Week 1-4: Review of ML Basics & Business Problem Solving with Dirty Datasets

**Objective**: Establish a solid understanding of machine learning algorithms while handling messy, real-world datasets. Focus on preprocessing, cleaning, and feature engineering.

Week 1: Review of Python Basics & Introduction to ML

**Day 1-2**: Review Python fundamentals, libraries (NumPy, pandas, scikit-learn).

 Day **1-2**: Review Python fundamentals, libraries (NumPy, pandas, scikit-learn).

 Day **3-4**: Introduction to basic ML concepts: Supervised vs. unsupervised learning, evaluation metrics.

 Day **5**: Case study discussion: Identifying business problems suitable for ML.

Week 2: Supervised Learning - Regression and Classification (Hands-on Projects) with Messy Datasets

**Objective**: Students will apply supervised learning techniques on real-world business problems, focusing on regression and classification, and learn how to clean and preprocess messy datasets.

Day 1-2: Regression (Predicting Continuous Outcomes)

 Dataset: Predicting **sales**, **house prices**, or **stock market trends** with missing values, outliers, and noise.

* Example: **House Prices** dataset with missing data and outliers.

 Learning **Goals**:

* **Data cleaning**: Handle missing values using **imputation** or **removal** techniques.
* **Outlier detection** and handling (using **IQR** or **Z-scores**).
* **Feature engineering**: Create new features (e.g., time-based features for sales prediction, polynomial features).
* Train a **linear regression** model and evaluate using **RMSE**, **R-squared**, and **MAE**.

Day 3-4: Classification (Predicting Categorical Outcomes)

 Dataset: Predicting **customer churn** or **loan approval** with noisy data, imbalanced classes, and missing features.

* Example: **Customer Churn** dataset with missing values and imbalanced classes.

 Learning **Goals**:

* **Handling imbalanced classes**: Techniques like **SMOTE**, **undersampling**, and **oversampling**.
* **Missing data**: Imputation or flagging as a feature.
* **Feature selection**: Using techniques like **feature importance** or **correlation** to choose the best features for the model.
* Train a **logistic regression** model, and other algorithms (KNN, Random Forest, SVM).
* Evaluate using **accuracy**, **precision**, **recall**, and **F1-score**.

Day 5: Real-World Application

 Project: Combine regression and classification problems in a single business scenario (e.g., **sales prediction** and **customer segmentation**).

 Focus on cleaning, preprocessing, and feature engineering to optimize the model performance.

**Week 3: Unsupervised Learning (Hands-on Projects) with Messy Datasets**

**Objective**: Students will apply unsupervised learning techniques to uncover patterns in data while dealing with noisy, unstructured data.

Day 1-2: Clustering (Grouping Similar Data Points)

* **Dataset**: **Customer Segmentation** dataset with missing, inconsistent, and noisy data.
  + Example: **Customer Behavior** dataset with missing customer demographic information.
* **Learning Goals**:
  + **Handling missing values**: Use **mean/mode imputation** or create flag features for missing data.
  + **Feature scaling**: Standardize/normalize features for better clustering.
  + Implement **K-means clustering** and **DBSCAN**.
  + Visualize clusters using **PCA** or **t-SNE**.
  + Analyze clusters to derive actionable business insights.

Day 3-4: Dimensionality Reduction (Simplifying Complex Data)

 Dataset: High-dimensional data for visual analysis.

* Example: **Market Basket Analysis** dataset.

 Learning **Goals**:

* **Preprocessing**: Remove irrelevant features and scale data.
* Use **PCA** and **t-SNE** for dimensionality reduction.
* Visualize high-dimensional data to understand patterns and relationships.

Day 5: Real-World Application

**Project**: Apply clustering and dimensionality reduction techniques to a business scenario (e.g., customer segmentation and pattern recognition).

* Clean and preprocess data to improve clustering performance.

Week 4: Quick Project on Clean and Messy Datasets (Hands-on Learning of Multiple Algorithms with Preprocessing)

**Objective**: Students will work on a quick project combining regression, classification, and unsupervised learning, with focus on cleaning, preprocessing, and feature engineering.

Day 1-2: Data Preprocessing and Exploration

* **Dataset**: Clean but noisy dataset (sales, employee data, etc.).
* **Learning Goals**:
  + Handle missing values, encoding categorical variables, and normalization.
  + Explore data using **descriptive statistics**, **visualizations** (histograms, box plots, etc.), and feature selection techniques.
  + Feature engineering: Derive new features that can improve model accuracy (e.g., time-based features or interaction features).

Day 3-4: Model Building and Evaluation

**Learning Goals**:

* Apply **regression** (e.g., sales prediction) and **classification** (e.g., churn prediction) models on a messy dataset.
* Use advanced techniques like **grid search** and **cross-validation** for model tuning.
* Evaluate models using appropriate metrics.

Day 5: Real-World Use Case

 Project: Create a combined model for predicting **sales performance** (regression) and **customer retention** (classification) while dealing with missing, noisy, and unstructured data.

 Use clustering for customer analysis and optimize the models using feature engineering.

Week 5-8: Model Deployment and API Development

**Objective**: Transition from model training to serving models and making predictions via APIs. Students will work on deployment using real-world data issues.

Week 5: Introduction to Model Deployment

 **Day 1-2**: Introduction to **model deployment** and **API development**.

* Frameworks: **Flask** or **FastAPI** for serving models.

 Day **3-4**: Build a **simple API** for real-time predictions.

* Use Flask to serve a regression or classification model.

 Day **5**: Hands-on exercise: Build an API for a business use case (e.g., **customer churn prediction**).

Week 6: Model Optimization and Testing

 Day **1-2**: Focus on optimizing models for production.

* Implement **model versioning**, **scaling**, and **monitoring**.

 Day **3-4**: Introduction to **Docker** for containerizing models.

* Hands-on: Containerize the model with Docker and deploy it locally.

 Day **5**: Testing and handling various input/output formats in APIs.

Week 7: Data Engineering for ML Pipelines

 Day **1-2**: Introduction to **data engineering** concepts.

* Understanding **SQL/NoSQL databases**, **ETL pipelines**.

 Day **3-4**: **Data orchestration** using **Airflow** or **Luigi** for pipeline automation.

* Example: Build a pipeline to automate data fetching, model training, and predictions.

 Day **5**: Hands-on exercise: Integrate models with databases for real-time data predictions.

Week 8: Building End-to-End Systems

 Day **1-2**: Designing an **end-to-end ML system**.

* Connecting models with databases and APIs.

 Day **3-4**: Building an automated **data pipeline** for processing and serving predictions.

 Day **5**: Real-world application: Design an end-to-end business prediction system.

Week 9-12: Deep Learning, NLP, Agents, and Chatbot Development

**Objective**: Focus on **LLM-based intelligent agents** capable of performing multiple tasks, decision-making, and executing real-world business actions.

Week 9: Deep Learning - CNNs & Transfer Learning

 Day **1-2**: Introduction to **CNNs** and their use in image classification.

* Example: Image classification using **YOLO** or pre-trained CNNs.

 Day **3-4**: **Transfer Learning** with pre-trained models.

* Hands-on: Fine-tune a pre-trained model like **YOLO** for a specific business problem (e.g., product classification).

 Day **5**: Implement **transfer learning** on a custom dataset.

Week 10: Large Language Models (LLMs) and RAG

 Day **1-2**: Introduction to **LLMs** and their applications in NLP.

* Fine-tuning pre-trained LLMs for text classification, summarization, and sentiment analysis.

 Day **3-4**: Understanding **RAG (Retrieval-Augmented Generation)** and **vector databases** for document retrieval.

* Hands-on: Implement RAG for improved search and question-answering systems.

 Day **5**: Build a **text classification** model using fine-tuned LLMs, focusing on real-time business applications.

Week 11: LLM Agents for Multiple Business Tasks

**Day 1-2: Introduction to LLM Agents**:

* **Definition**: LLM-based agents capable of performing complex tasks, such as customer support, decision-making, task automation, and dynamic interactions.
* **Key Capabilities**: These agents can interact with users, query databases, trigger workflows, and execute multiple actions within a business context.
* **Example**: An **agent** for an **e-commerce business** that can handle **inventory checks**, **order processing**, **customer inquiries**, and **personalized product recommendations**.

**Day 3-4: Building Task-Oriented LLM Agents**:

* **Task-Oriented Agents**: Focus on **multi-tasking**—e.g., agents that can handle diverse business operations like **automating employee queries**, **scheduling**, and **document processing**.
* **Practical Application**: Building an **LLM agent** that combines natural language processing and business task execution, such as:
  + **Email summarization and response** generation for support teams.
  + **Data retrieval** from databases and **workflow triggering** based on queries.
* **Integration** with **third-party APIs** to perform actions such as sending emails, updating CRM systems, or retrieving information from external services.

**Day 5: Real-World Business Application**:

* **Agent Development Project**: Build a multi-functional **LLM agent** capable of automating business tasks, like processing support tickets, generating automated emails, or analyzing customer feedback.
* Example: **Business Process Automation Agent** for handling customer orders, sending confirmations, and generating reports.

Week 12: Advanced LLM Agents with RAG & Speech Capabilities

 Day **1-2: Advanced LLM Agents with RAG**:

* Implement **RAG (Retrieval-Augmented Generation)** for intelligent, contextual responses.
* Example: Create an **agent** that uses RAG to retrieve relevant documents or database entries to respond to complex user requests in real-time.
* **Integration with business workflows**: Allow agents to access **internal knowledge bases** and external data sources for enhanced decision-making.

 Day **3-4: Adding Speech Capabilities to Agents**:

* Integrate **speech recognition** (e.g., using **Google Speech-to-Text** or **DeepSpeech**) into LLM agents.
* Develop a **voice-enabled assistant agent** that can handle both text and voice interactions, providing a conversational experience.
* **Example**: Build an agent that can handle customer orders, answer FAQs, and provide product recommendations through voice commands.

 Day **5: Final Project**:

* Build a **full-featured intelligent LLM agent** capable of interacting with users via text and voice, performing **multi-step business tasks** such as:
  + **Order processing**, **customer service**, **task automation**, and **data retrieval** from internal databases.
* This project will integrate all the previous components into a single functional system that can perform complex real-world business actions.

Final Project (Weeks 13-14)

**Project**: A comprehensive, real-world business application combining all the concepts learned throughout the course, particularly focusing on **LLM agents**.

* The final project will involve building an **intelligent LLM agent** that integrates with various systems (databases, APIs, etc.) to automate and perform complex business tasks.
* Example: Develop a **virtual business assistant agent** capable of handling **sales orders**, **inventory management**, **customer support**, and **marketing automation** through both **text** and **voice** interactions.