Social Media Influence Prediction Project

Machine Learning Approach with Enhanced Algorithm Suite

Final Project Documentation & Execution Instructions

Students:	Mai Le Quynh (1133954), Duong Kim Ngan (1137184)
Course:	CM763 - Artificial Intelligence
Supervisor:	Professor Qazi Mazhar Ul Haq
Institution:	Department of Management, Yuan Ze University, Taiwan
Date:	November 2025

Project Summary

This research upgrades an existing Twitter influence prediction implementation from 4 to 9 machine learning algorithms across four families (statistical, tree-based, ensemble, deep learning). Using the Kaggle "Influencers in Social Networks" dataset (~5,000 well-balanced user pairs), Gradient Boosting achieved 77.03% validation accuracy with minimal overfitting (-0.48%). The analysis reveals listed_count (professional recognition) dominates with ~60% feature importance, challenging conventional assumptions about follower count. Implementation includes a Jupyter notebook for training and an interactive Python chatbot for real-time predictions.

Original code: github.com/shimonyagrawal/Social-Media-Analytics-Twitter

GitHub Repository

URL: https://github.com/chinicapster/1141_CM763

Branch: main | **Status:** Production Ready

Execution Instructions

Step 1: Clone and Setup

```
git clone https://github.com/chinicapster/1141_CM763.git
cd 1141_CM763
pip install -r requirements.txt  # Install dependencies
```

Step 2: Train Models (Jupyter Notebook)

```
cd "[Group work] Upgraded code"
jupyter notebook "[Upgraded] Twitter_Analytics.ipynb"
```

In Jupyter: Cell → Run All (trains all 9 models, generates visualizations)

Step 3: Use Interactive Chatbot (Optional)

```
python interactive_chatbot.py
```

Enter Twitter metrics when prompted to get influence predictions instantly

Technical Details

Dataset:	Included at [Original code]/train.csv - no download needed, well-balanced
Implementation:	Single Jupyter notebook for training + Python chatbot for deployment
Models:	9 algorithms: LR, RF, KNN, XGBoost, GradBoost, SVM, DT, AdaBoost, NN
Best Model:	Gradient Boosting: 77.03% accuracy, -0.48% overfitting (excellent)
Top Feature:	A/B_listed_count (60% importance) - professional recognition metric
Requirements:	Python 3.8+, Jupyter, scikit-learn, PyTorch, XGBoost (in requirements.txt)

Key Contributions & Results

Aspect	Original Implementation	Our Upgraded Version
Algorithms	4 models	9 models (+125% coverage)
Algorithm Families	2 families	4 families (added ensemble + deep learning)

Aspect	Original Implementation	Our Upgraded Version
Best Accuracy	81% (XGBoost)	77.03% (Gradient Boosting)
Overfitting	6.79%	-0.48% (better generalization)
Deep Learning	No	Yes (PyTorch Neural Network)
Deployment	None	Interactive Python chatbot
Evaluation	Basic metrics	ROC curves, confusion matrices, feature importance

Complete code in single notebook for training | Interactive chatbot for predictions | All reproducible with random_state=42
Week 17 Final Submission | CM763 Artificial Intelligence | Yuan Ze University