Fraud Detection Model Analysis using IBM Watsonx.ai

In today's digital era, data is one of the most valuable resources, and extracting meaningful insights from it can lead to smarter decisions, better predictions, and improved efficiency. Machine Learning (ML) plays a key role in achieving this by enabling systems to learn from historical data and make accurate, data-driven predictions.

This project uses IBM watsonx.ai to build and train a machine learning model. Machine learning is a way to teach computers to learn from data and make predictions or decisions without being told exactly what to do step-by-step.

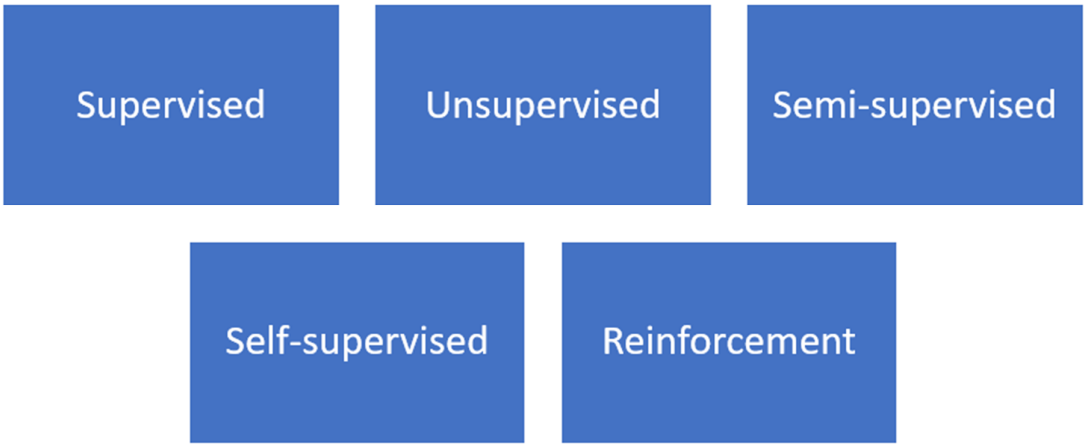
With watsonx.ai, we can easily upload our data, train different models, and choose the one that works best. The platform makes it simple to test ideas, improve accuracy, and quickly deploy the model so it can be used in real-life situations.

The main goal of this project is to create a smart system that can study data, find patterns, and give useful predictions. Over time, the system will keep learning and improving, making it more accurate and helpful.

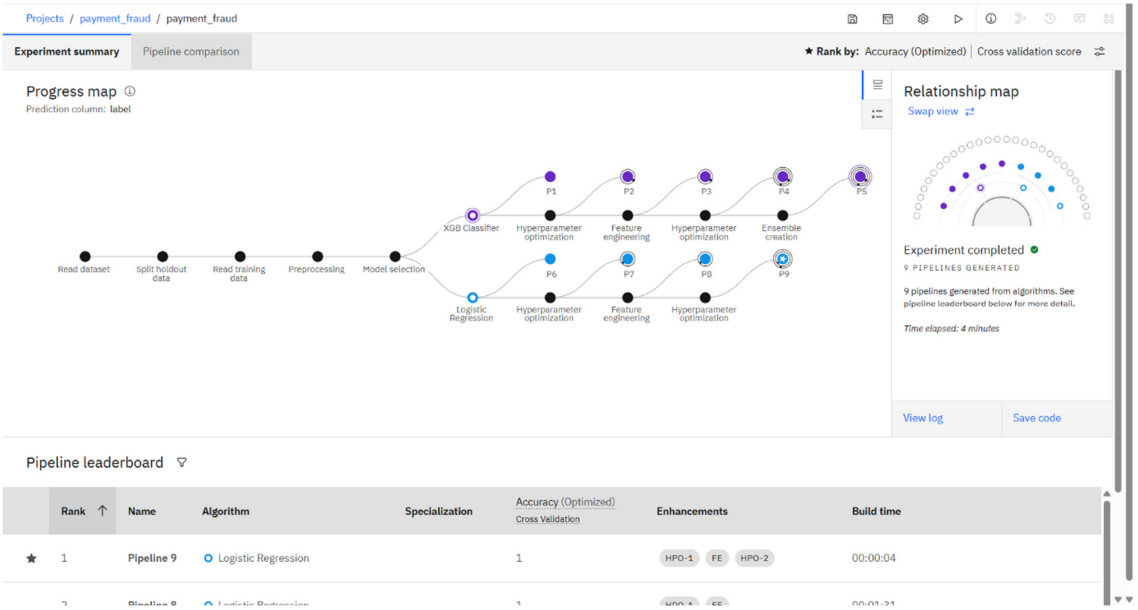
Definition
Machine Learning (ML) is a branch of Artificial Intelligence (AI) that enables systems to imitate human learning, perform tasks autonomously, and improve accuracy and performance over time through experience and exposure to more data.

- Decision Process – Predicting or classifying inputs.
- Error Function – Evaluating predictions for accuracy.
- Model Optimization – Iteratively updating model weights to minimize error until acceptable accuracy is achieved.

Types of Machine Learning Algorithm:



Explanation of the process



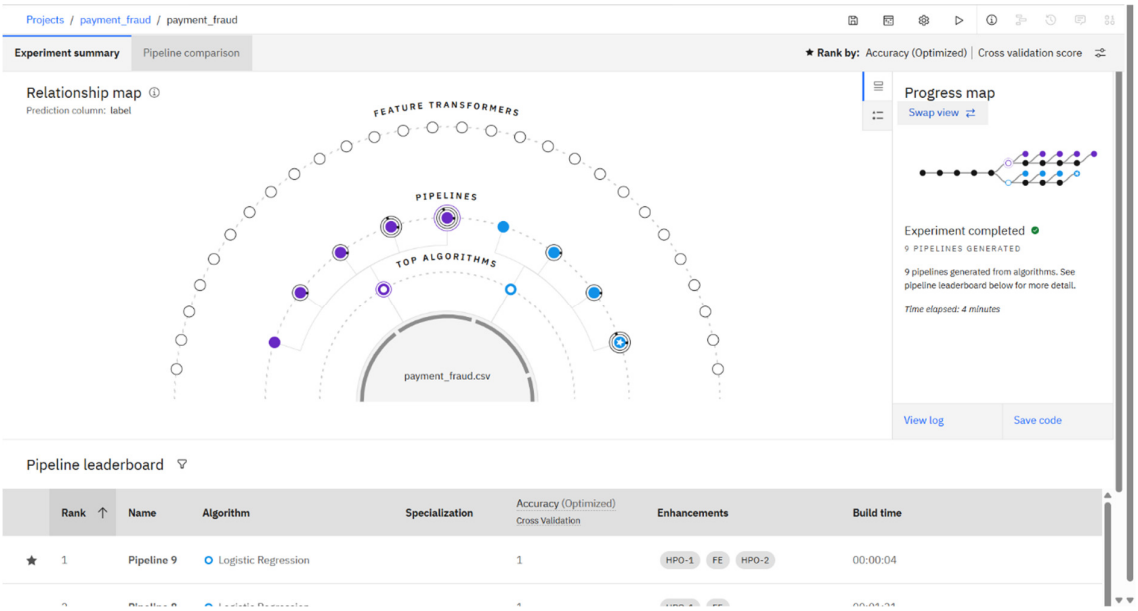
In this project the Payment Fraud Detection dataset from Kaggle is used and builds a machine learning model in IBM watsonx.ai to detect fraudulent transactions.

When we load the dataset into watsonx.ai, it automatically runs the full machine learning process for us:

1. Load the data – Reads the CSV file and prepares it for training.
2. Split the data – Divides the dataset into training and testing parts.
3. Clean and process the data – Handles missing values, converts text to numbers, and scales features.
4. Try different models – Tests several algorithms like Logistic Regression and XGB Classifier.
5. Tune the settings – Adjusts model parameters (Hyperparameter Optimization) to get better accuracy.
6. Engineer features – Creates or modifies features to help the model perform better.
7. Compare results – Ranks the models to see which one performs best.

In this experiment, Logistic Regression gave the highest accuracy (100% in testing), after going through feature engineering and multiple optimization steps.

Watsonx.ai makes the process quick and automated, allowing us to focus on understanding the results and checking if the model works well with real-world data.



- The center circle represents the dataset used: payment_fraud.csv.
- The next layer (Top Algorithms) displays the machine learning algorithms tested.
- The outer layer (Pipelines) represents variations of each algorithm with different preprocessing techniques and optimizations.
- Feature Transformers are shown on the outermost circle, indicating different ways features were processed before training.
- The star marks the best-performing pipeline based on your chosen ranking metric (in this case, Accuracy (Optimized)).
- The system tested 9 different pipelines (combinations of algorithms and settings).
- The relationship map shows how the dataset was processed through different algorithms and pipelines.
- The progress map confirms the experiment finished successfully in 4 minutes.
- The best model was Pipeline 9 using Logistic Regression.
- Accuracy was the highest for this pipeline after feature engineering and hyperparameter tuning.

This chart shows how each of the 9 pipelines (P1 to P9) performed across different evaluation metrics.Each pipeline is a different combination of an algorithm, feature processing, and parameter tuning.The colored lines represent the pipelines. The closer a line is to the *better end* of each metric, the better that pipeline performed.

The following metrics are considered:

1. **Accuracy**
    - Measures how often the model predicted correctly (both fraud and non-fraud).
    - Higher accuracy means fewer wrong predictions.
2. **Average Precision**
    - Measures precision across different decision thresholds.
    - Useful for imbalanced datasets like fraud detection where fraud cases are rare.
3. **Balanced Accuracy**
    - Adjusts accuracy for class imbalance.
    - Averages the recall for both fraud and non-fraud cases.
4. **F1 Score**
    - The balance between precision and recall**.**
    - A good F1 score means the model is both accurate and consistent in catching fraud.
5. **Log Loss** *(Lower is Better)*
    - Measures how well the model predicts probabilities.
    - Low log loss means the model is confident and correct.
6. **Precision**
    - Of the transactions predicted as fraud, how many were actually fraud.
    - High precision means fewer false alarms.
7. **Recall**
    - Of all real fraud cases, how many the model actually caught.
    - High recall means fewer missed fraud cases.
8. **ROC AUC (Area Under the Curve)**
    - Measures the ability to separate fraud from non-fraud.
    - Higher values mean better separation.

Conclusion:

This experiment proved that intelligent model selection and tuning can greatly enhance fraud detection accuracy.Among nine tested pipelines, Pipeline 9 that is Logistic Regression with feature engineering and hyperparameter optimization—stood out as the clear winner.It consistently delivered top performance across all metrics, catching more fraud cases while reducing false alerts.Such a model can significantly strengthen payment security and protect against financial losses in real-world applications.

****