Introduction to Artificial Intelligence

2nd Webinar

September 4th, 2023

UNIVERSITY OF LATVIA

LIKTA
LATVIAN INFORMATION AND COMMUNICATIONS TECHNOLOGY ASSOCIATION

Powered by Microsoft

Lithuanian GitHub repository

- In Lithuanian.
- The presentations of Lithuanian and English comments under them.
- PowerPoint presentations and PDF of the presentation.
- Additional material in either Lithuanian or English.
- Graphics.

Repo: https://github.com/aurimas13/Pazink-Dirbtini-Intelekta/tree/main

Machine Learning

Machine Learning is

Subfield of Artificial Intelligence that focuses on creating algorithms and **models that enable computers to learn from and make predictions (or decisions) based on data.**

It allows machines to improve their performance on a specific task without being explicitly programmed, but rather by learning from examples and experience.

3 main types of Machine Learning

Supervised Learning - algorithms are trained on a labeled dataset, which consists of input-output pairs.

Unsupervised Learning - algorithms work with unlabeled data, where the true output is unknown.

Reinforcement Learning - an agent learns to make decisions by interacting with its environment

Supervised Learning

- Supervised learning is the most common type of Machine Learning
- Algorithms learn from a labeled dataset containing input-output pairs.
- The primary goal is to develop a model that can generalize from given examples and accurately predict the output for new, unseen data.
- Supervised learning can be further divided into two main tasks: classification and regression.

Supervised Learning

Input - Features | **Output- Target**

gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Female	80.0	0	1	never	25.19	6.6	140	0
Female	54.0	0	0	No Info	27.32	6.6	80	0
Male	28.0	0	0	never	27.32	5.7	158	0
Female	36.0	0	0	current	23.45	5.0	155	0
Male	76.0	1	1	current	20.14	4.8	155	0
Female	20.0	0	0	never	27.32	6.6	85	0
Female	44.0	0	0	never	19.31	6.5	200	1
Female	79.0	0	0	No Info	23.86	5.7	85	0
Male	42.0	0	0	never	33.64	4.8	145	0
Female	32.0	0	0	never	27.32	5.0	100	0
Female	53.0	0	0	never	27.32	6.1	85	0
Female	54.0	0	0	former	54.7	6.0	100	0
Female	78.0	0	0	former	36.05	5.0	130	0
Female	67.0	0	0	never	25.69	5.8	200	0
Female	76.0	0	0	No Info	27.32	5.0	160	0
Male	78.0	0	0	No Info	27.32	6.6	126	0
Male	15.0	0	0	never	30.36	6.1	200	0
Female	42.0	0	0	never	24.48	5.7	158	0
Female	42.0	0	0	No Info	27.32	5.7	80	0
Male	37.0	0	0	ever	25.72	3.5	159	0
Male	40.0	0	0	current	36.38	6.0	90	0
Male	5.0	0	0	No Info	18.8	6.2	85	0
Female	69.0	0	0	never	21.24	4.8	85	0
Female	72.0	0	1	former	27.94	6.5	130	0
Female	4.0	0	0	No Info	13.99	4.0	140	0

Classification

Classification involves categorizing input data into predefined classes or categories. For example, an email can be classified as spam or not spam, or a patient can be classified as someone who has a diabetes or not.

Classification problems usually deal with discrete, categorical outputs.

Regression

Regression is used for predicting continuous values, such as prices, temperatures, or sales figures.

The objective is to determine the relationship between input features and the target output variable, enabling the model to make predictions for new data points.

Unsupervised Learning

- Works with unlabeled data, where the true output or structure of the data is unknown
- The primary goal is to discover hidden patterns or structures within the data
- Two main tasks in unsupervised learning: clustering and dimensionality reduction.

Clustering

Clustering involves grouping similar data points together based on their features.

The objective is to find a natural partitioning in the data such that data points within the same cluster are more similar to each other than to those in other clusters.

Clustering can reveal meaningful relationships or structures within the data that may not be apparent otherwise.

Data Cleaning and Transformation

Importance of Data Preprocessing

- An important step in the machine learning pipeline that involves preparing and cleaning the raw data to ensure that it is suitable for ML algorithms.
- Often the data collected from various sources may contain inconsistencies, errors, and missing values.
- These issues, if not addressed, can significantly impact the performance and reliability of the machine learning models.

Importance of Data Pre-processing

- The primary goal of data pre-processing is to improve data quality and increase the efficiency of the model-building process.
- Data pre-processing helps to ensure that the machine learning algorithms can effectively learn and extract insights from the data.
- This process improves the accuracy, precision, and generalizability of the models, which results in more reliable predictions and better decision-making.

Benefits of data preprocessing

Improved Data Quality

Data pre-processing helps identify and correct errors, inconsistencies, and inaccuracies in the raw data. This results in cleaner and more reliable data.

Better data quality improves the performance of machine learning models.

Benefits of data preprocessing

Reduced Complexity

Pre-processed data is often less complex and easier to work with, as redundant features and unnecessary information are removed during the pre-processing stage.

This can lead to more efficient model training and getting to the optimal solution faster.

Benefits of data preprocessing

Improved Model Performance

Data preprocessing techniques, such as feature scaling and normalization, ensure that the input data is in a consistent format and range.

This helps to prevent biases in the model due to the dominance of certain features and enables algorithms to better capture the underlying patterns in the data.

Benefits of data preprocessing

Better Generalization

Feature selection and extraction can help to reduce the dimensionality of the data and focus on the most relevant features.

This results in models that are less prone to overfitting and can generalize better to new, unseen data.

Handling Missing Values

Handling Missing Values

Missing values are a common issue in real-world datasets.

They can arise for various reasons - errors in data collection, inconsistencies in data entry, or the unavailability of certain information.

Missing values can significantly impact the performance and reliability of machine learning models, as they can introduce biases, distort relationships between features, and reduce the overall quality of the data.

Strategies to Handle Missing Values

- Imputation Methods (Mean, Median, Mode imputation)
- Deletion Methods (Listwise, Pairwise deletion)
- Interpolation and Extrapolation

Feature Scaling and Normalization

Introduction to Feature Scaling and Normalization

Helps to bring different features within a dataset to a similar range or scale.

Machine learning algorithms often perform better and converge faster when all features are on a comparable scale, as it ensures that no single feature dominates the others, thereby avoiding biases in the model.

Many algorithms are sensitive to the scale of input features, making it necessary to apply feature scaling and normalization before training the models.

Min-Max Scaling

- Transforms features to range [0, 1]
- Useful for neural networks and k-Nearest Neighbors algorithms
- Formula: $x_{scaled} = \frac{(x - x_{min})}{(x_{max} - x_{min})}$

Standard Scaling (Z-score Normalization)

- Transforms features to have a mean of 0 and standard deviation of 1
- Useful for Support Vector Machines and Linear Regression
- Formula: $x_{scaled} = \frac{(x - \mu)}{\sigma}$ or $x_{scaled} = \frac{(x - x_{mean})}{(x_{standard\ deviation})}$

Applying Scaling Parameters Consistently

It is important to remember that the same scaling parameters used for the training data should also be applied to any new, unseen data for consistent model performance.

By transforming the input features to a consistent range or scale, ML models are more accurate, efficient, and less prone to biases caused by the dominance of certain features.

Model Evaluation Techniques

Introduction to Model Evaluation

Importance of model evaluation

- Model evaluation is a crucial step in the machine learning process, as it helps to determine the performance and effectiveness of a model.
- It provides an objective measure to compare different models and select the best one for a given problem.
- By evaluating a model's performance, one can identify potential issues, such as overfitting or underfitting, and take appropriate steps to address them.

The role of evaluation in the ML process

- Model evaluation is an important part of the iterative machine learning process, which typically involves the following steps: data preprocessing, feature selection, model training, model evaluation, and model optimization.
- After training a model on a dataset, it is essential to evaluate its performance on unseen data to ensure that it generalizes well to new instances.
- Model evaluation not only helps in selecting the best model but also guides further improvements and refinements in the machine learning process.
- By analyzing the evaluation results, one can gain insights into the model's strengths and weaknesses and identify the areas that need improvement. This information can be used to adjust the model's hyperparameters or even to select a different algorithm better suited for the problem at hand.

Training and test data

How the data are splitted?

- **Training data:** This is the largest portion of data used to train the machine learning model. It helps the model learn the underlying patterns and trends in the data. Usually, about 70-80% of the total data should be allocated for training.
- **Test data:** The test data is used to evaluate the model's final performance after training. This data should remain untouched during the model development process and only be used once to estimate the model's real-world performance. Typically, around 20-30% of the available data is reserved for testing purposes.

The primary goal of using test data is to evaluate the model's performance on unseen data, ensuring that it generalizes well to new instances.

Separate validation set

If dataset is split in 3 parts:

- Training data are 60 to 70%
- Validation data are 15% to 20%
- Test data are 15% to 20%

By using three splits, you significantly reduce the risk of overfitting and improve the model's ability to generalize to unseen data. The additional validation set assists in finding an optimal balance between model complexity and performance, leading to the development of more reliable and accurate machine learning models.

Cross-validation

Overview of cross-validation

- Cross-validation is an evaluation technique that helps to assess a model's performance more accurately and reliably.
- It involves dividing the dataset into multiple subsets or "folds" and then training and evaluating the model multiple times, using different combinations of these folds.
- By averaging the performance metrics of each iteration, cross-validation provides a more reliable estimate of the model's performance on unseen data.

Types of cross-validation

k-fold cross-validation: In this method, the dataset is divided into 'k' equally sized folds. The model is trained and evaluated 'k' times, each time using one of the folds as the testing set and the remaining 'k-1' folds as the training set.

The performance metrics are then averaged to obtain the final evaluation score.

Common choices for 'k' are 5 or 10, as these values have been shown to provide a good balance between accuracy and computational efficiency.

k-fold cross-validation

gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Female	80.0	0	1	never	25.19	6.6	140	0
Female	54.0	0	0	No Info	27.32	6.6	80	0
Male	28.0	0	0	never	27.32	5.7	158	0
Female	36.0	0	0	current	23.45	5.0	155	0
Male	76.0	1	1	current	20.14	4.8	155	0
Female	20.0	0	0	never	27.32	6.6	85	0
Female	44.0	0	0	never	19.31	6.5	200	1
Female	79.0	0	0	No Info	23.86	5.7	85	0
Male	42.0	0	0	never	33.64	4.8	145	0
Female	32.0	0	0	never	27.32	5.0	100	0
Female	53.0	0	0	never	27.32	6.1	85	0
Female	54.0	0	0	former	54.7	6.0	100	0
Female	78.0	0	0	former	36.05	5.0	130	0
Female	67.0	0	0	never	25.69	5.8	200	0
Female	76.0	0	0	No Info	27.32	5.0	160	0
Male	78.0	0	0	No Info	27.32	6.6	126	0
Male	15.0	0	0	never	30.36	6.1	200	0
Female	42.0	0	0	never	24.48	5.7	158	0

k-fold cross-validation

gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Female	80.0	0	1	never	25.19	6.6	140	0
Female	54.0	0	0	No Info	27.32	6.6	80	0
Male	28.0	0	0	never	27.32	5.7	158	0
Female	36.0	0	0	current	23.45	5.0	155	0
Male	76.0	1	1	current	20.14	4.8	155	0
Female	20.0	0	0	never	27.32	6.6	85	0
Female	44.0	0	0	never	19.31	6.5	200	1
Female	79.0	0	0	No Info	23.86	5.7	85	0
Male	42.0	0	0	never	33.64	4.8	145	0
Female	32.0	0	0	never	27.32	5.0	100	0
Female	53.0	0	0	never	27.32	6.1	85	0
Female	54.0	0	0	former	54.7	6.0	100	0
Female	78.0	0	0	former	36.05	5.0	130	0
Female	67.0	0	0	never	25.69	5.8	200	0
Female	76.0	0	0	No Info	27.32	5.0	160	0
Male	78.0	0	0	No Info	27.32	6.6	126	0
Male	15.0	0	0	never	30.36	6.1	200	0
Female	42.0	0	0	never	24.48	5.7	158	0

k-fold cross-validation

gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Female	80.0	0	1	never	25.19	6.6	140	0
Female	54.0	0	0	No Info	27.32	6.6	80	0
Male	28.0	0	0	never	27.32	5.7	158	0
Female	36.0	0	0	current	23.45	5.0	155	0
Male	76.0	1	1	current	20.14	4.8	155	0
Female	20.0	0	0	never	27.32	6.6	85	0
Female	44.0	0	0	never	19.31	6.5	200	1
Female	79.0	0	0	No Info	23.86	5.7	85	0
Male	42.0	0	0	never	33.64	4.8	145	0
Female	32.0	0	0	never	27.32	5.0	100	0
Female	53.0	0	0	never	27.32	6.1	85	0
Female	54.0	0	0	former	54.7	6.0	100	0
Female	78.0	0	0	former	36.05	5.0	130	0
Female	67.0	0	0	never	25.69	5.8	200	0
Female	76.0	0	0	No Info	27.32	5.0	160	0
Male	78.0	0	0	No Info	27.32	6.6	126	0
Male	15.0	0	0	never	30.36	6.1	200	0
Female	42.0	0	0	never	24.48	5.7	158	0

k-fold cross-validation

gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Female	80.0	0	1	never	25.19	6.6	140	0
Female	54.0	0	0	No Info	27.32	6.6	80	0
Male	28.0	0	0	never	27.32	5.7	158	0
Female	36.0	0	0	current	23.45	5.0	155	0
Male	76.0	1	1	current	20.14	4.8	155	0
Female	20.0	0	0	never	27.32	6.6	85	0
Female	44.0	0	0	never	19.31	6.5	200	1
Female	79.0	0	0	No Info	23.86	5.7	85	0
Male	42.0	0	0	never	33.64	4.8	145	0
Female	32.0	0	0	never	27.32	5.0	100	0
Female	53.0	0	0	never	27.32	6.1	85	0
Female	54.0	0	0	former	54.7	6.0	100	0
Female	78.0	0	0	former	36.05	5.0	130	0
Female	67.0	0	0	never	25.69	5.8	200	0
Female	76.0	0	0	No Info	27.32	5.0	160	0
Male	78.0	0	0	No Info	27.32	6.6	126	0
Male	15.0	0	0	never	30.36	6.1	200	0
Female	42.0	0	0	never	24.48	5.7	158	0

k-fold cross-validation

gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Female	80.0	0	1	never	25.19	6.6	140	0
Female	54.0	0	0	No Info	27.32	6.6	80	0
Male	28.0	0	0	never	27.32	5.7	158	0
Female	36.0	0	0	current	23.45	5.0	155	0
Male	76.0	1	1	current	20.14	4.8	155	0
Female	20.0	0	0	never	27.32	6.6	85	0
Female	44.0	0	0	never	19.31	6.5	200	1
Female	79.0	0	0	No Info	23.86	5.7	85	0
Male	42.0	0	0	never	33.64	4.8	145	0
Female	32.0	0	0	never	27.32	5.0	100	0
Female	53.0	0	0	never	27.32	6.1	85	0
Female	54.0	0	0	former	54.7	6.0	100	0
Female	78.0	0	0	former	36.05	5.0	130	0
Female	67.0	0	0	never	25.69	5.8	200	0
Female	76.0	0	0	No Info	27.32	5.0	160	0
Male	78.0	0	0	No Info	27.32	6.6	126	0
Male	15.0	0	0	never	30.36	6.1	200	0
Female	42.0	0	0	never	24.48	5.7	158	0

Types of cross-validation

Leave-one-out cross-validation: This is a special case of k-fold cross-validation, where 'k' is equal to the number of instances (rows) in the dataset.

In each iteration, a single row is used as the testing set, while the remaining instances form the training set.

This method is computationally expensive but provides the most unbiased estimate of the model's performance, especially for small datasets.

Types of cross-validation

gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Female	80.0	0	1	never	25.19	6.6	140	0
Female	54.0	0	0	No Info	27.32	6.6	80	0
Male	28.0	0	0	never	27.32	5.7	158	0
Female	36.0	0	0	current	23.45	5.0	155	0
Male	76.0	1	1	current	20.14	4.8	155	0
Female	20.0	0	0	never	27.32	6.6	85	0
Female	44.0	0	0	never	19.31	6.5	200	1
Female	79.0	0	0	No Info	23.86	5.7	85	0
Male	42.0	0	0	never	33.64	4.8	145	0
Female	32.0	0	0	never	27.32	5.0	100	0
Female	53.0	0	0	never	27.32	6.1	85	0
Female	54.0	0	0	former	54.7	6.0	100	0
Female	78.0	0	0	former	36.05	5.0	130	0
Female	67.0	0	0	never	25.69	5.8	200	0
Female	76.0	0	0	No Info	27.32	5.0	160	0
Male	78.0	0	0	No Info	27.32	6.6	126	0
Male	15.0	0	0	never	30.36	6.1	200	0
Female	42.0	0	0	never	24.48	5.7	158	0

Types of cross-validation

gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Female	80.0	0	1	never	25.19	6.6	140	0
Female	54.0	0	0	No Info	27.32	6.6	80	0
Male	28.0	0	0	never	27.32	5.7	158	0
Female	36.0	0	0	current	23.45	5.0	155	0
Male	76.0	1	1	current	20.14	4.8	155	0
Female	20.0	0	0	never	27.32	6.6	85	0
Female	44.0	0	0	never	19.31	6.5	200	1
Female	79.0	0	0	No Info	23.86	5.7	85	0
Male	42.0	0	0	never	33.64	4.8	145	0
Female	32.0	0	0	never	27.32	5.0	100	0
Female	53.0	0	0	never	27.32	6.1	85	0
Female	54.0	0	0	former	54.7	6.0	100	0
Female	78.0	0	0	former	36.05	5.0	130	0
Female	67.0	0	0	never	25.69	5.8	200	0
Female	76.0	0	0	No Info	27.32	5.0	160	0
Male	78.0	0	0	No Info	27.32	6.6	126	0
Male	15.0	0	0	never	30.36	6.1	200	0
Female	42.0	0	0	never	24.48	5.7	158	0

Types of cross-validation

gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Female	80.0	0	1	never	25.19	6.6	140	0
Female	54.0	0	0	No Info	27.32	6.6	80	0
Male	28.0	0	0	never	27.32	5.7	158	0
Female	36.0	0	0	current	23.45	5.0	155	0
Male	76.0	1	1	current	20.14	4.8	155	0
Female	20.0	0	0	never	27.32	6.6	85	0
Female	44.0	0	0	never	19.31	6.5	200	1
Female	79.0	0	0	No Info	23.86	5.7	85	0
Male	42.0	0	0	never	33.64	4.8	145	0
Female	32.0	0	0	never	27.32	5.0	100	0
Female	53.0	0	0	never	27.32	6.1	85	0
Female	54.0	0	0	former	54.7	6.0	100	0
Female	78.0	0	0	former	36.05	5.0	130	0
Female	67.0	0	0	never	25.69	5.8	200	0
Female	76.0	0	0	No Info	27.32	5.0	160	0
Male	78.0	0	0	No Info	27.32	6.6	126	0
Male	15.0	0	0	never	30.36	6.1	200	0
Female	42.0	0	0	never	24.48	5.7	158	0

Types of cross-validation

gender	age	hypertension	heart_disease	smoking_history	bmi	HbA1c_level	blood_glucose_level	diabetes
Female	80.0	0	1	never	25.19	6.6	140	0
Female	54.0	0	0	No Info	27.32	6.6	80	0
Male	28.0	0	0	never	27.32	5.7	158	0
Female	36.0	0	0	current	23.45	5.0	155	0
Male	76.0	1	1	current	20.14	4.8	155	0
Female	20.0	0	0	never	27.32	6.6	85	0
Female	44.0	0	0	never	19.31	6.5	200	1
Female	79.0	0	0	No Info	23.86	5.7	85	0
Male	42.0	0	0	never	33.64	4.8	145	0
Female	32.0	0	0	never	27.32	5.0	100	0
Female	53.0	0	0	never	27.32	6.1	85	0
Female	54.0	0	0	former	54.7	6.0	100	0
Female	78.0	0	0	former	36.05	5.0	130	0
Female	67.0	0	0	never	25.69	5.8	200	0
Female	76.0	0	0	No Info	27.32	5.0	160	0
Male	78.0	0	0	No Info	27.32	6.6	126	0
Male	15.0	0	0	never	30.36	6.1	200	0
Female	42.0	0	0	never	24.48	5.7	158	0

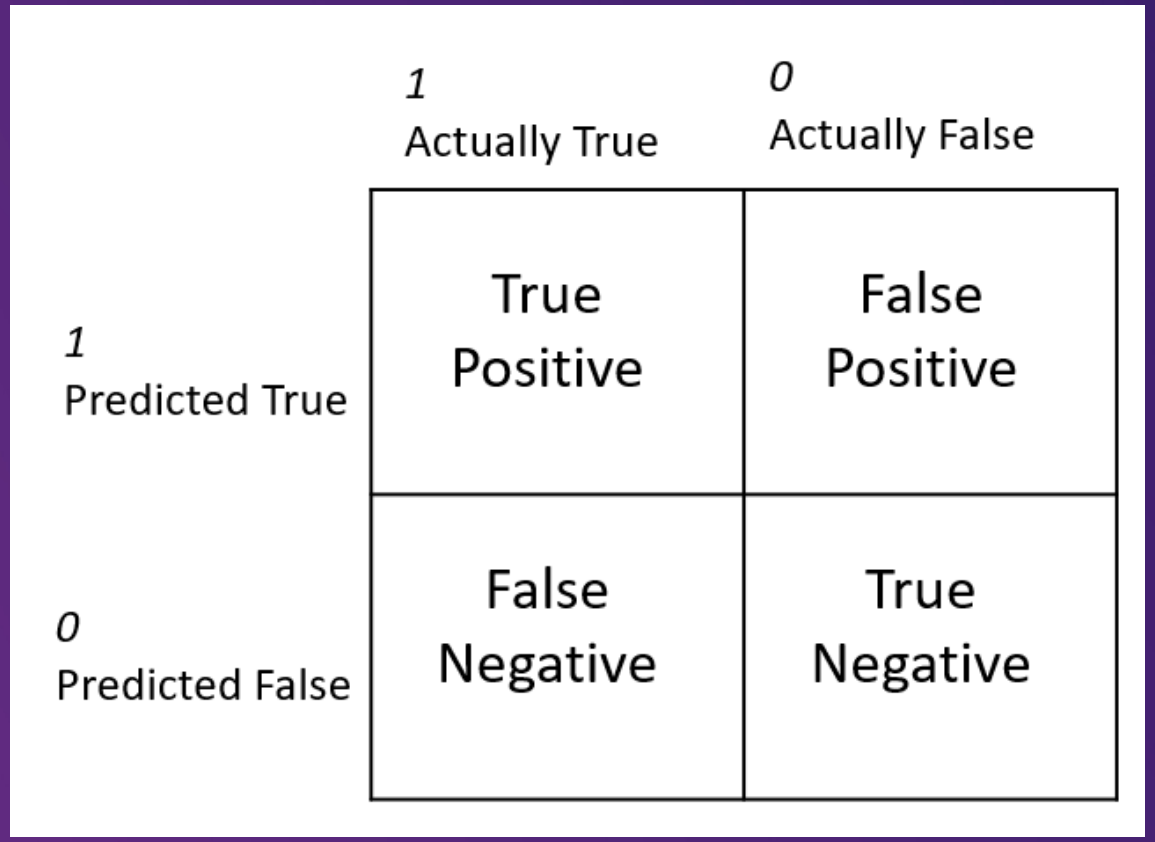
Benefits of cross-validation

- Cross-validation helps to reduce the risk of overfitting and provides a more reliable estimate of the model's performance on new, unseen data.
- It utilizes the entire dataset for both training and evaluation, ensuring that the model is exposed to a variety of instances and reducing the likelihood of introducing biases due to data splitting.
- Cross-validation allows for a more comprehensive understanding of the model's performance, as it provides information on the variability and stability of the model's predictions across different subsets of the data.

Performance metrics for classification models

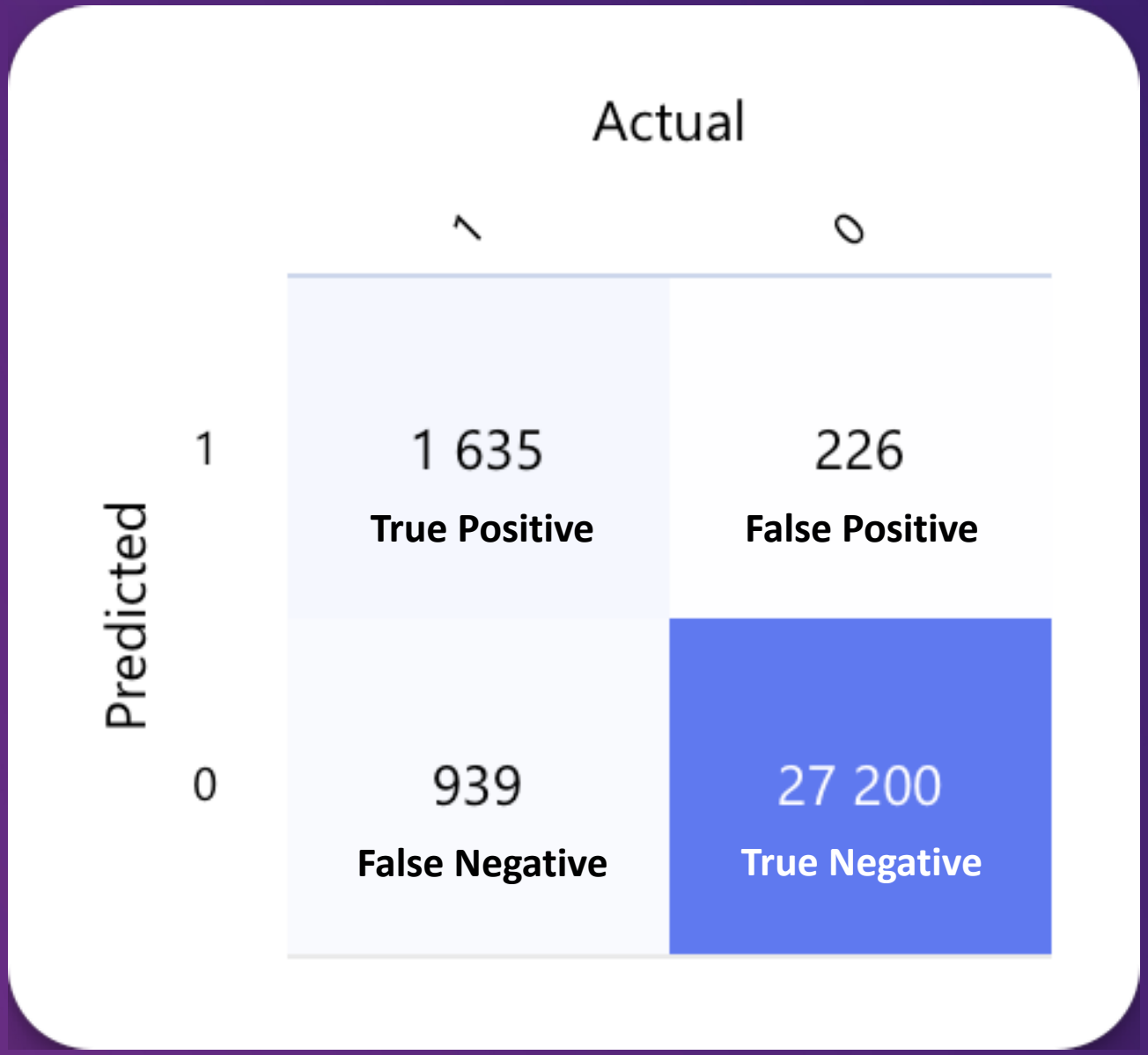
Confusion matrix

- A confusion matrix is a table that presents the number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by a classification model.
- Compares predicted labels vs. actual labels
- It is a useful tool for visualizing and understanding the model's performance



Confusion matrix

Suppose there are 30 000 patients in the dataset, with 2574 patients actually having diabetes and 27426 not having diabetes.
If the model predicts 1861 patients have diabetes, and 1635 of them are true positives, the confusion matrix can be visualized like this ->



Accuracy, precision, recall, and F1-score

Accuracy: The proportion of correct predictions (both true positives and true negatives) out of the total number of predictions made. It is a commonly used metric but can be misleading in cases of imbalanced class distributions.

Precision: The proportion of true positive predictions out of the total number of positive predictions made**.** It measures the model's ability to correctly identify positive instances and avoid false positives.

Recall: The proportion of true positive predictions out of the total number of actual positive instances. It measures the model's ability to identify all the positive instances.

F1-score: The harmonic mean of precision and recall, providing a balanced measure of both metrics. It is particularly useful when dealing with imbalanced datasets, as it takes both false positives and false negatives into account.

Formulas for the performance metrics

1. Accuracy = (True Positives + True Negatives) / (True Positives + False Positives + False Negatives + True Negatives)

2. Precision = True Positives / (True Positives + False Positives)

3. Recall = True Positives / (True Positives + False Negatives)

4. F1-score = 2 * (Precision * Recall) / (Precision + Recall)

Formulas for the performance metrics

1. Accuracy:

Formula: Accuracy = (True Positives + True Negatives) / (True Positives + False Positives + False Negatives + True Negatives)

Guideline: An accuracy above 90% is generally considered good. However, in imbalanced datasets, accuracy can be misleading, so it is essential to evaluate other metrics as well.

2. Precision:

Formula: Precision = True Positives / (True Positives + False Positives)

Guideline: A precision above 80% is considered good, but it depends on the specific problem and dataset. A high precision is essential when the cost of false positives is considerable.

Formulas for the performance metrics

3. Recall:

Formula: Recall = True Positives / (True Positives + False Negatives)

Guideline: A recall above 80% is considered good, but like precision, it depends on the specific problem and dataset. A high recall is crucial when identifying positive instances is more important than avoiding false positives.

4. F1-score:

Formula: F1-score = 2 * (Precision * Recall) / (Precision + Recall)

Guideline: An F1-score above 0.8 is considered good; however, it depends on the context of the problem and dataset. An F1-score above 0.9 is excellent, while an F1-score between 0.6 and 0.8 is moderate, and below 0.6 is considered poor.

Summary of the webinar

- Before you start to to train your ML model, do the data pre-processing
- Make sure your data has the most important features and they have similar scales
- Split the data in (at least) two parts – training data and test data (70:30)
- Make sure you try different ML algorithms
- Evaluate and pick the best model for your usecase