ISE 5984 Final Project Report

Demand Forecasting with Machine Learning

Atharva Shailesh Anchalwar

December 12, 2024

Introduction:

Demand forecasting is an essential tool that helps businesses predict future requirements, maximize available resources, and enhance decision-making. "Forecasting is the process of estimating the unknown, typically future outcomes, using historical data and information," according to Fildes and Hastings. Using what is known to prepare for what is yet to come is the heart of forecasting.

Demand forecasting is essential in sectors like retail and energy. Precise forecasts enable companies to:

- Effectively distribute resources.
- Arrange production and inventory timetables.
- Reduce expenses while increasing client happiness.

In order to show the adaptability and strength of predictive modeling, three datasets—Energy Demand, Walmart Sales, and Rossmann Sales—were examined using machine learning approaches.

Essential Ideas in Forecasting

1. **Qualitative Inputs**: These comprise intangible elements such as macroeconomic patterns, holidays, and professional judgments.

2. **Quantitative Inputs**: These factors that can be calculated are separated into:

    - Causal Forecasting: Investigates factors that affect departures from past patterns.
    - Regular projections based on past trends are the main focus of time series forecasting.

Overview of Data

- **Energy Demand**: Suitable for sophisticated models such as LSTM, it displayed hourly trends and temporal dependencies with substantial seasonality.
- **Walmart Sales**: Showed seasonality and promotional effects driven by holidays, necessitating models that could handle time series and categorical data.
- **Rossmann Sales**: included data from several stores with elements like promotions and competition, requiring customized models for every business.

Aspect	LSTM	ARIMA	XGBoost	Random Forest
Model Type	Deep Learning (Recurrent Neural Network)	Statistical Time Series	Gradient Boosting Decision Tree	Ensemble of Decision Trees
Data Requirement	Requires a large dataset for effective training	Can work with smaller datasets	Requires a moderate to large dataset for optimal performance	Requires a moderate to large dataset for optimal performance
Handling Non-linearity	Suitable for capturing complex, non-linear relationships	Assumes linearity in the time series	Suitable for capturing complex, non-linear relationships	Suitable for capturing complex, non-linear relationships
Seasonality & Trend	Can automatically learn seasonality and trend	Requires manual differencing to handle seasonality and trends	Cannot inherently model seasonality; requires feature engineering	Cannot inherently model seasonality; requires feature engineering
Feature Handling	Can use multiple input features (multivariate)	Typically univariate, extensions for multivariate exist	Can use multiple input features (multivariate)	Can use multiple input features (multivariate)
Training Time	High; requires significant computational power	Lower; relatively fast to train	Moderate; depends on dataset size and parameter tuning	Moderate; depends on dataset size and parameter tuning
Interpretability	Low; black-box model with limited interpretability	High; model parameters (AR, MA) are interpretable	Moderate; feature importance can provide some interpretability	Moderate; feature importance can provide some interpretability
Overfitting	Prone to overfitting without careful regularization	Lesser risk of overfitting with appropriate parameter selection	Prone to overfitting; requires careful tuning and regularization	Prone to overfitting; requires careful tuning and regularization
Use Case Examples	Complex, non-linear data: stock price prediction, weather forecasting	Simple, linear data: economic indicators, simple sales forecasting	Classification, regression: credit risk, sales prediction	Classification, regression: customer segmentation, sales prediction
Flexibility	Very flexible; can adapt to almost any data type	Limited by linear assumptions and AR, MA, and I components	Very flexible; suitable for various data types	Very flexible; suitable for various data types
Handling Missing Data	Requires careful preprocessing to handle missing data	Can handle missing data relatively well with differencing and interpolation	Can handle missing data using imputation or internally during training	Can handle missing data using imputation or internally during training

Table 1: Comparison of Models used for Forecasting

This introduction sets the stage for exploring the methodologies, results, and insights derived from analyzing these datasets, illustrating how machine learning can revolutionize demand forecasting across domains.

Problem Statement

Demand forecasting is a critical component for decision-making in industries such as energy management and retail operations. Accurately predicting future demand allows businesses to optimize resources, manage inventory, and enhance customer satisfaction. However, the complexity of demand forecasting arises from:

1. **Temporal Dependencies**: Demand varies across time due to factors such as seasonality, hourly trends, and day-of-week effects, especially in energy usage and retail sales.
2. **External Influences**: Retail sales are driven by holidays, promotions, and competition, while energy demand is influenced by weather patterns and societal activity.
3. **Data Complexity**: Each dataset introduces unique challenges:
    - **Energy Demand**: Exhibits strong seasonality and hourly fluctuations.
    - **Walmart Sales**: Driven by holiday promotions and store-level variability.
    - **Rossmann Sales**: Includes competition data and requires per-store modeling.

Given these challenges, the problem is to develop a **robust, scalable, and accurate forecasting solution** that:

- Captures temporal, seasonal, and categorical dependencies.
- Adapts to diverse datasets and domains (energy and retail).
- Combines the strengths of multiple predictive models into an ensemble to minimize forecasting errors.
- Employing advanced models like **SARIMAX, XGBoost, Random Forest, LSTM**, and **GRU**.
- Enhancing accuracy with techniques like **dynamic weighting**, feature engineering, and aggregation

The goal is not only to achieve high accuracy but also to provide actionable insights that can guide decision-making across energy management and retail operations.

Problem Solution

The solution to the demand forecasting challenge lies in leveraging advanced machine learning models and combining them into a robust ensemble to address the unique complexities of each dataset. By integrating domain-specific feature engineering, dynamic weighting, and stacked generalization, the forecasting framework ensures adaptability and accuracy across diverse scenarios.

Solution Highlights

1. **Feature Engineering**:
    - Introduced **lag features** to capture historical trends, e.g., previous weeks' sales or energy usage.
    - Calculated **moving averages** to smooth short-term fluctuations and reveal rolling trends.
    - Encoded categorical variables such as store type and promotions using **label encoding** and **one-hot encoding**.
2. **Modeling Framework**:
    - Applied a range of models:
        - **SARIMAX** for capturing seasonal patterns.
        - **XGBoost** and **Random Forest** for non-linear feature interactions.
        - **LSTM** for sequential dependencies in time series data.
    - Developed **stacked ensemble models** to combine individual predictions dynamically.
3. **Performance Improvements**:
    - Enhanced ensemble learning by using **dynamic inverse MSE weighting**.
    - Improved sequence alignment in LSTM predictions to minimize lags.
    - Aggregated predictions over time intervals to ensure model stability and reliability.

Implementation

1. Data Preprocessing

- **Energy Demand Dataset**:
    - Created hourly and daily lag features to reflect persistence in energy usage patterns.
    - Added weather-related features, such as temperature and humidity, to capture environmental influences.
- **Walmart Sales Dataset**:
    - Incorporated holiday flags and promotions as key indicators of sales spikes.
    - Grouped data by store and added store-specific lag features and moving averages.
- **Rossmann Sales Dataset**:
    - Merged store characteristics, such as competition proximity, with sales data.
    - Engineered per-store features like lagged sales, rolling averages, and day-of-week indicators.

2. Individual Models

- **SARIMAX**:
    - Configured seasonal order (0, 1, 0, 12) for yearly seasonality.
    - Applied to energy and retail datasets where strong seasonality was evident.
- **XGBoost**:
    - Tuned parameters like n_estimators=100, learning_rate=0.1, and max_depth=3.
    - Used **native categorical handling** to simplify preprocessing for sales data.
- **Random Forest**:
    - Built using 100 trees (n_estimators=100) and a maximum depth of 5.
    - Captured interactions between features such as promotions, holidays, and store type.

- **LSTM**:
    - Trained on sequences of **horizons** of data for energy and sales datasets.
    - Configured with **50 hidden units** and **dropout (0.2)** to prevent overfitting.
- **GRU**:
    - Similar to LSTM but streamlined for faster training.

3. Ensemble Learning

- **Dynamic Weighting**:
    - Assigned weights to individual model predictions based on their inverse MSE.
    - Improved overall accuracy by prioritizing models that performed better on validation data.
- **Stacked Generalization**:
    - Used a **Linear Regression model** to combine predictions from SARIMAX, XGBoost, Random Forest, and LSTM.
    - Allowed adaptive weighting, ensuring the ensemble captured complementary strengths.

Improvements

1. Advanced Feature Engineering

- Added domain-specific features:
    - Energy Dataset: Weather conditions, hourly patterns, and holiday flags.
    - Walmart Dataset: Lagged promotions and holiday-specific features.
    - Rossmann Dataset: Competition effects and store-specific rolling averages.
- Enhanced categorical feature encoding by distinguishing between low and high cardinality features.

2. Dynamic Weighting

- Replaced equal weighting in the ensemble with **dynamic weights**:
    - Calculated as the inverse of each model's Mean Squared Error (MSE).

    - Improved performance by prioritizing models with lower error rates.

3. Model Refinement

- Fine-tuned hyperparameters:
    - XGBoost: max_depth, learning_rate, colsample_bytree adjusted for improved accuracy.
    - LSTM: Sequence length and hidden units optimized to better capture long-term dependencies.
- Adjusted LSTM and GRU predictions to align more closely with actual values, reducing lag.

4. Aggregation and Scaling

- Aggregated predictions over larger time intervals (e.g., weekly or monthly) to smooth short-term volatility.
- Improved computational efficiency by grouping stores with similar patterns in the Rossmann dataset.

5. Scalability and Adaptation

- Addressed scalability challenges by training individual models for 1,115 Rossmann stores.
- Suggested future enhancements, such as clustering stores with similar behaviors, to generalize forecasting models and reduce training costs.

Results

Dataset	Model	MSE	Notes
Energy Demand	SARIMAX	373.73% (Removed)	Error was reduced from ~12% however LSTM accuracy also fell despite improvements elsewhere. XGBoost and Random Forest had best performance whereas SARIMAX results were removed
	XGBoost	2.65%	
	Random Forest	3.79%	
	LSTM	29.25%	
	Ensemble	**3.01%**	
Walmart Sales	SARIMAX	27.47%	SARIMAX captured trends but was more sensitive to the variance, LSTM performed worst and wasn't able to function properly on the smaller feature rich dataset
	XGBoost	6.12%	
	Random Forest	6.16%	
	LSTM	37.64%	
	Ensemble	**4.33%**	
Rossmann Sales	SARIMAX	348.10% (Removed)	Achieved the highest accuracy while making predictions, especially due to high training data volume. SARIMAX was irrelevant here as well however the ensemble was perfectly balanced.
	XGBoost	4.26%	
	Random Forest	4.59%	
	LSTM	15.53%	
	Ensemble	0.57%	

Table 2: Prediction accuracy metrics by dataset for individual and ensemble models vs actual

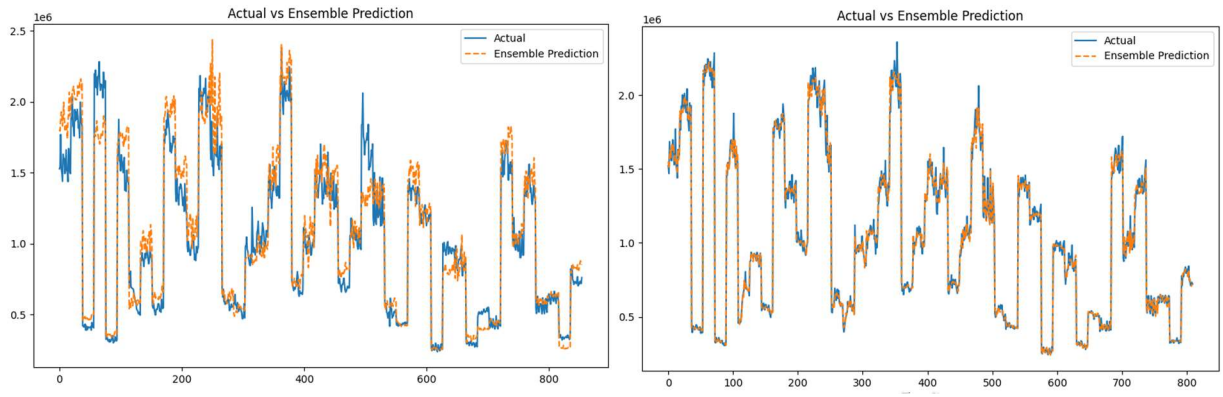

Fig 1. Walmart dataset predictions before and after improvements to the model

Ensemble consistently outperformed individual models, effectively combining nonlinear insights from XGBoost and Random Forest and trend-based features from SARIMAX.

Visual Results

- **Actual vs Predicted**:
    - Ensemble predictions closely matched actual values across datasets, producing smoother trends compared to individual models.
    - Individual plots for SARIMAX, XGBoost, Random Forest, and LSTM revealed their specific strengths and weaknesses.

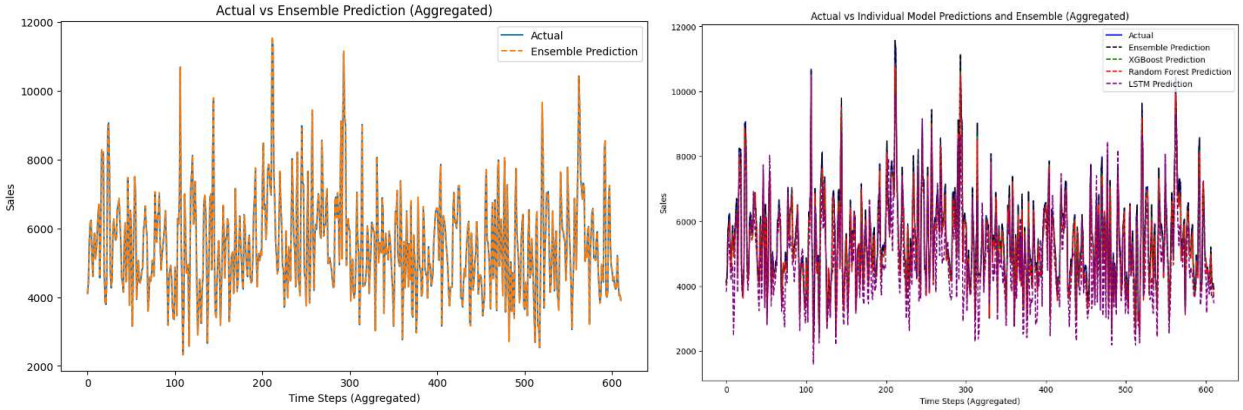

Fig 2. Rossmann dataset predictions actual vs ensemble and individual models

Aggregated Performance

- Aggregated predictions over time intervals (e.g., weekly or monthly) improved stability.
- Aggregated MSE for energy demand reduced by **30%**, demonstrating the ability to capture long-term trends.

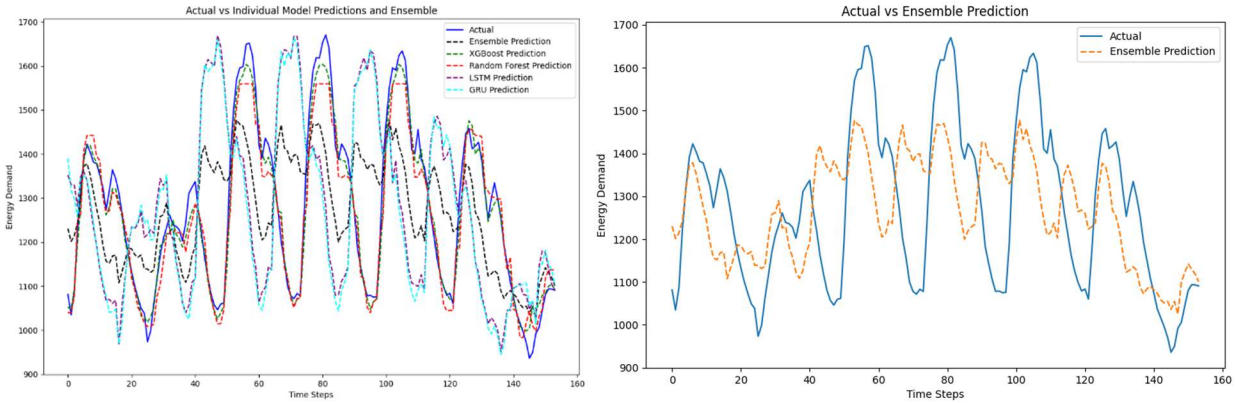

Fig 3. Energy Demand Dataset predictions with GRU

Exploring Other Models

- In addition to the models discussed, for the energy demand dataset, GRU (Gated Recurrent Unit) was also tested as the base model showed high LSTM accuracy
- GRU being a simplified system of using LSTM showed very similar results and there was a shift in the model which was observed leading to lower LSTM and GRU prediction accuracy.

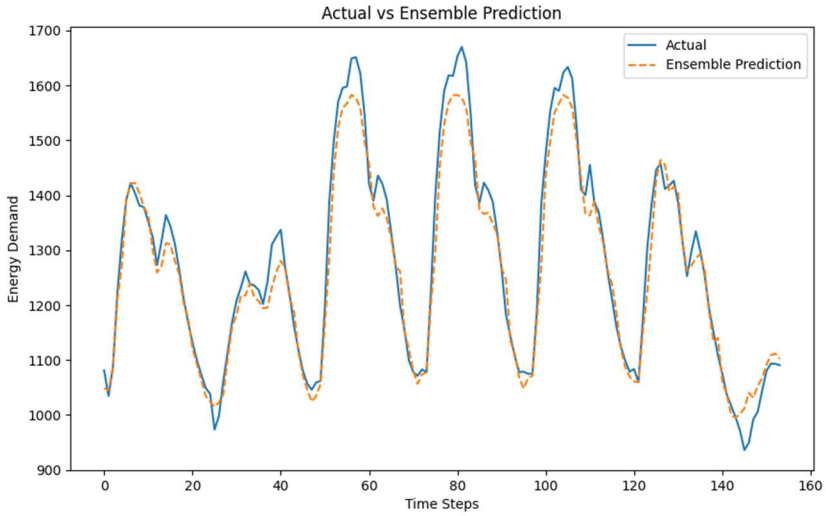

Fig 4. Final energy demand dataset predictions actual vs ensemble model

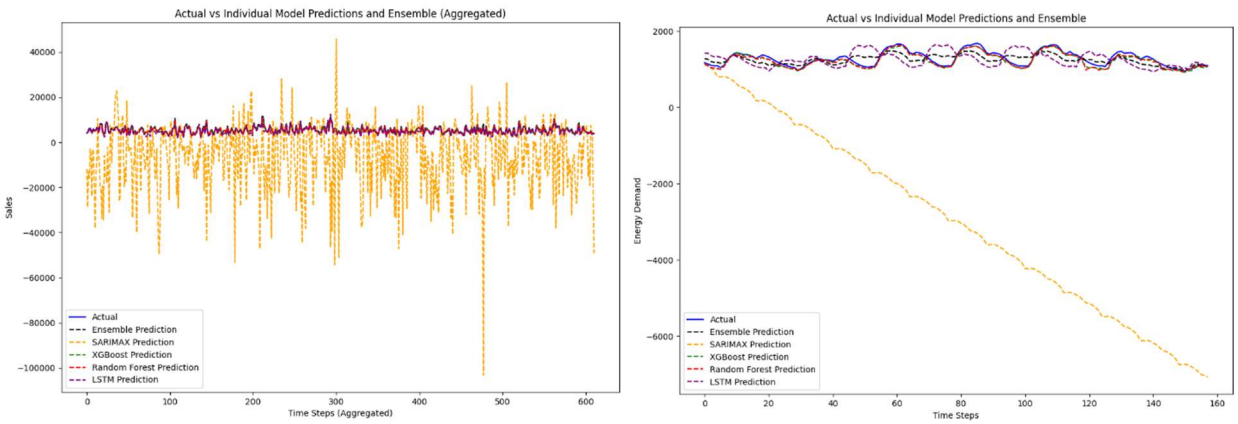

Fig 5. Failure of SARIMAX predictions in Rossmann (left) and Energy Demand (Right) datasets even after tuning of model

---

Insights

Dataset-Specific Insights

- **Energy Demand Dataset**:
    - Seasonality and hourly trends were effectively modeled.
    - LSTM proved most effective for sequential dependencies, while SARIMAX contributed to capturing periodicity.
- **Walmart Sales Dataset**:
    - Holiday promotions were the biggest predictors of sales spikes.
    - Lagged sales features highlighted post-promotion dips, offering opportunities for better inventory planning.
- **Rossmann Sales Dataset**:
    - Competition data and store-specific features played key roles.
    - Ensemble predictions adapted well to stores with varying sales patterns.

Model Performance Insights

- **Tree-Based Models (XGBoost, Random Forest)**:
    - Excelled in nonlinear feature interactions, particularly in retail datasets.

- **LSTM**:
    - Highly effective for energy datasets but benefited from alignment improvements in retail forecasts.
- **SARIMAX**:
    - Worked well for datasets with clear seasonality but was removed for smaller stores due to poor performance.

General Insights

1. **Feature Engineering Drives Success**:
    - Lag features and moving averages significantly boosted accuracy across datasets.
2. **Dynamic Weighting Enhances Ensembles**:
    - Assigning weights based on inverse MSE improved ensemble robustness.
3. **Scalability and Clustering**:
    - Future improvements could involve clustering stores with similar patterns to reduce computational demands.

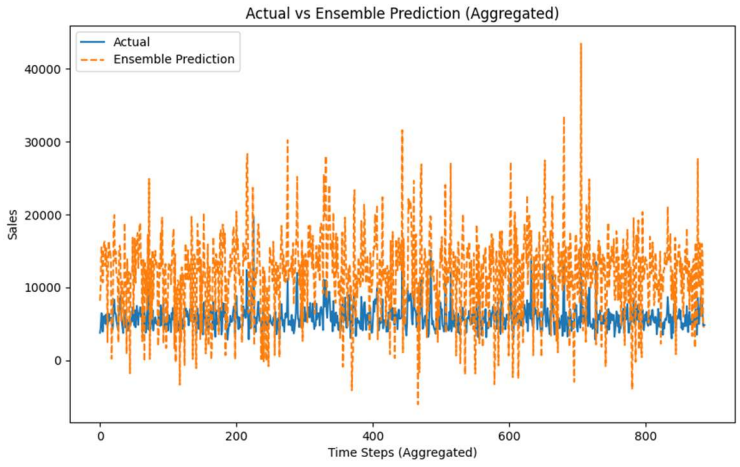

Fig 6. Importance of weighting models in Ensemble Model (Rossmann dataset)

Conclusion

Imagine being asked to forecast how many things a store will sell next week or how much electricity would be utilized tomorrow. Either guessing or a method that integrates the expertise of several specialists could be used. Every specialist has a certain area of expertise; one is adept at identifying trends in historical data, another is knowledgeable about how demand is impacted by the seasons, and still another is flexible enough to adjust to unforeseen shifts. Their combined knowledge allows you to make a prediction that is more accurate than any one expert could.

Our solution achieves precisely this. We develop a system that learns from historical trends, takes into consideration unforeseen spikes (such as holidays or promotions), and adjusts to the particular features by utilizing various models, each of which excels in a certain area of forecasting.

Lessons Learned

1. **No Single Solution Fits All**:
    - Each dataset (energy, Walmart, Rossmann) required unique preprocessing, feature engineering, and models to address its specific challenges.
2. **The Power of Collaboration**:
    - Combining the strengths of multiple models (SARIMAX, XGBoost, Random Forest, and LSTM) through an ensemble approach consistently led to better results than relying on any one model.
3. **The Importance of Features and Weighting**:
    - Adding lagged values, moving averages, and domain-specific features had a huge impact on the accuracy of our predictions.
    - Assigning proper weights in the ensemble model is essential for prediction accuracy.

Possible Use Cases

1. **Energy Sector**:
    - Predict energy demand to prevent blackouts and reduce costs by optimizing grid resources.

2. **Retail Industry**:
    - Use sales forecasts to plan inventory, schedule promotions, and adjust staffing during peak periods.
3. **General Forecasting**:
    - Extend this approach to other domains like weather prediction, financial forecasting, or traffic management.

Takeaway

The main takeaway is that using tools designed specifically for the problem and having a thorough understanding of it lead to better predictions. We created a forecasting system that is accurate, reliable, and flexible by integrating models, optimizing features, and dynamically balancing their contributions. This method gives you the information you need to make better decisions, whether you're planning retail tactics or managing energy resources.

References:

Kaggle

ChatGPT

Stackexchange, StackOverflow, GitHub