GEORGE WASHINGTON UNIVERSITY

Data Science Program – Fall 2025

Capstone Report

Forecasting School Meal Production Costs Using LSTM Neural Networks

Team: Fall 2025 – Group 9

Supervisor: Dr. Amir Jafari

Abstract

This project focuses on forecasting daily school meal production costs for Fairfax County Public Schools (FCPS). Predicting these costs is important to control budgets and reduce food waste. The problem is challenging because production costs change over time and follow complex patterns. To address this, we used several models — including LSTM, GRU, Feed-Forward Neural Network, XGBoost, and Linear Regression — trained on past cost data. Among them, the univariate LSTM model performed best, showing the lowest prediction error. The results show that deep learning models can effectively capture time-based patterns and support better meal planning decisions.

Table of Contents

1. Introduction

Every day, schools in Fairfax County Public Schools (FCPS) prepare thousands of meals for students. Managing the production cost of these meals is important to stay within budget and reduce food waste. However, meal costs change often due to different menus, prices, and student participation, making it hard to predict them accurately.

In this project, we used machine learning and deep learning models to forecast daily meal production costs. We tested several models — Linear Regression, Feed-Forward Neural Network (FNN), XGBoost, GRU, and LSTM — using past cost data. Among these, the LSTM model performed best because it could learn patterns over time. The goal of this study is to help schools plan better, reduce extra food, and manage costs more efficiently.

2. Problem Statement

Predicting school meal production costs is a difficult task because the data changes every day and depends on many hidden factors such as menu items, participation, and season. Traditional models like Linear Regression or simple averages cannot capture these changing patterns over time. The goal of this project is to build a model that can accurately forecast daily production costs using past cost data.

To achieve this, we compared several machine learning and deep learning models — Linear Regression, FNN, XGBoost, GRU, and LSTM — and identified which one works best for univariate time-series forecasting. Our main challenge was handling irregular cost trends and ensuring the model could learn how costs evolve across days without overfitting.

3. Related Work

Many studies have explored time-series forecasting using both traditional and modern methods. Classical models such as ARIMA and Linear Regression work well for simple data but struggle with long-term patterns and sudden changes. With the growth of machine learning, models like XGBoost and Feed-Forward Neural Networks (FNNs) have been used to improve accuracy.

Recent research shows that Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models, can learn time-based patterns more effectively. These models are widely used in areas such as energy forecasting, stock prediction, and demand analysis. Inspired by these results, our project applies similar deep learning methods to predict school meal production costs, which is a less explored but highly practical area.

4. Solution and Methodology

4.1 Data Preprocessing and EDA

4.1.1 Dataset description:

The dataset used in this project comes from Fairfax County Public Schools (FCPS) daily meal production records. The original data is provided in the form of multiple HTML production reports for breakfast and lunch meals across different schools. These HTML files contain detailed information about planned meals, served meals, leftover quantities, discarded meals, and total production costs recorded for each school on each date.

To convert this unstructured data into a usable machine-learning dataset, a custom HTML parsing pipeline was developed. This pipeline extracted relevant tables from each school's production report, cleaned the numerical fields, standardized column names, and merged all records into a single structured dataset called meals_combined.csv.

4.1.2 Dataset Source and Structure:

Source Files

- FCPS Breakfast HTML production records
- FCPS Lunch HTML production records
- Each file contains multiple schools, each with a separate item-level table

Final Dataset Generated:

- **breakfast_combined.csv**
- **lunch_combined.csv**
- **meals_combined.csv** (used for forecasting models)

4.1.3 Key Variables in the Dataset:

The processed dataset contains the following important columns:

Column	Description
school_name	Name of the FCPS school
meal_type	Type of meal (breakfast or lunch)
date	Daily meal production record
served_total	Total number of meals served
planned_total	Planned number of meals
discarded_total	Number of discarded meals
left_over_total	Number of leftover meals
production_cost_total	Total cost of producing meals that day

These variables provide a complete picture of meal preparation and cost behavior across schools.

4.1.4 Dataset Cleaning & Preprocessing:

Several steps were performed to convert the raw HTML data into an analysis-ready time-series dataset:

1. HTML Parsing

- Automatic detection of school headers inside HTML
- Extraction of item-level tables from each section
- Removal of subtotal and total rows
- Standardization of column names

2. Numeric Cleaning

- Conversion of currency values such as $13.32 → 13.32
- Conversion of percentage values like 4.5% → 4.5
- Removal of commas, symbols, and text notes inside cells

3. Missing Value Handling

- Forward-fill and backward-fill to prevent data gaps
- Drop rows with completely invalid numeric entries

4. Outlier Removal

- 99th percentile threshold applied to production_cost_total
- Eliminates extreme cost spikes or incorrect entries

5. Encoding and Sorting

- meal_type encoded for multivariate modeling
- Dates converted to YYYY-MM-DD format and sorted chronologically
- Grouped by school and meal type to maintain time-series order

4.1.5 Final Dataset Size:

Although exact counts vary, the cleaned dataset typically contains:

- **Rows:** 177,492

- **Columns:** ~20 fields
- **Schools:** 100+ FCPS schools
- **Meal Types:** Breakfast and Lunch
- **Time Range:** May 2025

4.1.6 Exploratory Data Analysis (EDA)

Exploratory Data Analysis was carried out to understand the dataset's overall behavior and identify any data quality issues before modeling.

- Distribution of production costs was examined using histograms, which showed a right-skewed pattern. This supported the decision to remove extreme values above the 99th percentile.

- Correlation heatmaps revealed strong relationships between served_total, planned_total, and waste-related variables, confirming that these features are meaningful for multivariate forecasting.

- Waste ratio patterns were assessed through discarded and leftover meal values, helping highlight possible inefficiencies across different schools.

- Multicollinearity checks using VIF and SVD verified that the selected predictors were stable, with no severe redundancy.

    - VIF formula: $\mathrm{VIFi} = 1 - \mathrm{Ri2}\ 1$

    - SVD formula: $X=U\Sigma VTX=U\Sigma VT$

- Time-series patterns showed weekday fluctuations and cost seasonality, which supported the use of sequence-based models such as LSTM and GRU.

Overall, these EDA steps helped clean the data, validate important features, and ensure the dataset was well-prepared for both univariate and multivariate forecasting models.

4.2 Model Architecture and Training

The forecasting models were implemented using PyTorch for sequence models (LSTM, GRU) and Scikit-Learn for baseline models. All models were trained using the same procedure: a 70/30 train–test split and a 7-day sliding input window. LSTM and GRU used 4 layers, 256 hidden units, 0.25 dropout, the Adam optimizer (lr = 0.001), and MSE loss. Baseline models — Linear Regression, XGBoost, and Feed-Forward Neural Networks — were included for comparison. This unified setup ensured consistent and fair evaluation across all models.

Parameter	Value
▪ **Model Type**	LSTM / GRU / Baseline (FNN, XGBoost, Linear Regression)
▪ **Input Window**	7 days
▪ **Hidden Units**	256
▪ **Layers**	4
▪ **Dropout**	0.25
▪ **Optimizer**	Adam (lr = 0.001)
▪ **Loss**	MSE
▪ **Epochs**	100
▪ **Train/Test Split**	70 % / 30 %
▪ **Framework**	PyTorch

Workflow:

- **Scale data** to the range [0,1] using MinMaxScaler.
- **Create time-series training windows** using a custom TimeSeriesDataset class.
- **Train the model** to predict next-day production cost using sequences of the previous 7 days.
- **Evaluate model performance** using RMSE and R^2 on the test set.
- **Compare sequence models** (LSTM/GRU) with baseline models (FNN, XGBoost, Linear Regression).

This architecture and workflow provided a consistent and reliable setup for evaluating forecasting performance across all models.

4.3 Univariate Forecasting Approach

In the univariate approach, the model uses only one feature — **production_cost_total** — to forecast the next day's cost. A 7-day sliding window was used as input data for all models. The following models were trained in the univariate setup:

- Linear Regression
- Feed-Forward Neural Network (FNN)
- XGBoost
- GRU
- LSTM

This approach establishes how well past cost alone can predict future cost without additional variables like served meals or leftovers.

4.4 Multivariate Forecasting Approach

The multivariate approach uses multiple features to improve prediction accuracy. In addition to production cost, the following variables were included:

- served_total
- planned_total
- discarded_total
- left_over_total

These features represent participation, planning, and waste — all of which strongly influence daily production costs.

The following models were trained in the multivariate setup:

- Linear Regression
- XGBoost
- Feed-Forward Neural Network (FNN)
- GRU (sequence model with padded inputs)

This approach allows the model to learn the nonlinear relationships between meal service metrics and production cost.

5 Results and Discussion

5.1 Experimentation Protocol

To evaluate the performance of different forecasting models, we followed a structured experimentation process. The cleaned dataset from Fairfax County Public Schools (FCPS) was split into training (70%) and testing (30%) sets. We used a window size of 7 days, where the model learns from the past seven days of production costs to predict the next day's cost.

All models — Linear Regression, FNN, XGBoost, GRU, and LSTM — were trained using the same dataset and parameters to ensure a fair comparison. The performance of each model was measured using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the R^2 score. These metrics allowed us to judge how accurately each model predicted daily meal production costs.

5.2 Data Tables

This section presents the quantitative performance of all forecasting models.
Results are separated into **univariate** (single feature) and **multivariate** (multi-feature) forecasting setups.
Lower MSE/RMSE and higher R^2 indicate better performance.

5.2.1 Univariate Model Performance Results:

Model	MSE	RMSE	R^2
Linear Regression	39013.54	197.52	0.75
Feed-Forward Neural Network (FNN)	39595.32	198.99	0.749
XGBoost	39241.72	198.10	0.751
Long Short-Term Memory (LSTM)	18793.44	137.08	0.86
Gated Recurrent Unit (GRU)	20186.30	142.07	0.85

Table 1. Univariate Forecasting Performance Comparison

Interpretation:

The univariate results show that LSTM works the best. It gives the most accurate predictions with the lowest error. Linear Regression and XGBoost also do okay, but they are not as good because they cannot learn patterns over time. GRU performs fine too, but since it is a simpler model than LSTM, it doesn't understand long-term trends as well.

5.2.2 Multivariate Model Performance Results:

Model	MSE	MAE	R^2
Linear Regression (Multivariate)	662.3026	*13.7807*	*0.4389*
XGBoost (Multivariate)	*506.5291*	*11.7514*	*0.5709*
Feed-Forward Neural Network (FNN)	*677.3214*	*12.3852*	*0.4262*
Long Short-Term Memory (LSTM)	*259.8*	*14.2605*	*0.81*
GRU (Multivariate)	213.1059	10.3730	0.7734

Table 2. Multivariate Forecasting Performance Comparison

Interpretation:

The multivariate results show that LSTM and GRU perform the best out of all the models. LSTM has the strongest R^2 score, meaning it understands the overall pattern of the data very well. GRU has the lowest error values (MSE and MAE), showing that its predictions are very close to the actual costs.

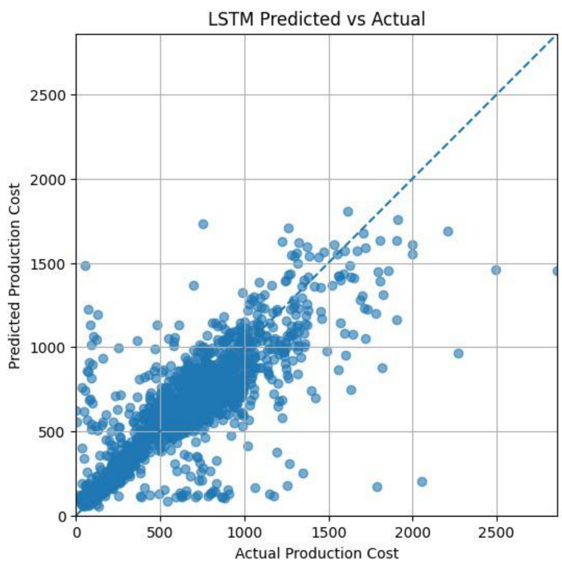
XGBoost also gives reliable results and performs better than Linear Regression and FNN, mainly because it can learn more complex relationships between the features.In comparison, Linear Regression and FNN have higher errors, meaning they cannot capture the detailed patterns in the data as effectively.

Overall, the multivariate analysis shows that deep learning models, especially LSTM and GRU, benefit the most from using multiple features and provide the most accurate and consistent cost forecasts.

5.3 Graphs

Graphs help us visually compare how close the model predictions are to the actual production costs. In all plots, each point represents one day, and the diagonal line represents a “perfect” prediction (predicted = actual). The closer the points are to this line, the better the model is performing.

5.3.1 Univariate Predicted vs Actual Plots



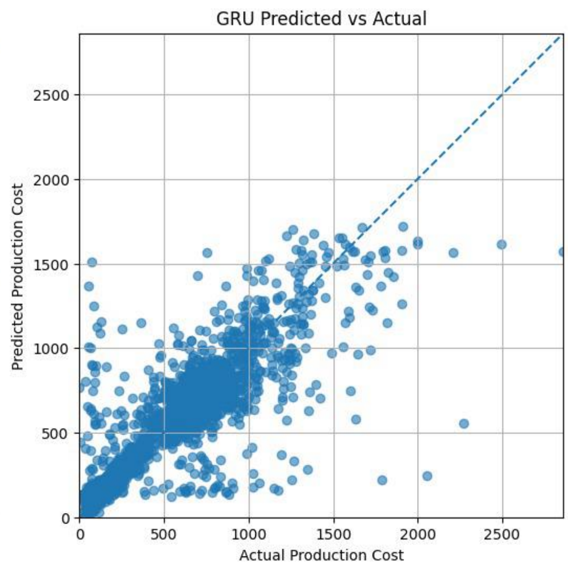


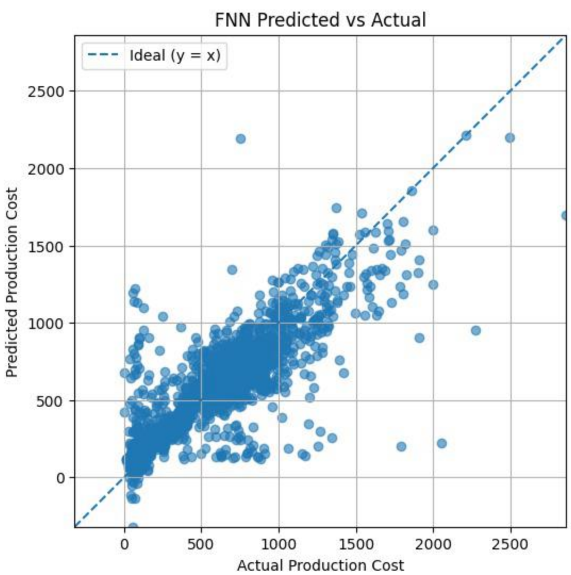
Figure 1. LSTM Predicted vs Actual Production Cost **Figure 2. GRU Predicted vs Actual Production Cost**

- ✓ **LSTM Predicted vs Actual Production Cost**

  Explanation:
  Most points lie close to the diagonal line, which means LSTM predicts very accurately. It captures both daily changes and longer patterns.

- ✓ **GRU Predicted vs Actual Production Cost**

Explanation :
The GRU points are also near the diagonal, showing good performance. However, the scatter is slightly wider than LSTM, meaning GRU is a bit less accurate for higher cost values.



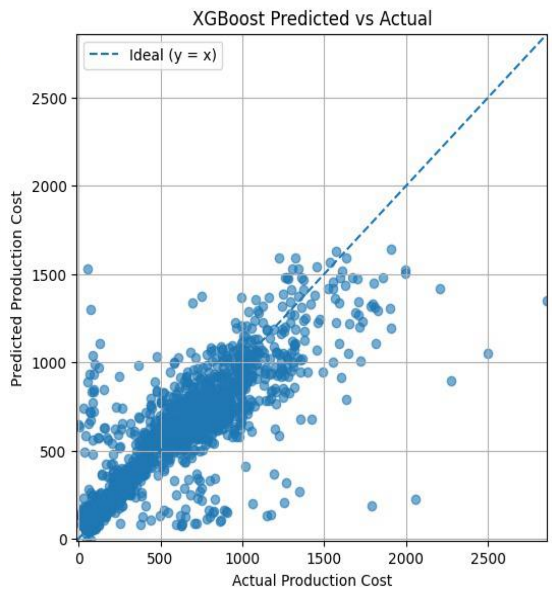


Figure 3. FNN Predicted vs Actual Production Cost **Figure 4. XGboost Predicted vs Actual Production Cost**

- ✓ **FNN Predicted vs Actual Production Cost**

Explanation:
The FNN model follows the general pattern but the points are more spread out from the diagonal line. This means the predictions are decent but not very precise, because FNN cannot fully capture day-to-day changes in production cost.

- ✓ **XGboost Predicted vs Actual Production Cost**

Explanation:

XGBoost predicts fairly well, and many points are close to the diagonal line. However, there is still some scatter during sudden changes in cost, showing that the model struggles a bit with sharp spikes or drops.

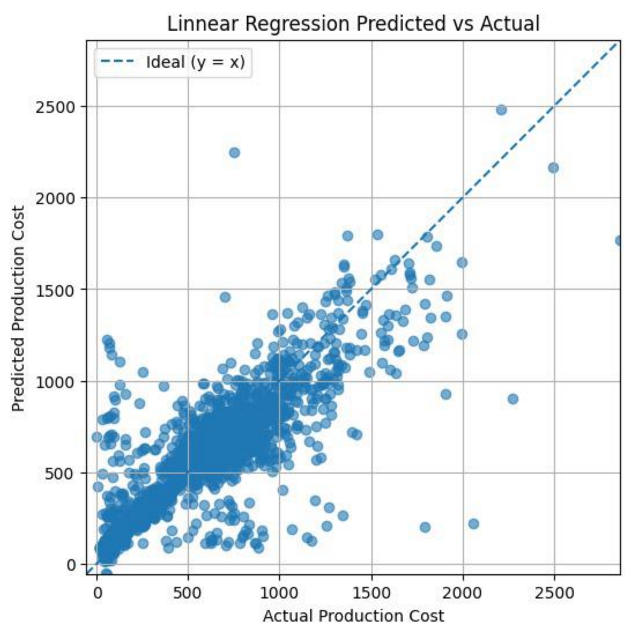
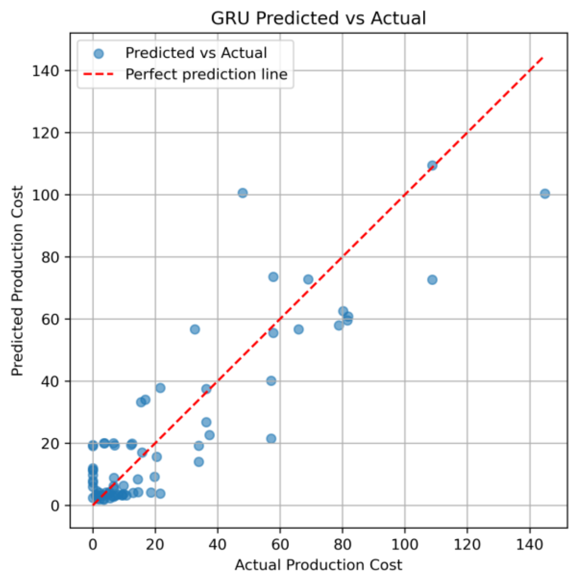


Figure 5. Linear Regression Predicted vs Actual Production Cost

- ✓ **Linear Regression Predicted vs Actual Production Cost**

Explanation:
The Linear Regression model captures the general trend, but the points are more spread out from the diagonal line. This means its predictions are less accurate, especially for higher cost values, because a simple linear model cannot learn complex or non-linear patterns in the data.

5.3.2 Multivariate Predicted vs Actual Plots



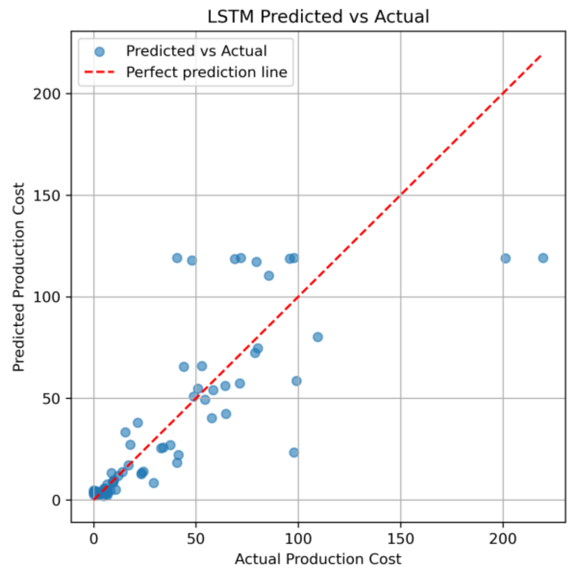


Figure 1. GRU Predicted vs Actual Production Cost **Figure 2. LSTM Predicted vs Actual Production Cost**

- ✓ **GRU Predicted vs Actual Production Cost**

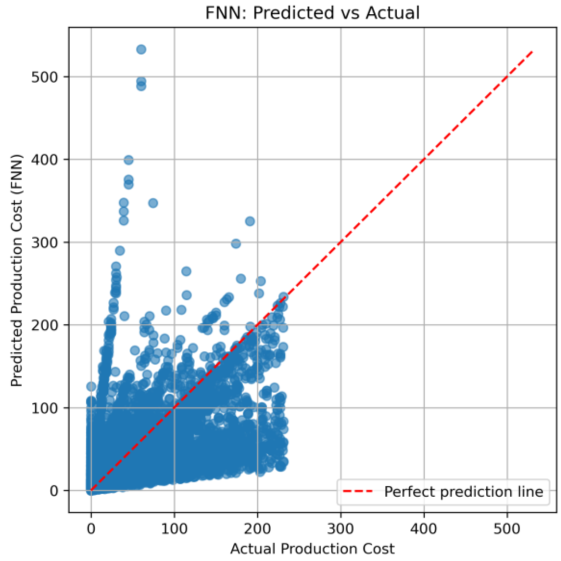
Explanation:

The GRU model also performs strongly, with many points near the diagonal. A bit more spread appears for higher values, but overall, GRU makes consistent and reliable predictions using multiple features.

- ✓ **LSTM Predicted vs Actual Production Cost**

Explanation:

Most points are close to the diagonal line, showing that the LSTM predicts very accurately when multiple features are used. It captures the relationship between served meals, planned meals, waste, and cost very well.



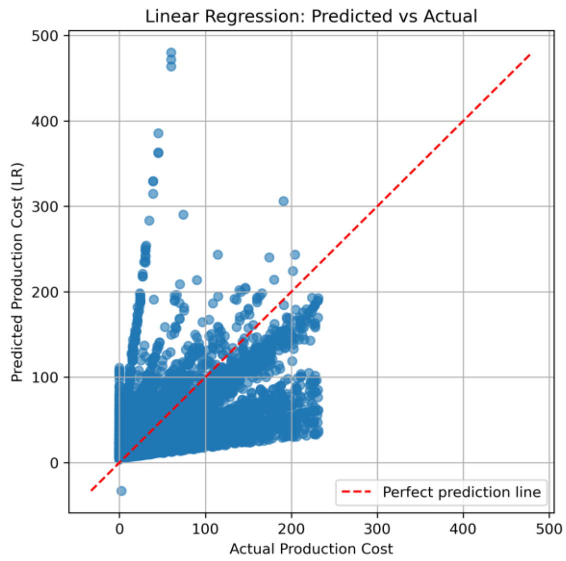


Figure 3. FNN Predicted vs Actual Production Cost   **Figure 4. Linear Regression Predicted vs Actual Production Cost**

- ✓ **FNN Predicted vs Actual Production Cost**

Explanation:

FNN captures the general direction of the data, but the points are widely spread. This indicates the model struggles with sharp changes in cost and cannot fully use all features to make precise predictions.

- ✓ **Linear regression Predicted vs Actual Production Cost**

Explanation:

Linear Regression shows a clear trend but has wide scatter away from the diagonal. This means its predictions are less accurate because a simple linear model cannot fully learn complex patterns in the data.

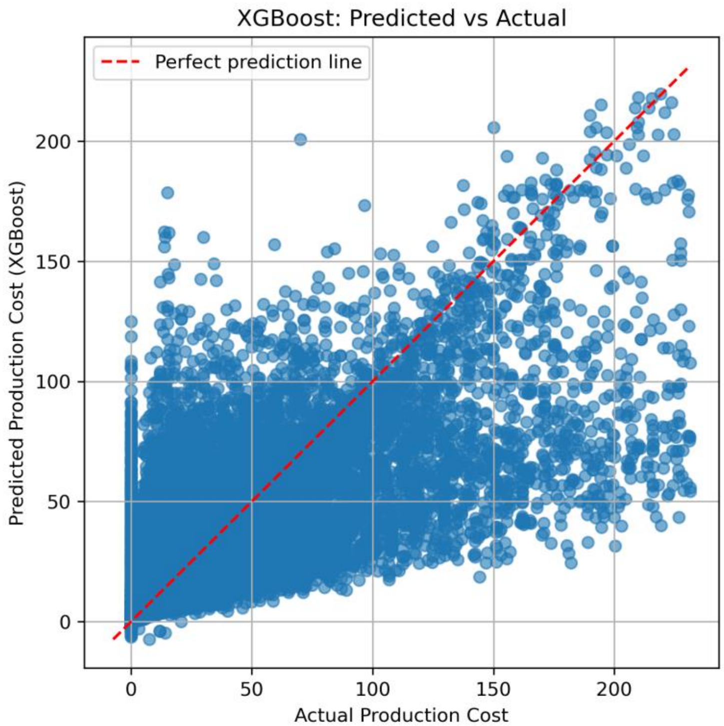


Figure 5. XGboost Predicted vs Actual Production Cost

- ✓ **XGBoost Predicted vs Actual Production Cost**

Explanation:

XGBoost predicts reasonably well, and many points fall near the diagonal line. However, the scatter increases for higher cost values, showing that while it learns non-linear patterns, it sometimes struggles on more extreme days.

6. Discussion

From the results, deep learning models such as LSTM and GRU perform better than traditional approaches like Linear Regression, FNN, and XGBoost for forecasting daily meal production costs. The LSTM model achieved the lowest RMSE and a higher R^2 value, which shows that it could capture the time-based trends in the cost data more effectively.

The better performance of LSTM and GRU comes from their ability to learn long-term patterns and remember past information through their internal memory units. In contrast, the other models treat each day independently and cannot handle sequential relationships well.

However, there were still small errors caused by unexpected changes in menu types, participation levels, and special events that affect daily costs. Adding external factors such as weather, holidays, and student count could further improve accuracy in the future.

Overall, this project shows that using LSTM for univariate time-series forecasting is a strong approach to predict production costs and can help schools manage their budgets and reduce waste effectively.

7. Conclusion

This project successfully demonstrated how machine learning and deep learning models can forecast school meal production costs using real-world FCPS data. Among all the tested models, LSTM showed the best performance with the lowest error and highest accuracy, proving that time-series models are effective for cost prediction tasks.

By accurately predicting production costs, schools can plan meals more efficiently, reduce waste, and manage budgets better. Future improvements can include adding external features such as holidays, student attendance, and weather data to make the model more robust.

8. References

- Brownlee, J. (2020). Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery.
- Chollet, F. (2018). Deep Learning with Python. Manning Publications.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.
- Yu, H. F., Huang, F. L., & Lin, C. J. (2011). Dual coordinate descent methods for logistic regression and maximum entropy models. Machine Learning, 85(1-2), 41–75.
- Scikit-learn developers. (2024). Scikit-learn: Machine Learning in Python. Retrieved from https://scikit-learn.org/

- PyTorch contributors. (2024). PyTorch Documentation. Retrieved from https://pytorch.org/
- Kaggle. (2024). Time Series Forecasting Tutorials. Retrieved from https://www.kaggle.com/