Instructions

Datathon 2025 - Group 24 - Cat B

Table of contents

I. Introduction

Identifying corporate ownership structures is crucial for businesses, investors, and regulators to assess financial control, decision-making power, and market influence. Our project explores which factors play more important roles in determining whether a company is Domestic Ultimate or Global Ultimate; and thus applies machine learning models (Logistic Regression, KNN, Random Forest, Support Vector Machine, CNN, XGBoost) to predict corporate ownership classifications based on those factors.

II. Requirements

The following libraries are required to run the notebook, and they are typically available in Google Colab:

- Python 3.10
- Libraries: numpy, pandas, scikit-learn, matplotlib, seaborn, xgboost, tensorflow, scipy
- Additional libraries can be installed within the notebook using !pip install.

III. Usage

1. Running the notebook
    a. **Open the Notebook in Colab** - Click on the Colab link above to open the notebook in Google Colab.
    b. **Run Cells Sequentially** - Run each cell in sequence. Some cells may prompt for data upload or require specific configurations, depending on the dataset and tasks.
2. Data Upload:
   If the notebook requires data files:
   - You may upload the cancer dataset directly to Colab, or
   - Use Google Drive to load large or persistent files:
   \`\`\`python
   from google.colab import drive
   drive.mount('/content/drive') \`\`\`

IV. Notebook Overview

1. **Data loading**

- Loads the Champions_Group_2025.csv dataset , from a CSV file.
- Provides an initial preview of the data, examining its shape, feature names, and the first few rows to understand its structure.

2. **Data preprocessing**
    - Delete columns with high proportion of missing values such as “Square Footage”, “Import/Export Status”, “Fiscal Year End”, “Employees (Single Site)”
    - Exploratory Data Analysis (EDA)
        - For numerical variables, we visualize the distribution of key features, using histograms, scatter plots, and box plots to highlight potential patterns.
        - For categorical variables, we do statistical tests to check its importance and correlation to the target variable
3. **Feature engineering/selection**
    - Remove input variables that have weak association to the target variable
    - Keep input variables that have strong association to the target variable which are Log_Employee (Domestic), Log_Employee (Global), Log_Sales (Domestic), Log_Sales (Global), Mismatch, Ownership Type, and Age.
4. **Model training**
    - Trains multiple machine learning and deep learning models (e.g. Multinomial Logistic Regression, KNN, Random Forest, XGBoost, and Convolutional Neural Network) on the preprocessed data.
    - Configures model parameters and applies validation to assess stability and accuracy.
5. **Model evaluation**
    - Evaluates models on the test data using metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.
    - Visualizes model performance with confusion matrices and ROC curves to help assess classification effectiveness.
6. **Prediction and Interpretation**
    - After splitting data into train data and test data with ratio 80/20, we make predictions on the test data
7. **Hyperparameter Tuning**
    - Optimizes model parameters (GridSearchCV for Multinomial Logistic Regression, KNN, and Random Forest, Manual Hyperparameter Tuning for XGBoost, and RandomizedSearchCV for Convolutional Neural Network) to achieve the best possible performance.

V. Results & Key Insights

1. **Results**

Logistic Regression

Below is the evaluation report after using Logistic Regression to predict.

Accuracy	Precision	Recall	F1-Score	ROC-AUC Score
0.61	0.62	0.62	0.58	0.74
Structure				
Class	Precision	Recall	F1-Score	
0	0.79	0.54	0.64	
1	0.57	0.22	0.32	
2	0.60	0.87	0.71	

Table 11. Evaluation metrics of Logistic Regression

Overall, the Logistic Regression - as a base model - performs relatively well.

K Nearest Neighbour

Accuracy	Precision	Recall	F1-Score	ROC-AUC Score
0.70	0.70	0.70	0.70	0.85
Structure				
Class	Precision	Recall	F1-Score	
0	0.82	0.77	0.80	
1	0.59	0.58	0.58	
2	0.72	0.74	0.73	

Random Forest

Accuracy	Precision	Recall	F1-Score	ROC-AUC Score

0.74	0.73	0.73	0.73	0.89
Structure				
Class	Precision	Recall	F1-Score	
0	0.87	0.90	0.89	
1	0.63	0.56	0.59	
2	0.74	0.77	0.75	

XGBoost

Below is the evaluation report after using XGBoost to predict.

Accuracy	Precision	Recall	F1-Score	ROC-AUC Score
0.73	0.73	0.73	0.73	0.89
	Structure			
Class	Precision	Recall	F1-Score	ROC-AUC
0	0.89	0.89	0.89	
1	0.61	0.60	0.61	
2	0.74	0.75	0.75	

CNN

Accuracy	Precision	Recall	F1-Score	ROC-AUC Score
0.70	0.70	0.70	0.70	0.87
	Structure			
Class	Precision	Recall	F1-Score	
0	0.76	0.94	0.84	
1	0.61	0.51	0.56	

2	0.72	0.72	0.72	

Comparison between Models & Discussion

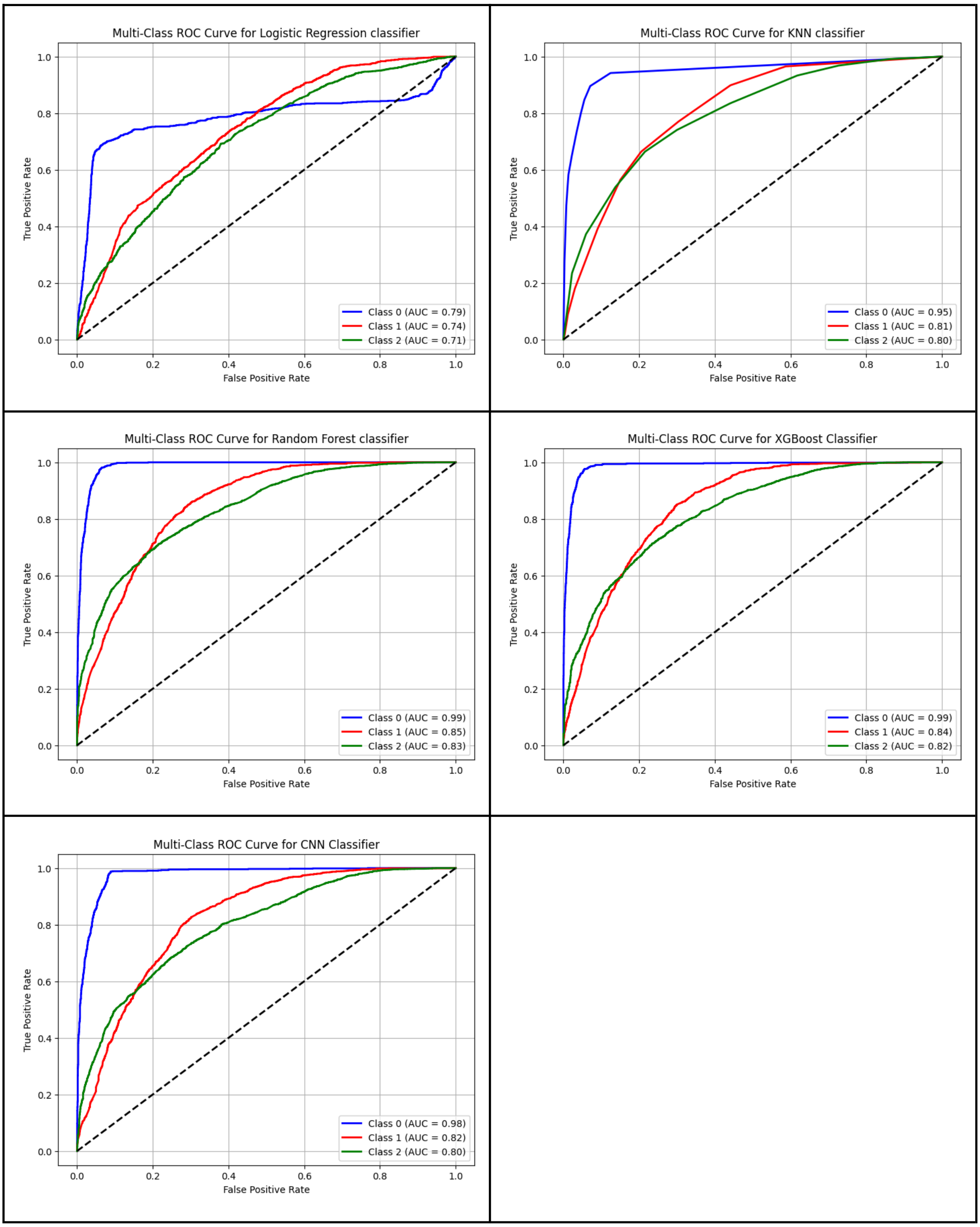
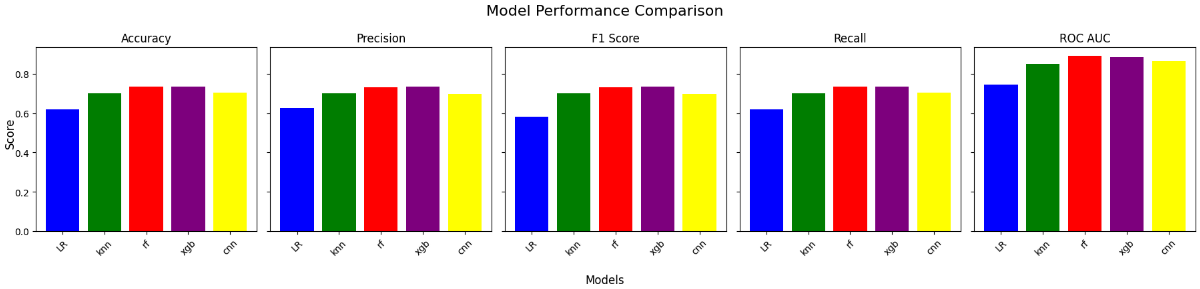


Figure. ROC curve and AUC value of 5 models



Based on the model performance comparison, the Random Forest (RF) and XGBoost (XGB) models stand out as top performers across key evaluation metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. Both models achieve high scores consistently, making them reliable choices, especially in business contexts where balanced performance and robustness are crucial. RF slightly edges out in ROC-AUC, indicating superior capability in distinguishing between classes, which is critical in scenarios like fraud detection or risk assessment. If the business requires high precision (e.g., minimizing false positives in customer targeting), RF and XGB are preferable.

However, if recall is more important (e.g., identifying as many potential risks as possible in medical diagnoses), RF also performs strongly. While K-Nearest Neighbors (KNN) and CNN show decent results, they lag slightly behind, and Logistic Regression (LR) performs the weakest overall, making it less suitable for complex business applications. In summary, **RF** and **XGB** are the recommended models for business-critical tasks due to their balanced and high performance across all relevant metrics.