Lesson 6 — Classification in Machine Learning

1. Introduction to Classification

Classification is a type of machine learning where the goal is to put data into groups, called *classes*. The model learns from past examples and then predicts the right class for new data. For example, in healthcare, a model can classify patients as “high risk” or “low risk.”

Classification is different from regression. Regression predicts numbers (continuous values), while classification predicts labels (categories).

- **Classification example**: Predict if a bank transaction is fraud or not.
- **Regression example**: Predict the price of a stock in the market.

2A. Classification Algorithms

Logistic Regression

- **How it works**: Uses math to calculate the probability of an outcome. If the probability is above 0.5, the model chooses one class, otherwise the other.
- **Use case**: Predicting if a machine in a factory will break down soon.
- **Pros**: Simple, fast, easy to understand.
- **Cons**: Not good for complex data with many patterns.

Decision Trees

- **How it works**: Splits the data into branches based on questions about features, until it reaches a decision at the end.
- **Use case**: Deciding if a loan application should be “approved” or “denied.”
- **Pros**: Easy to explain and visualize.
- **Cons**: Can make mistakes if the data changes slightly (overfitting).

Random Forest

- **How it works**: Builds many decision trees on random parts of the data, then combines their answers by voting.
- **Use case**: Detecting fraud in financial transactions.
- **Pros**: More accurate than a single decision tree, less likely to overfit.
- **Cons**: Harder to explain, takes more time to train.

2B. Extra Algorithm — Neural Networks

- **Problem solved**: Neural Networks are good at solving very complex problems where relationships between features are not obvious, such as images, speech, and big financial data.
- **How it works**: Inspired by the human brain, they use layers of "neurons." Each neuron takes inputs, applies weights, and passes the result forward. Multiple layers allow the network to learn deep patterns in the data.
- **Use case**: In healthcare, Neural Networks are used to classify medical images (e.g., detecting tumors in X-rays).
- **Pros**: Very powerful, can handle huge datasets, works well with complex data (images, audio, text).
- **Cons**: Requires a lot of data and computing power, harder to interpret compared to simple models like logistic regression.

3. Classification Metrics

- **Accuracy**: How many predictions were correct overall. Works well when classes are balanced.
- **Precision**: Of all predicted positives, how many were actually correct. Important when false alarms are costly.
- **Recall**: Of all actual positives, how many did the model find. Important when missing positives is costly.
- **F1-Score**: Combines precision and recall into one number. Good for imbalanced data.
- **Confusion Matrix**: A table showing where the model was correct and where it made mistakes.

Metric	**What it Focuses On**	**Best Use Case**	**Weakness**
Accuracy	Overall correctness	Balanced data	Misleading with imbalanced data
Precision	Correct positive predictions	When false positives are costly	Ignores missed positives
Recall	Capturing all positives	When false negatives are costly	Ignores false positives
F1-Score	Balance between precision/recall	Imbalanced data	Harder to explain
Confusion Matrix	Detailed breakdown of results	When we need error analysis	Not a single number

4. Imbalanced Data Problem

Imbalanced data happens when one class has many more examples than the other. For example, in predictive maintenance, only 5% of machines may fail while 95% run normally.

In this case, accuracy is misleading. A model that predicts “no failure” every time would be 95% accurate but still useless.

Better metrics include **Precision, Recall, F1-Score, and Confusion Matrix**, since they show how well the model handles the rare but important cases.

5. Real-World Case Study — Fraud Detection in Banking

Goal: A bank wanted to find fraudulent credit card transactions quickly to save money.

Data: Millions of transactions, with details like amount, location, and merchant type. Fraudulent transactions were marked in the data.

Model: The bank used Random Forest and Gradient Boosting. These models handled large amounts of data and found patterns of fraud.

Results: The models caught most fraud cases (high recall). Some normal transactions were flagged by mistake, but this was better than missing fraud. The bank saved millions of dollars by stopping fraud early.

References

- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O’Reilly Media.
- Scikit-learn documentation: https://scikit-learn.org