***METHODOLOGY AND EXPERIMENTS***

1. ***Aim of Study***

The primary objective of this project is to develop a robust predictive model capable of accurately forecasting course sales in the edtech industry, with a specific focus on Kaggle Learn courses. This model aims to assist in resource allocation, budgeting, and strategic planning by providing reliable sales projections. The study will investigate the relationships between various factors such as time, course attributes, and geographical location to predict the number of courses sold.

1. ***Response (Dependent) and Independent Variables***

In this experiment, we distinguish between the dependent variable, which we aim to predict, and the independent variables, which serve as inputs to our time series models.

**Dependent Variable:**

* **num\_sold:** This is the target variable, representing the number of courses sold. It's a continuous numerical variable.

**Independent Variables:**

* **date:** The date of the sale, which will be used to capture temporal patterns, seasonality, and trends.
* **country:** The country where the sale occurred, allowing for analysis of geographical differences in sales patterns.
* **store:** The platform or store where the course is sold (in this case, Kaggle Learn).
* **product:** The specific course being sold, which may have unique characteristics affecting its sales performance.

1. ***Factors and Levels***

* **Date**: This factor will be analyzed at a daily level (e.g., 2017-01-01). We'll also derive additional temporal features such as day of week, month, quarter, and year to capture various cyclical patterns.
* **Country:** This is a categorical variable. While the example shows Argentina, we'll include all countries where sales occur. Each unique country will be treated as a separate level.
* **Store:** In the given data, there's only one store (Kaggle Learn). If more stores are included in the full dataset, each would be treated as a separate level.
* **Product:** This is a categorical variable representing different courses. Each unique course title (e.g., "Using LLMs to Improve Your Coding") will be treated as a separate level.

1. ***Experiment and Design***

The experiment is designed as a comprehensive time series analysis with multiple features, employing advanced machine learning techniques for preprocessing, model building, and evaluation.

The key components of this design are as follows:

**a) Data Preparation and Feature Engineering:** The first step involves preparing the raw data for analysis. This includes handling missing values, removing duplicates, and converting date strings to proper datetime objects. Feature engineering is crucial in time series analysis, as it can significantly improve model performance. We create several types of engineered features:

* Temporal features: Extracting components like day of week, month, quarter, and year from the date.
* Lagged features: Creating past values of the target variable (num\_sold) as predictors, which allows the model to capture autoregressive patterns.
* Cyclical features: Transforming cyclical time components (like day of week or month) into sine and cosine components to capture periodicity without imposing ordinality.

**b) Preprocessing Pipeline:** A critical aspect of the experiment design is the creation of a robust preprocessing pipeline. This ensures that all data transformations are applied consistently across training, validation, and test sets. The preprocessing pipeline includes:

* Standardization for numerical features: This scales features to have zero mean and unit variance, which is important for many machine learning algorithms.
* One-hot encoding for categorical features: This converts categorical variables into a format that can be provided to machine learning algorithms to do a better job in prediction.
* Min-Max scaling for time-based features: This scales features to a fixed range, typically between zero and one.

The use of a ColumnTransformer allows different preprocessing steps to be applied to different types of features within a single pipeline.

**c) Model Selection:** The experiment incorporates multiple state-of-the-art machine learning models suited for regression tasks:

* LightGBM: A gradient boosting framework that uses tree-based learning algorithms. It's designed for distributed and efficient training.
* XGBoost: Another gradient boosting library known for its speed and performance.
* CatBoost: A gradient boosting library that handles categorical features automatically and usually requires less hyperparameter tuning.
* Gradient Boosting Regressor: A classic ensemble method that builds a series of weak learners (typically decision trees) sequentially.

These models are chosen for their ability to handle complex relationships in data and their generally good performance on a wide range of problems.

**d) Time Series Cross-Validation:** Traditional cross-validation techniques are not suitable for time series data due to the temporal dependency of observations. Instead, we use Time Series Cross-Validation, which respects the time order of the data. This method creates multiple training-test splits, each time using a different cutoff point in the time series. This approach provides a more robust estimate of the model's performance on future, unseen data.

**e) Hyperparameter Tuning:** Hyperparameter tuning is crucial for optimizing model performance. We use GridSearchCV, which performs an exhaustive search over a specified parameter grid. The key aspects of this process are:

* Parameter grid: A set of hyperparameters and their possible values to be tested.
* Scoring metric: The criterion used to evaluate performance across different hyperparameter combinations (e.g., negative mean squared error).
* Cross-validation: Using TimeSeriesSplit to ensure that the hyperparameter tuning respects the temporal nature of the data.

This process helps in finding the optimal combination of hyperparameters for each model.

**f) Feature Importance Analysis**: Understanding which features contribute most to the predictions is crucial for both model interpretation and potential feature selection. Different models provide feature importance in different ways:

* Tree-based models (like LightGBM, XGBoost) typically provide feature importance based on how often a feature is used to split the data across all trees.
* Linear models may use the magnitude of coefficients as a measure of feature importance.

Analyzing feature importance can provide insights into the factors that most influence course sales, which can be valuable for business decision-making.

**g) Experiment Workflow:** The overall workflow of the experiment involves:

* Data preparation and feature engineering
* Creating and applying the preprocessing pipeline
* Splitting the data into training and testing sets, respecting time order
* For each model:
  + Performing hyperparameter tuning
  + Training the model with the best hyperparameters
  + Evaluating the model on the test set
  + Analyzing feature importance
* Comparing models based on evaluation metrics
* Selecting the best performing model

1. ***Experiment Performance and Revision***

* Initial Model Training: Start with simple models (e.g., LightGBM, XGBoost, CatBoost) to establish a baseline.
* Model Refinement: Based on initial results, refine the model through:

a) Feature engineering: Create new features or transform existing ones.

b) Hyperparameter tuning: Use techniques like grid search or random search to optimize model parameters.

* Iterative Improvement: Repeat the process of training, evaluation, and refinement until satisfactory performance is achieved or improvements plateau.

1. **Measuring Classifier Performance**

We'll use Symmetric mean absolute percentage error metrics to evaluate model performance:

SMAPE is an accuracy measure based on percentage errors. It's a variation of the Mean Absolute Percentage Error (MAPE) that addresses some of MAPE's limitations, particularly when dealing with data that includes zero or near-zero values.

1. ***Algorithm Comparison and Selection***

We compare three advanced gradient boosting algorithms: LightGBM, XGBoost, and CatBoost. Each of these algorithms has unique characteristics that make them suitable for our time series forecasting task. We use the Symmetric Mean Absolute Percentage Error (SMAPE) as our primary evaluation metric due to its advantages in handling time series data.

a) LightGBM:

* Developed by Microsoft, known for its speed and efficiency.
* Uses a leaf-wise tree growth strategy, which can lead to better accuracy but may be prone to overfitting on small datasets.
* Particularly effective with large datasets and handles categorical features well.

b) XGBoost:

* Widely used in various machine learning tasks, known for its high predictive accuracy.
* Implements regularization techniques to prevent overfitting.
* Efficient handling of sparse data and built-in cross-validation capabilities.

c) CatBoost:

* Developed by Yandex, specifically designed to handle categorical features effectively.
* Implements ordered boosting to reduce prediction shift.
* Often requires less hyperparameter tuning compared to other algorithms.

**Evaluation Metric - SMAPE:**

Symmetric Mean Absolute Percentage Error is chosen as the primary metric for several reasons:

* Scale-independent, allowing comparison across different scales of sales volumes.
* Handles zero and near-zero values effectively, which is crucial for courses that might have no sales on some days.
* Symmetric, penalizing both over-predictions and under-predictions equally.
* Bounded between 0% and 200%, making it easier to interpret compared to unbounded metrics.

**Hyperparameter Tuning:**

Key hyperparameters to tune include:

* Number of estimators (trees)
* Learning rate
* Tree depth
* Minimum samples per leaf
* L1 and L2 regularization terms

**For each model, we examine the feature importance rankings:**

* Identify which features are consistently ranked as important across all models.
* Compare how different algorithms weigh various features, which can provide insights into their decision-making processes.
* Align feature importance with domain knowledge to ensure the models are capturing meaningful patterns.

**VIII. Predictions Graph**

A graph of blue and green lines

Description automatically generated

**(Prediction Graph)**

**Based on the XGBoost predictions graph.**

* **Overall Trend:** The model predicts a general upward trend in course sales from 2017 to 2022, with some fluctuations and seasonal patterns visible.
* **Seasonality:** There appear to be regular peaks and troughs in the sales data, suggesting seasonal patterns. These could be related to academic calendars, holiday periods, or other cyclical factors affecting course purchases.
* **Volatility:** The predictions show varying levels of volatility over time. The range of predicted values (shown by the lighter shaded areas) seems to widen in some periods, indicating increased uncertainty.
* **COVID-19 Impact:** There's a noticeable dip in sales around early 2020, which likely corresponds to the initial impact of the COVID-19 pandemic. The model captures this disruption and the subsequent recovery.
* **Recent Growth:** The latter part of the graph shows a stronger upward trend, suggesting accelerated growth in course sales in recent periods.
* **Prediction Confidence:** The light blue and green shaded areas represent the confidence intervals of the predictions. These appear to widen for future predictions, indicating increased uncertainty for longer-term forecasts.
* **Model Performance:** The model seems to capture the overall trends and patterns in the training data (blue) well. The test data predictions (green) appear to follow similar patterns, suggesting good generalization.