Binary Brains

Team Members:

1. Ambalika Rajendran- Algorithm Designer
2. Favour Asu- Model trainer
3. Joseph Hiller- Application Specialist
4. Misty Roberts- Data collector, intro, conclusion
5. Zaid Tahir Jamil- Model trainer

We chose Retail Inventory Predictor because there were plenty of examples for us to learn how companies like Apple, Nike, and other large companies complete their inventory. After doing some extensive reading we created a system that forecasts product demand to optimize the best inventory management to its ability.

**DATA COLLECTOR**

1. Identify Data Sources:

* Internal Data:
  + Historical sales data (quantities, dates, product types, locations)
    - Replenishment and Demand Planning: Based on historical data, forecasts are generated to predict future demand and determine optimal inventory levels. This helps in planning procurement, production, and replenishment activities to meet customer demand while minimizing excess inventory or stockouts.
  + Inventory levels
  + Pricing data
  + Customer data (purchase history, demographics, preferences)
    - Customer Relationship Management (CRM) Systems: companies may have CRM systems that store customer information, including purchase history, preferences, and warranty details. This data can be used to analyze customer behavior and forecast demand.
  + Marketing and promotional activity data
  + Customer feedback and reviews
  + Point-of-sale (POS) data from stores and authorized resellers
    - Point of Sale (POS) Systems: Retail stores and authorized resellers collect data on product sales, through their POS systems. This data includes information on the quantity and type of products sold.
  + Manufacturing and production data from their suppliers and contract manufacturers
  + Warehouse and distribution center data on stock levels and movements
  + Sales forecasts and demand planning data from various departments
  + Customer returns and warranty claims data
  + Online Sales Platforms: E-commerce platforms, such as the companies' official website or third-party online retailers, gather data on online sales. These platforms capture information about product availability, sales volume, and customer orders.
  + Supply Chain and Manufacturing Systems: internal systems and suppliers' systems provide data on manufacturing, shipment, and delivery of products. This includes information on inventory levels, production schedules, and logistics.
* External Data:
  + Market trends and research
  + Industry reports
  + Economic indicators
  + Competitor data
  + Social media trends
  + Weather patterns
  + Local events
  + Market research reports on consumer demand and industry trends
  + Shipping and logistics data from partners and carriers
  + Economic indicators and global events that might impact supply or demand
  + Social media sentiment analysis to gauge product popularity

2. Data Collection Methods:

* + Integration with internal systems:
  + ERP (Enterprise Resource Planning)
  + POS (Point of Sale)
    - Barcoding and Scanning: Barcodes and scanning technologies are used to track inventory movement and update stock levels accurately. This method is commonly employed in warehouses and distribution centers to capture data quickly and efficiently.
  + CRM (Customer Relationship Management)
  + Web scraping: Extract relevant data from external websites
  + APIs: Access data from external sources (e.g., weather data, social media)
  + Manual entry: For data not available through automated methods
  + Third-party data providers:
    - Purchase industry or market-specific data
    - Integration with Other Systems: Inventory data is often integrated with other business systems, such as sales, finance, and supply chain management. This integration ensures the availability of accurate and up-to-date inventory information across different departments.

3. Data Collection Tools and Technologies:

* + Databases: Store and manage collected data (e.g., MySQL, PostgreSQL)
  + ETL (Extract, Transform, Load) tools: Move data from various sources into a centralized database
  + Data cleaning and preparation tools: Validate, clean, and prepare data for analysis (e.g., Python, R, Excel)

4. Data Quality Assurance:

* + Validation: Check for completeness, accuracy, consistency, and timeliness
  + Data cleaning: Handle missing values, errors, and inconsistencies
  + Data profiling: Understand data characteristics and identify potential issues

5. Data Processing:

* + Data cleaning and validation: Checking for accuracy, completeness, and consistency
  + Data transformation: Formatting and standardizing data from various sources
  + Data aggregation and analysis: Summarizing and analyzing data to identify trends and patterns
  + Aggregation and Consolidation: Data from multiple sources is aggregated and consolidated into a central inventory management system. This allows for a comprehensive view of inventory across different channels and locations.
  + Demand forecasting: Using advanced statistical models to predict future demand
  + Inventory optimization: Utilizing algorithms to determine optimal stock levels, production scheduling, and distribution planning
  + Data is analyzed to identify trends, patterns, and anomalies. Various inventory performance metrics, such as stock levels, turnover rates, and sales forecasting, are calculated. Reports and dashboards are generated to provide actionable insights for decision-making.

Additional Considerations:

* Data privacy and compliance: Adhere to data protection regulations (e.g., GDPR, CCPA)
* Data security: Protect against unauthorized access and data breaches
* Data governance: Establish policies and procedures for data management
* Data security and privacy: Implementing robust measures to protect sensitive data
* Data governance: Establishing policies and procedures for data management and access
* Sustainability: Optimizing inventory levels to minimize waste and environmental impact

**ALGORITHM DESIGNER**

Algorithms Selection: **Long Short-Term Memory (LSTM)**

*About LSTM:*

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) that are particularly well-suited for sequential data analysis, making them a suitable choice for forecasting tasks such as retail inventory management, especially in the context of Apple retail stores.

Why LSTM is a suitable choice for predicting product demand in Apple retail stores:

1. **Sequential Data Handling:**

Apple retail sales data, including iPhone, iPad, Mac, and accessory sales, typically exhibit sequential patterns over time. LSTM networks are adept at capturing and learning from sequential dependencies in data, making them highly effective for time-series forecasting tasks.

In Apple retail inventory management, past sales data, inventory levels, pricing fluctuations, and promotional activities influence future demand. LSTM can effectively capture these temporal dynamics and use them to make accurate predictions, helping optimize inventory levels and meet customer demand.

1. **Long-Term Dependencies:**

Traditional feedforward neural networks often struggle to capture long-term dependencies in sequential data due to the vanishing gradient problem. LSTM networks address this issue by maintaining a memory cell that can retain information over long periods.

In Apple retail, demand patterns can vary seasonally, with new product launches, or in response to marketing campaigns. LSTM's ability to capture long-term dependencies enables it to model these complex patterns and make accurate forecasts, aiding in inventory planning and management.

1. **Handling Non-Linear Relationships:**

Demand forecasting in Apple retail stores involves dealing with non-linear relationships between various factors such as product launches, pricing strategies, customer preferences, and external factors like economic conditions or technological trends.

LSTM networks, being neural networks, have the capacity to model complex, non-linear relationships between input variables and output predictions. They can learn intricate patterns and dependencies within the data, making them suitable for the inherently non-linear nature of demand forecasting in Apple retail environments.

1. **Ability to Handle Multivariate Time Series:**

Apple retail inventory data typically involves multiple variables such as sales quantities for different product categories, pricing changes, promotional events, and external factors like market trends or competitor actions.

LSTM networks can effectively handle multivariate time series data, allowing them to incorporate information from various sources into the forecasting model. This capability enables more accurate predictions by considering the interplay between different factors influencing demand at Apple retail stores.

1. **Adaptability to Data Dynamics:**

Apple retail environments are dynamic, with demand patterns evolving over time due to changing consumer behaviors, product innovations, and market trends. LSTM networks are adaptive and can continuously update their internal state based on incoming data.

By training on historical data and updating the model with new observations over time, LSTM algorithms can adapt to changing demand patterns in Apple retail stores and maintain accuracy in forecasting future demand, facilitating efficient inventory management and allocation.

**Assumptions**

* LSTM assumes that past patterns and dependencies within the data will continue, which may not always hold true in rapidly changing retail environments. However, LSTM's adaptability helps mitigate this assumption by continuously updating the model with new data from Apple retail stores.
* LSTM performance relies on the availability and quality of historical data from Apple retail stores. Insufficient or noisy data may hinder the model's ability to accurately capture patterns and make reliable forecasts, emphasizing the importance of data quality and preprocessing.
* While LSTM is effective for sequential data analysis, it may not perform optimally if the underlying demand patterns in Apple retail stores are highly irregular or unpredictable. Additional feature engineering or preprocessing may be necessary to enhance model performance and address specific characteristics of Apple retail sales data.

**Machine Learning Algorithm Types**

Machine learning algorithms can be broadly categorized into several types based on their functionality, learning style, and application domains. Here is a comprehensive list of different types of machine-learning algorithms:

**Supervised Learning Algorithms:**

* Linear Regression
* Logistic Regression
* Support Vector Machines (SVM)
* Decision Trees
* Random Forests
* Gradient Boosting Machines (GBM)
* Neural Networks (Deep Learning)
* k-Nearest Neighbors (k-NN)
* Naive Bayes Classifier
* Ensemble Methods (e.g., Bagging, Stacking)

**Unsupervised Learning Algorithms:**

* K-Means Clustering
* Hierarchical Clustering
* Gaussian Mixture Models (GMM)
* Principal Component Analysis (PCA)
* t-Distributed Stochastic Neighbor Embedding (t-SNE)
* Self-Organizing Maps (SOM)
* Apriori Algorithm (Association Rule Learning)

**Semi-Supervised Learning Algorithms:**

* Self-training
* Co-training
* Tri-training

**Reinforcement Learning Algorithms:**

* Q-Learning
* Deep Q-Networks (DQN)
* Policy Gradient Methods
* Actor-Critic Methods
* Monte Carlo Tree Search (MCTS)

**Instance-based Learning Algorithms:**

* k-Nearest Neighbors (k-NN)
* Case-Based Reasoning (CBR)

**Bayesian Learning Algorithms:**

* Naive Bayes Classifier
* Bayesian Networks
* Gaussian Processes

**Evolutionary Algorithms:**

* Genetic Algorithms
* Genetic Programming
* Evolution Strategies
* Particle Swarm Optimization (PSO)

**Deep Learning Architectures:**

* Convolutional Neural Networks (CNN)
* Recurrent Neural Networks (RNN)
* Long Short-Term Memory Networks (LSTM)
* Gated Recurrent Units (GRU)
* Autoencoders
* Generative Adversarial Networks (GAN)
* Transformer Models

**Other Learning Algorithms:**

* Decision Trees
* Rule-Based Learning
* Fuzzy Logic Systems
* Learning Vector Quantization (LVQ)
* Locally Weighted Learning (LWL)

### **Model Training Process**

**Data Preprocessing:** Data preprocessing is crucial in training an LSTM model because it ensures the input data is clean, normalized, and appropriately structured, allowing the model to learn patterns effectively and preventing issues like vanishing or exploding gradients during training. Proper preprocessing also helps capture the temporal dependencies in the time series data, enabling the LSTM to make accurate and meaningful predictions.

* + Clean the data by handling missing values, outliers, and anomalies.
  + Normalize or scale numerical features.
  + Encode categorical variables if needed.

**Time Series Split:** This approach ensures the model is trained and validated on temporally relevant data, allowing it to capture and leverage the inherent time dependencies crucial for accurate product demand predictions.

* + Split the data into training and testing sets, maintaining the chronological order to simulate real-world forecasting scenarios.



**Feature Engineering:** Well-crafted features, such as lag variables or moving averages, enhance the model's ability to understand and predict temporal dependencies, ultimately improving its accuracy in forecasting product demand.

* + Extract relevant features for the LSTM model, such as lag features, moving averages, and other time-based features.

**Model Architecture:** The choice of layers, neurons, and hyperparameters influences the model's ability to handle long-term dependencies, adapt to varying demand patterns, and ultimately optimize inventory management decisions.

* + Design an LSTM neural network architecture suitable for time series forecasting.
  + Configure the number of layers, neurons, and other hyperparameters.
  + Incorporate dropout layers to prevent overfitting.

**Model Training:** Through iterative training, the LSTM refines its weights and biases, improving its ability to capture temporal dependencies and optimize inventory decisions based on evolving demand scenarios.

* + Train the LSTM model using the training set.
  + Utilize backpropagation through time (BPTT) to update weights.
  + Use an appropriate optimization algorithm (e.g., Adam, RMSprop).

**Hyperparameter Tuning:** Hyperparameter tuning is the process of systematically optimizing the configuration settings, known as hyperparameters, of a machine learning model to improve its performance on a specific task. These hyperparameters, such as learning rates or the number of hidden layers, are external to the model and are adjusted through experimentation to achieve optimal results.

* + Optimize hyperparameters through techniques like grid search or random search.

**Model Evaluation:** Assesses the model's accuracy, helping ensure that the forecasting results are reliable and can be trusted for making informed inventory optimization decisions.

* + Evaluate the model's performance on the validation set.
  + Adjust hyperparameters as needed.

**Fine-Tuning:** Iterative adjustments to the model based on validation results, ensuring its continual improvement and adaptability to evolving patterns in demand data.

* + Refine the model by incorporating feedback from validation results.
  + Repeat training and evaluation until satisfactory performance is achieved.

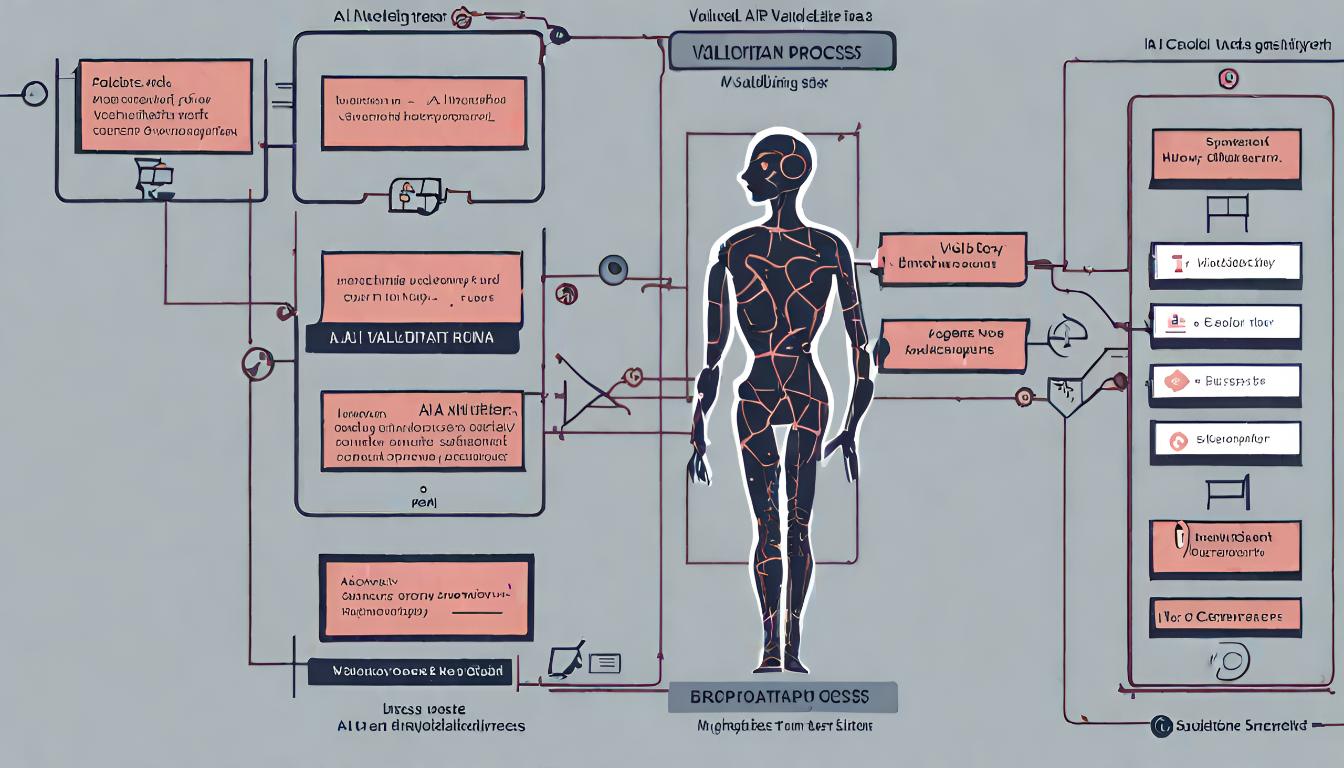
### **Validation Methods:**

**Time Series Cross-Validation:** validation technique used in training models, particularly for time-dependent data like in forecasting, where the dataset is split into multiple consecutive folds, ensuring that each fold's training data precedes the validation data, mimicking real-world scenarios and providing a more reliable evaluation of a model's performance over time. This approach helps assess the model's ability to generalize and make accurate predictions on unseen future data points, considering the temporal dependencies inherent in time series datasets.

* + Implement k-fold cross-validation, considering the time series nature of the data.
  + Preserve the temporal order to mimic real-world forecasting scenarios.

**Holdout Validation:** Holdout validation is a technique where a portion of the dataset is reserved for final model evaluation after training and tuning, ensuring an unbiased assessment of the model's performance on unseen data.

* + Reserve a portion of the data for final model testing after training and validation.



### **Performance Metrics**

**Mean Absolute Error (MAE):** A metric that calculates the average absolute differences between predicted and actual values, providing a measure of the average magnitude of errors in a prediction model

* + Calculate the absolute difference between predicted and actual values and average across all instances.

**Mean Squared Error (MSE):** Metric that calculates the average of the squared differences between predicted and actual values, emphasizing larger errors in a prediction model.

* + Square the differences between predicted and actual values, then average.

**Root Mean Squared Error (RMSE):** Root Mean Squared Error (RMSE) is the square root of the Mean Squared Error (MSE), and while both metrics measure the average magnitude of errors in a prediction model, RMSE provides a measure in the same units as the target variable, making it more **interpretable** and suitable for direct comparison with the original data. RMSE tends to penalize large errors more heavily than MSE, as it accounts for the squared differences but is presented on the original scale of the data.

* + Take the square root of the MSE to provide a more interpretable scale.

**Mean Absolute Percentage Error (MAPE):** Mean Absolute Percentage Error (MAPE) is a percentage-based metric that calculates the average absolute percentage differences between predicted and actual values, emphasizing the relative magnitude of errors, while Root Mean Squared Error (RMSE) measures the average magnitude of errors in the original units of the data, regardless of their sign.

* + Calculate the percentage difference between predicted and actual values and average across all instances.

**Forecast Bias:** The systematic tendency of a predictive model to consistently overestimate or underestimate the actual values, indicating a directional error in its forecasts.

* + Measure the tendency of the model to consistently over- or under-forecast.

**R-squared (R2):** A statistical metric that represents the proportion of variance in the dependent variable (target) explained by the model, indicating the goodness of fit or how well the model's predictions match the actual data.

* + Assess the proportion of variance in the dependent variable explained by the model.

*Regularly monitor the model's performance on new data and consider retraining or updating the model as necessary to maintain accuracy over time.*

**Reflection:** Notice how many steps our model must go through before we start training the model. The multi-step process leading up to actual training data in the training process highlights the meticulous and structured approach required to develop effective AI models. Initiating with data collection ensures a comprehensive understanding of the problem domain, while subsequent steps like preprocessing, time series split, feature engineering, and model architecture design collectively contribute to creating a robust and informed training dataset. This structured approach is crucial because it addresses challenges such as data quality, temporal dependencies, and the model's ability to comprehend complex patterns, ultimately enhancing the model's performance in tasks like predicting product demand for inventory management. Each step contributes to a refined and well-prepared dataset, laying the foundation for a successful model training process.

**APPLICATION SPECIALIST**

**Understanding the Application Context:** As an Application Specialist, the first task is to understand the specific needs of Apple’s inventory management system. Apple is known for its efficient supply chain and inventory management practices. The company leverages advanced technology, such as automation and data analytics, to optimize its operations. The role involves understanding how these systems work and identifying how the LSTM model can enhance its performance. This includes understanding the data flow, the current technology stack, and how predictions from the LSTM model can be utilized for decision-making.

**API (Application Programming Interface) Development:** Once a clear understanding of the application context is established, the next step is to develop an Application Programming Interface (API). The API will serve as a bridge between the application and the LSTM model. This is crucial as it allows the application to send input data to the model and receive the output predictions without any compatibility issues. The API should be designed to handle large volumes of data and should be robust enough to handle any potential errors or exceptions.

**Model Deployment:** After the API is developed, the LSTM model needs to be deployed in a production environment. This involves choosing a suitable platform for deployment, ensuring the model is accessible by the application, and setting up necessary security measures. Tools like TFX, Mlflow, and Kubeflow can simplify the deployment process and ensure the model is accessible for real-world use.

**Integration with the Application:** Once the LSTM model is deployed, it needs to be integrated with Apple’s inventory management application. This involves modifying the application to call the model’s API, passing the necessary inputs (like historical sales data), and using the model’s output (like predicted future sales) for decision-making. This step is crucial as it allows the application to leverage the predictive capabilities of the LSTM model for inventory management.

**Testing** After the LSTM model is integrated with the application, rigorous testing needs to be conducted. This includes unit testing, integration testing, and system testing to validate that the model’s predictions are correctly interpreted by the application and that the system meets the required performance standards. Any bugs or issues identified during the testing phase should be fixed before the system is fully deployed.

**Monitoring and Updating** Finally, a monitoring system needs to be set up to continuously track the performance of the LSTM model and the overall system. This is crucial as the performance of machine learning models can degrade over time if they are not updated with new data. Any significant changes in performance should trigger a retraining or adjustment of the model.

In conclusion, inventory is a continuous ongoing process. There are great impacts to businesses by having models that help them keep track of inventory, but like anything, there will always be challenges and ways to improve the model. The potential impacts include increased sales and reduced lost sales, enhanced efficiency, better planning, and less on-hand overage inventory. The challenges that businesses go through are data inaccuracies, changing trends, and ethical considerations. With every challenge, there are improvements like AI-powered forecasting, advanced data capture, better inventory management, and automated decision-making. When companies address the challenges and embrace future improvements, they have the potential to unlock significant benefits in sales, efficiency, and most importantly they will have happier customers.

**References:**

* <https://finbox.com/BASE:AAPL/explorer/inventory/>
* <https://www.annualreports.com/Company/apple-inc>
* <https://finance.yahoo.com/quote/AAPL/reports/>
* <https://www.apple.com/newsroom/2023/11/apple-reports-fourth-quarter-results/>
* <https://www.apple.com/newsroom/pdfs/FY23_Q1_Consolidated_Financial_Statements.pdf>
* <https://www.gurufocus.com/term/Inventory/AAPL/Total-Inventories/Apple>
* <https://www.stock-analysis-on.net/NASDAQ/Company/Apple-Inc/Analysis/Inventory>
* <https://investor.apple.com/investor-relations/default.aspx>
* <https://dfreight.org/blog/an-insight-into-apples-supply-chain-strategy/#:~:text=Apple%20works%20with%20logistics%20partners,and%20manage%20its%20inventory%20levels>.
* <https://venturebeat.com/business/apple-inventory/>
* <https://machinelearning.apple.com/research/language-identification-from-very-short-strings>
* <https://www.analyticsvidhya.com/blog/2022/01/the-complete-lstm-tutorial-with-implementation/>
* <https://ranko-mosic.medium.com/lstm-primer-6d7e1cfa704a>
* <https://blog.cambridgespark.com/deploying-a-machine-learning-model-to-the-web-725688b851c7>
* <https://machinelearning.apple.com/research/language-identification-from-very-short-strings>
* Aggarwal, | BY Shivanshu. (n.d.). *The Ultimate Guide to Building Your Own LSTM models*. ProjectPro. <https://www.projectpro.io/article/lstm-model/832>