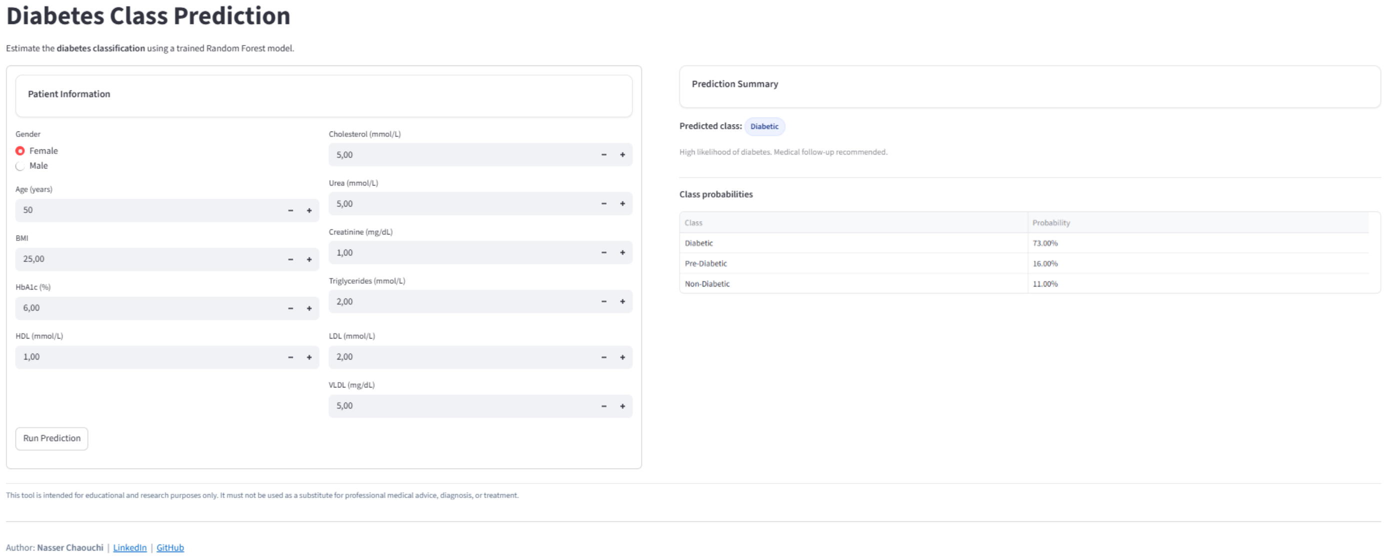












Portfolio

Nasser CHAOUCHI

MY PROJECTS – THE CKD AND DIALYSIS PREDICTION

CONTEXT

- **Dataset**: Kidney Disease Risk Dataset
- **Goal**: Predict CKD status and dialysis need based on clinical and biological data
- **Type**: Supervised, Binary classification (2 targets: CKD_Status, Dialysis_Needed)
- **Size**: 2304 patients, 9 features

BUILT WITH

- Scikit-learn
- Pandas, NumPy
- Seaborn
- Matplotlib
- XGBoost
- Streamlit

Open the repository

Open the interface

APPROACH

- **EDA & Preprocessing**: Explored feature relationships, handled missing values, encoded categorical data, scaled numerical features.
- **Model tested**: Logistic Regression, Random Forest, Gradient Boosting, XGBoost and K-Nearest Neighbour
- **Cross Validation**: Ensured robustness and reduced overfitting risk.
- **Evaluation**: Classification Report, ROC-AUC, F1-Score, Accuracy.
- Best model was Gradient Boosting, but due to class imbalance, Random Forest gave more reliable results for generalization.

  → **Accuracy**: 100% for CKD_Status — Accuracy 100% but a F1-Score 0.97 (class imbalance) for the Dialysis_Needed.

  → **Separate models trained for each target**.

WHAT I LEARNED

- **Managing dual target classification**.
- **Handling noisy** and **medical data**.
- Improving interpretability with **SHAP values**.

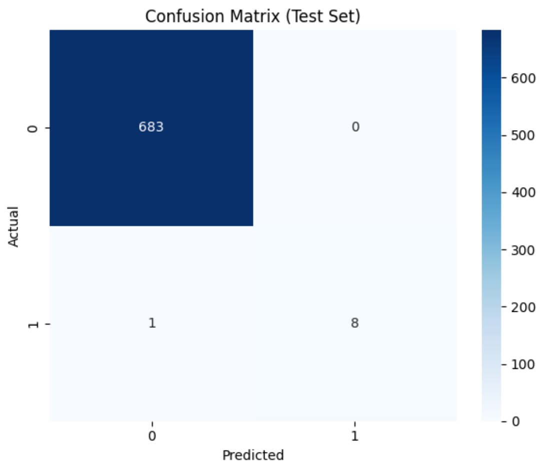
MY PROJECTS – THE CKD AND DIALYSIS PREDICTION

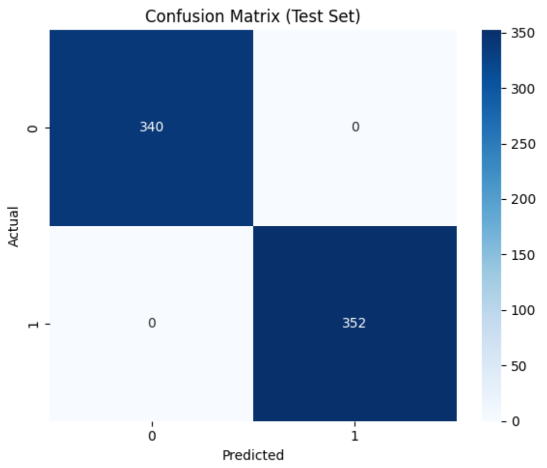
Evaluation on Test Set – CKD_Status

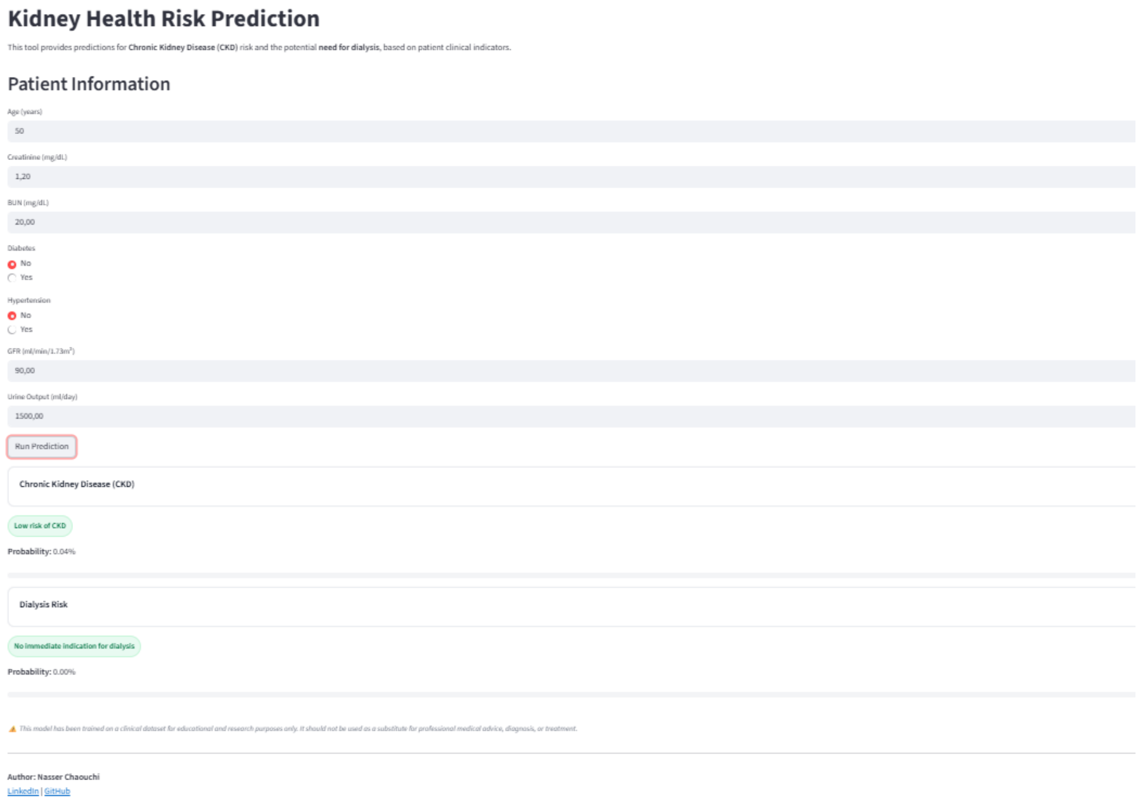
Class	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	340
1	1.00	1.00	1.00	352
Accuracy			**1.00**	692
Macro avg	1.00	1.00	1.00	692
Weighted avg	1.00	1.00	1.00	692

Evaluation on Test Set – Dialysis_Needed

Class	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	683
1	1.00	0.89	0.94	9
Accuracy			**1.00**	692
Macro avg	1.00	0.94	0.97	692
Weighted avg	1.00	1.00	1.00	692







Don’t hesitate to reach out to me