Predicting Learning Commons Usage: Duration and Occupancy

A Non-Linear Approach

Naiyue Liang (Emma), Ryan Renken, Jaryt Salvo, & Jason Turk

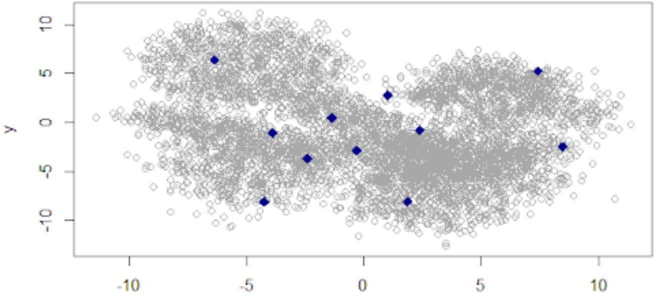
MATH 7560 Statistical Learning II || BGSU

April 24, 2025

K-means: Data & Motivation

Clustering Task

- **Goal**: Partition $\mathbb{R}^2$ data into $k = 11$ clusters.
- **Comparison**: Standard K-means vs K-means++ initialization.
- **K-means++ Motivation**: Better initial centroids for potentially faster/better clustering.



K-means vs K-means++ Results

Standard K-means:

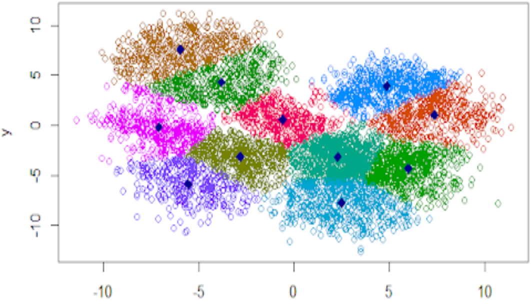
- **Stats**: 5 iterations, WCSS = 22,824.

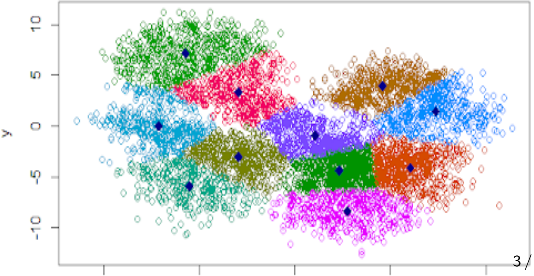
K-means++ Initialization:

- **Stats**: 8 iterations, WCSS = 22,943.

Observation

K-means++ offered no clear advantage over standard K-means for this dataset (visually or by WCSS).

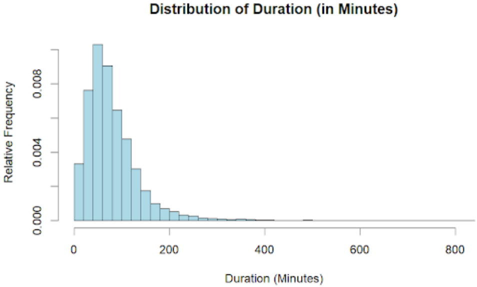




Distribution Statistics

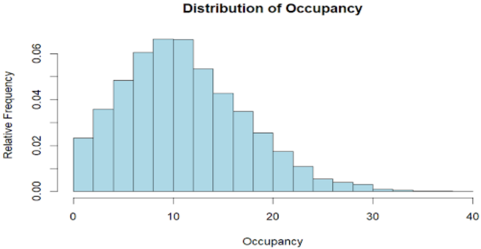
Duration (minutes):

Statistic	Value
Minimum	6.00
1st Quartile	44.00
Median	68.00
Mean	81.78
3rd Quartile	103.00
Maximum	822.00



Occupancy (students):

Statistic	Value
Minimum	1.00
1st Quartile	7.00
Median	11.00
Mean	11.62
3rd Quartile	15.00
Maximum	40.00



Feature Categories Overview

Category	Key Features
Temporal	Time of day, Day of week, Week of semester
Academic	Course level, GPA categories, Credit load
Visit	Duration patterns, Group sizes, Visit frequency
Course	Subject areas, Level progression, Course mix
Student	Major groups, Class standing, Academic progress

External Source	Key Features
R library ‘lunar‘	Moon phase data
R library ‘openmeteo‘	Hourly weather metrics (temperature, humidity, pressure, cloud cover, wind, radiation, precipitation, & soil conditions)

Dropped Raw Features

Raw Feature	**Engineered Into**
Student_IDs	Total_Visits, Semester_Visits, Avg_Weekly_Visits
Class_Standing	Class_Standing_Self_Reported, Class_Standing_BGSU
Major	Major_Category, Has_Multiple_Majors
Expected_Graduation	Expected_Graduation_Date, Months_Until_Graduation
Course_Name	Course_Name_Category
Course_Number	Unique_Courses, Course_Level_Mix
Course_Type	Course_Type_Category
Course_Code_by_Thousands	Course_Level, Advanced_Course_Ratio

Feature Engineering Strategy

Raw features were transformed into more informative derived features, capturing higher-level patterns and relationships in the data.

Hyperparameters Tuned (Both Tasks)

Model	Hyperparameters Tuned
MARS	Number of terms, Product degree
Random Forest	Number of trees, Minimum node size, Number of variables tried
XGBoost	Number of trees, Tree depth, Learning rate, Minimum node size, Number of variables tried
GRU	Learning rate, Batch size, Hidden dimension, Number of layers, Expansion factor, Dropout rate, Weight decay, Activation function
Transformer	Learning rate, Batch size, Model dimension, Number of heads, Number of layers, Hidden dimension, Dropout rate, Weight decay

Preprocessing Pipeline Details

R ‘recipes‘ Pipeline Steps

1. Define roles (outcome, predictors)
2. Remove specified ID/date/unwanted columns
3. Convert `Check_In_Time` to minutes past midnight
4. Impute missing numerics (mean)
5. Handle novel factor levels
6. Create dummy variables (drop first)
7. Remove zero-variance predictors
8. Normalize numeric predictors

Top Model Performance (Holdout Set)

Duration Task

Model	RMSE	R²
XGBoost	59.9	0.099
Random Forest	60.1	0.090
Transformer	61.3	0.010
MARS	61.6	0.045
GRU	63.8	0.041

Occupancy Task

Model	RMSE	R²
XGBoost	1.83	0.911
Random Forest	1.93	0.902
GRU	3.16	0.738
Transformer	3.26	0.706
MARS	3.78	0.617

Key Performance Observations

Based on holdout set metrics, XGBoost performed best for both tasks.

Best Model Configurations

Duration Prediction:

Component	**Value**
Model	XGBoost
CV Method	5-fold
RMSE	59.9
R^2	0.099
Trees	75
Tree Depth	21
Learning Rate	0.05
Min Node Size	15
Variables Tried (mtry)	15

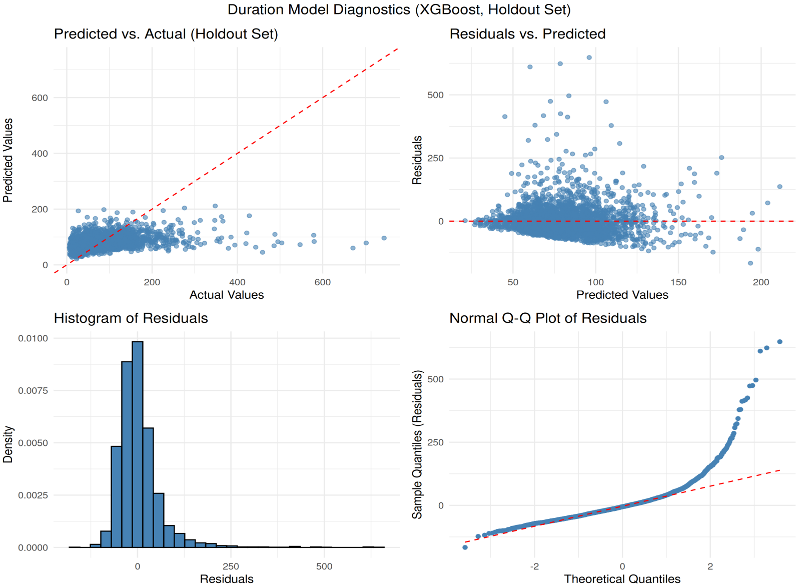
Occupancy Prediction:

Component	**Value**
Model	XGBoost
CV Method	5-fold
RMSE	1.83
R^2	0.911
Trees	450
Tree Depth	8
Learning Rate	0.1
Min Node Size	2
Variables Tried (mtry)	35

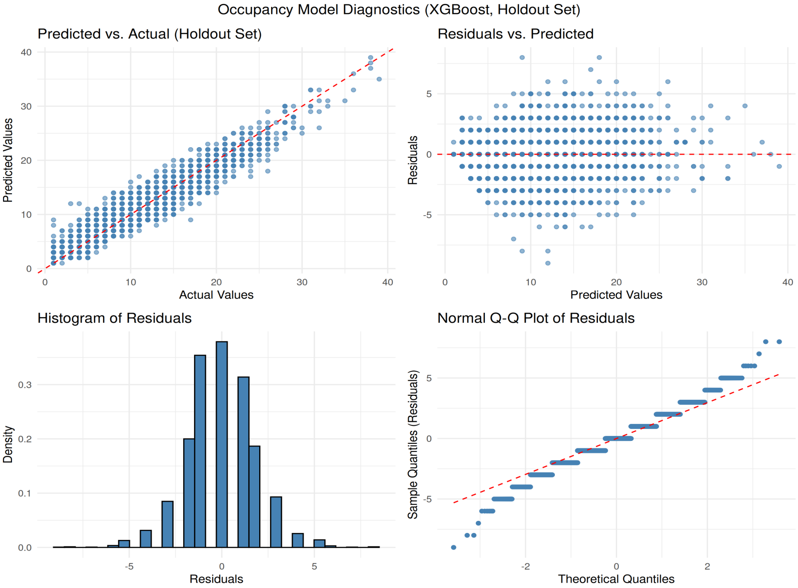
Key Insight

XGBoost provided the best performance for both tasks using 5-fold cross-validation. Final performance was boosted by using a **weighted average** of predictions (weight = 0.75) and training set duration means (weight = 0.25).

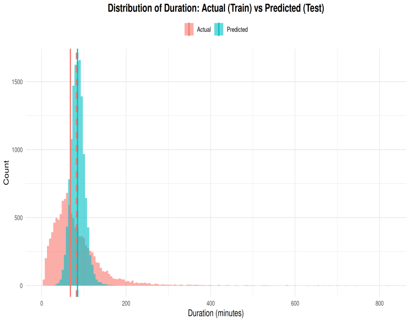
Duration: Best Model Diagnostics

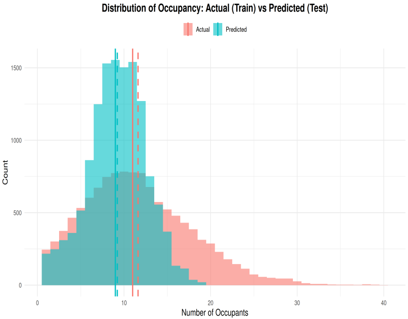


Occupancy: Best Model Diagnostics

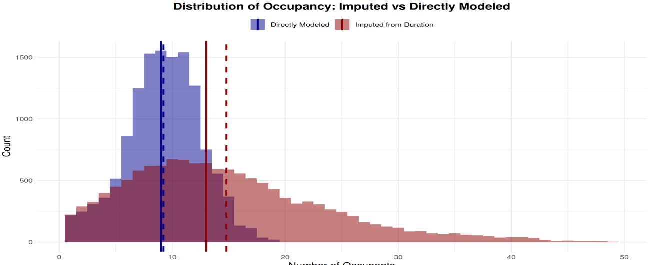


Sanity Check


Distribution of Duration: Actual (Train) vs Predicted (Test)


Distribution of Occupancy: Actual (Train) vs Predicted (Test)

Dashed lines: means; Solid lines: medians


Distribution of Occupancy: Imputed vs Directly Modeled

Key Findings

Main Results

1. **XGBoost** achieved the best performance for both tasks. Weighting further boosted performance.
2. **Occupancy** prediction was highly successful on our training data(Holdout $R^2 = 0.91$).
3. **Duration** prediction improved with a weighted average approach, but remains challenging (Holdout $R^2 = 0.10$).
4. **K-means++** initialization did not show a clear advantage over standard K-means for the sample dataset presented.

Impact

The Occupancy model demonstrates strong predictive power. Duration prediction, while improved, highlights the difficulty of modeling individual student behavior.

Thank You

For Your Attention

Questions & Discussion Welcome