AI1010 - Introduction to AI
Office Category Classification Challenge
Data Challenge Assignment

Fall 2025

1 Introduction

Welcome to your first machine learning data challenge! In this project, you will apply the machine learning techniques you've learned in class to solve a real-world classification problem.

1.1 The Problem

Imagine you work for a real estate analytics company that needs to automatically categorize office buildings into different quality tiers based on their characteristics. Your task is to build a machine learning model that can predict the **OfficeCategory** (ranging from 0 to 4, where 0 represents the lowest tier and 4 represents the highest tier) for office buildings based on various features such as:

- Physical characteristics (office space, plot size, number of floors)
- Quality indicators (building grade, condition)
- Amenities (parking spots, meeting rooms, restrooms)
- Location and zoning information
- And many more features!

This is a **multi-class classification problem** where you need to predict one of five categories (0, 1, 2, 3, or 4) for each office building.

1.2 Competition Details

This challenge is hosted on **Kaggle** as an in-class competition. To participate, you must:

1. Create a free Kaggle account if you don't have one
2. Register for the competition using this URL:
   https://www.kaggle.com/t/ea6f778790f8481bac58d98d25ff4b51

Important: You can submit predictions to Kaggle up to **5 times per day**, so use your submissions wisely!

2 Dataset Description

You are provided with the following files on the Kaggle competition page:

2.1 Training Data

- `office_train.csv`: Contains 35,000 office buildings with 79 features and the target variable `OfficeCategory`
- This is your main dataset for training and validating your models
- The target variable (`OfficeCategory`) takes values: 0, 1, 2, 3, or 4

2.2 Test Data

- `office_test.csv`: Contains 15,000 office buildings with the same 79 features but **without** the `OfficeCategory` column
- You will use your trained model to predict the `OfficeCategory` for these buildings
- Your predictions on this dataset determine your Kaggle leaderboard score

2.3 Baseline Template

- `template.ipynb`: A Jupyter notebook that provides a simple baseline solution
- This baseline uses Logistic Regression and achieves approximately 52% accuracy
- Use this as a starting point to understand the data and build better models!

2.4 Understanding the Features

The dataset contains 79 features that can be grouped into several categories:

1. **Size Features**: OfficeSpace, PlotSize, BasementArea, ParkingArea, etc.
2. **Quality Features**: BuildingGrade, BuildingCondition, ExteriorQuality, etc.
3. **Count Features**: MeetingRooms, Restrooms, ParkingSpots, TotalRooms, etc.
4. **Year Features**: ConstructionYear, RenovationYear
5. **Categorical Features**: ZoningClassification, BusinessDistrict, BuildingType, etc.

Note: Some features have missing values, which you'll need to handle appropriately!

3 Task and Evaluation

3.1 Your Mission

For each office building in the test set (`office_test.csv`), your model should predict its `OfficeCategory` (0, 1, 2, 3, or 4) based on the provided features.

3.2 Evaluation Metric

The evaluation metric for this competition is **Accuracy**. Accuracy measures the proportion of correct predictions:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} I(\hat{y}_i = y_i) \tag{1}$$

where:

- N is the total number of predictions (15,000 for the test set)
- $\hat{y}_i$ is your predicted category for building i
- y_i is the true category for building i
- $I(\hat{y}_i = y_i)$ equals 1 if your prediction matches the true label, 0 otherwise

Example: If you correctly predict 12,000 out of 15,000 buildings, your accuracy is:

$$\text{Accuracy} = \frac{12000}{15000} = 0.80 = 80\%$$

4 Getting Started

4.1 Step 1: Set Up Your Environment

You can work on this project using:

- **Google Colab** (recommended for beginners - free GPU access!)
- Jupyter Notebook on your local machine
- Any Python IDE (VSCode, PyCharm, etc.)

4.2 Step 2: Download the Data

1. Register for the competition on Kaggle
2. Download `office_train.csv`, `office_test.csv`, and `template.ipynb`
3. Upload these files to your working environment

4.3 Step 3: Explore the Baseline

Start by running `template.ipynb` to:

- Understand how to load and preprocess the data
- See a simple Logistic Regression model in action
- Generate your first submission file

4.4 Step 4: Improve Your Model

The template provides many ideas for improvement. Try different techniques and track your results!

5 Useful Python Libraries

Here are the key libraries you'll need:

5.1 Core Libraries

- **pandas**: Data manipulation and analysis
- **numpy**: Numerical computing
- **matplotlib / seaborn**: Data visualization

5.2 Machine Learning

- **scikit-learn**: ML algorithms, preprocessing, evaluation
  - Documentation: https://scikit-learn.org/
- **XGBoost**: Gradient boosting (often best for tabular data)
  - Documentation: https://xgboost.readthedocs.io/
- **LightGBM**: Fast gradient boosting
  - Documentation: https://lightgbm.readthedocs.io/
- **CatBoost**: Handles categorical features well
  - Documentation: https://catboost.ai/

5.3 Deep Learning (Optional)

- **PyTorch**: Deep learning framework
  - Documentation: https://pytorch.org/
- **TensorFlow / Keras**: Alternative deep learning framework
  - Documentation: https://www.tensorflow.org/

6 Team Formation and Grading

6.1 Team Requirements

- Teams must consist of **2-3 students**
- No solo work or teams larger than 3 will be accepted

6.2 Grading Breakdown (Total: 50% of module grade)

Your project grade consists of three components:

Component	Weight
Oral Presentation	30% (60% of project grade)
Written Report	10% (20% of project grade)
Kaggle Performance & Code	10% (20% of project grade)
Total Project Grade	**50%**

6.3 Kaggle Performance & Code (10%)

Requirements:

- Submit predictions to Kaggle (maximum 5 submissions per day)
- Your code must be reproducible - the teaching team should be able to:
  1. Run your code using only the provided data
  2. Train your model from scratch

3. Generate the exact predictions you submitted

Grading criteria:

- **Performance (50%)**:
  - Must exceed baseline (52% accuracy)
  - Higher accuracy = higher score
  - Top teams will receive bonus points
- **Code Quality (50%)**:
  - Clean, well-organized code
  - Clear comments explaining your approach
  - Code runs without errors
  - Reproducible results

What to submit (on Moodle):
A zipped folder named `TeamName_Code.zip` containing:

1. **code/** folder: All Python scripts or Jupyter notebooks
2. **README.txt**: Instructions on how to run your code
3. **requirements.txt**: List of required Python packages

6.4 Written Report (10%)

Format:

- Maximum 5 pages (excluding cover page and references)
- PDF format
- Include: Team name, student names, and Kaggle username(s)

Required sections:

6.4.1 Section 1: Data Exploration & Preprocessing (30%)

Describe your data analysis and preprocessing steps:

- What insights did you gain from exploring the data?
- How did you handle missing values? Why?
- How did you handle categorical features?
- Did you identify any outliers or anomalies?
- Include visualizations (distribution plots, correlation heatmaps, etc.)

6.4.2 Section 2: Feature Engineering (35%)

Explain the features you created:

- What new features did you create and why?
- What was the motivation/intuition behind each feature?
- Did you perform feature selection? How?
- Show the impact of feature engineering on performance
- Example table:

Feature Set	Validation Accuracy
Original features only	52.0%
+ Improvement 1	58.5%
+ Improvement 2	64.2%
+ Improvement 3	68.7%

6.4.3 Section 3: Model Selection & Tuning (35%)

Describe your modeling approach:

- What models did you try? Why?
- Compare performance of different models
- Explain your hyperparameter tuning process
- How did you prevent overfitting?
- Present a comparison table of your models

Example comparison table:

Model	Train Acc.	Val. Acc.	Test Acc.
Logistic Regression			
Random Forest			
XGBoost			
XGBoost (tuned)			
Ensemble			

6.5 Oral Presentation (30%)

Schedule:

- Week **14**
- Duration: 15 minutes presentation + 5 minutes Q&A

Requirements:

- Both team members must be present

- Both team members must speak and contribute
- Prepare slides (PowerPoint, Google Slides, or PDF)

Suggested structure:

1. **Introduction (1 minute)**
    - Team members and roles
    - Problem overview
    - Your final results
2. **Data Exploration & Insights (2 minutes)**
    - Key findings from data exploration
    - Important visualizations
    - Challenges encountered
3. **Feature Engineering (2 minutes)**
    - Most impactful features you created
    - Why they worked
    - Show performance improvement
4. **Model Development (3 minutes)**
    - Models you tried
    - Comparison of results
    - Hyperparameter tuning approach
    - Your final model architecture
5. **Results & Conclusions (2 minutes)**
    - Final Kaggle performance
    - What worked best?
    - What didn't work?
    - Key learnings
    - What would you do with more time?

Grading criteria:

- **Content (50%)**: Technical correctness, depth of analysis, clarity
- **Presentation skills (30%)**: Clear communication, time management, visual aids
- **Understanding (20%)**: Ability to answer questions, demonstrate understanding

7 Important Rules & Deadlines

7.1 Academic Integrity

- You may use online resources and research papers, but you **must cite them**
- You may discuss ideas with other teams, but your code and report must be your own
- Copy-pasting code from others will result in **zero marks** for the entire project
- Using external datasets with labels is **strictly prohibited**

7.2 Submission Process

7.2.1 Kaggle Submissions

- Format: CSV file with two columns: `Id`, `OfficeCategory`
- Use your final best submission for grading

Example submission format:

```
Id,OfficeCategory
0,3
1,2
2,4
3,1
...
```

7.2.2 Moodle Submission

Submit a single ZIP file named `TeamName_Project.zip` containing:

- **code/** folder with all your code files
- **README.txt** with instructions
- **requirements.txt** with Python package versions
- **Report.pdf** (your 5-page report)

8 Frequently Asked Questions

8.1 General Questions

Q: Can I use external data?
A: No, you should only use the provided `office_train.csv` and `office_test.csv` files. Using external datasets with labels is prohibited and will result in penalties.

Q: Can I use external libraries?
A: Yes! You can use any publicly available Python libraries (scikit-learn, XGBoost, PyTorch, etc.) and standard preprocessing techniques. Just make sure to cite them in your report.

Q: How many times can I submit to Kaggle?
A: Maximum 5 submissions per day. Plan your submissions strategically!

Q: What if my code doesn't run on the first try?
A: That's normal! Debug locally before submitting. Make sure your code runs on a fresh environment.

8.2 Technical Questions

Q: My validation accuracy is much lower than training accuracy. What's wrong?
A: You're likely overfitting. Try:

- Reducing model complexity (lower max_depth, fewer trees)
- Using regularization
- Adding more data augmentation

- Using cross-validation

Q: How do I handle missing values?
A: Common approaches:

- Numeric features: median or mean imputation
- Categorical features: mode imputation or create "Missing" category
- Advanced: KNN imputation or model-based imputation

Q: Should I use One-Hot Encoding or Label Encoding for categorical features?
A: It depends:

- Tree-based models (Random Forest, XGBoost): Label Encoding works fine
- Linear models: One-Hot Encoding is better
- Try both and see what works better for your model!

Q: My code is too slow. What can I do?
A: Tips for faster training:

- Use `n_jobs=-1` to use all CPU cores
- Reduce the parameter search space in GridSearchCV
- Use RandomizedSearchCV instead of GridSearchCV
- Sample a subset of data for initial experiments
- Use Google Colab's GPU (if using neural networks)

8.3 Submission Questions

Q: What format should my submission file be in?
A: CSV format with two columns:

- `Id`: Integer from 0 to 14999
- `OfficeCategory`: Your prediction (0, 1, 2, 3, or 4)

Q: Do I need to submit all my experiments?
A: No, just submit:

- Your final best model code
- Code that can reproduce your Kaggle submission
- Your report explaining all approaches you tried

Q: Can I update my report after the deadline?
A: No, late submissions will not be accepted. Plan ahead!

9 Resources and Further Reading

9.1 Online Courses (Free)

- **Kaggle Learn**: https://www.kaggle.com/learn
  - Intro to Machine Learning
  - Intermediate Machine Learning
  - Feature Engineering
- **Fast.ai**: https://www.fast.ai/
  - Practical Deep Learning

9.2 Documentation

- Scikit-learn User Guide: https://scikit-learn.org/stable/user_guide.html
- XGBoost Tutorials: https://xgboost.readthedocs.io/en/stable/tutorials/index.html
- Pandas Documentation: https://pandas.pydata.org/docs/

9.3 Kaggle Competitions to Learn From

Look at kernels/notebooks from these beginner-friendly competitions:

- Titanic: Machine Learning from Disaster
- House Prices: Advanced Regression Techniques
- Digit Recognizer

9.4 Books (Optional)

- *Hands-On Machine Learning* by Aurélien Géron
- *Introduction to Statistical Learning* by James et al. (Free PDF)
- *Python Data Science Handbook* by Jake VanderPlas (Free online)

10 Final Checklist

Before your final submission, make sure:

10.1 Code Submission

- Code runs without errors on a fresh environment
- README.txt contains clear instructions
- requirements.txt lists all packages and versions
- Code is well-commented
- Can reproduce your Kaggle submission
- All files are in the correct folder structure

10.2 Report

- Maximum 5 pages (excluding cover and references)
- PDF format
- Team name and student names on cover page
- All three required sections included
- Figures and tables are clear and labeled
- References cited properly
- Proofread for typos and clarity

10.3 Presentation

- Presentation slot booked
- Slides prepared (10 minutes of content)
- Both team members have speaking parts
- Practiced timing
- Prepared for Q&A
- Backup plan if technology fails

10.4 Kaggle

- Made at least one successful submission
- Final submission selected
- Kaggle username matches team registration

11 Good Luck!

This project is your opportunity to apply everything you've learned in class to a real-world problem. Don't be discouraged if things don't work perfectly at first - machine learning is an iterative process!

Remember:

- Start early
- Experiment often
- Learn from failures
- Ask for help when stuck
- Have fun with it!

The teaching team wishes you the best of luck with this challenge!