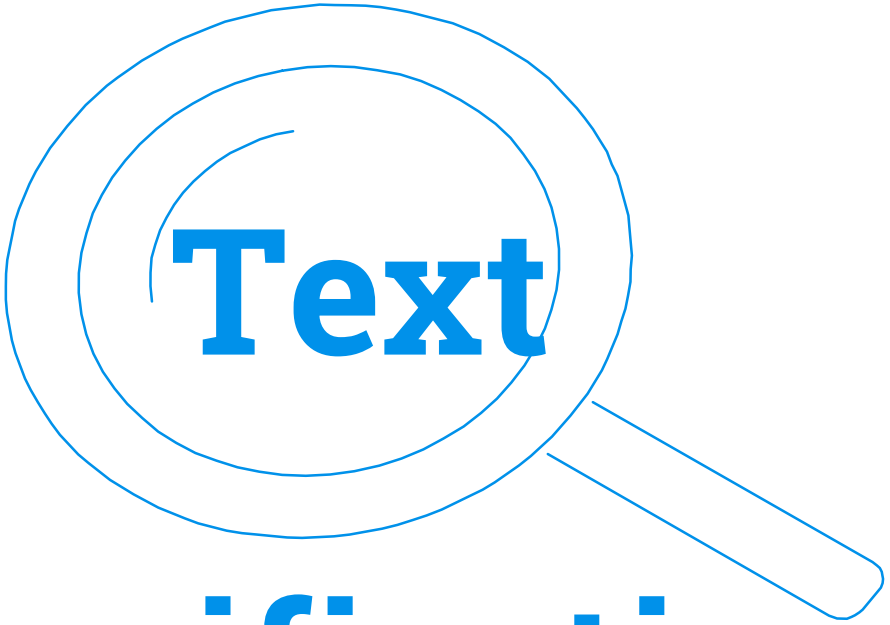


# Text Classification

SOFIA UNIVERSITY
ST. KLIMENT OHRIDSKI
EST. 1888

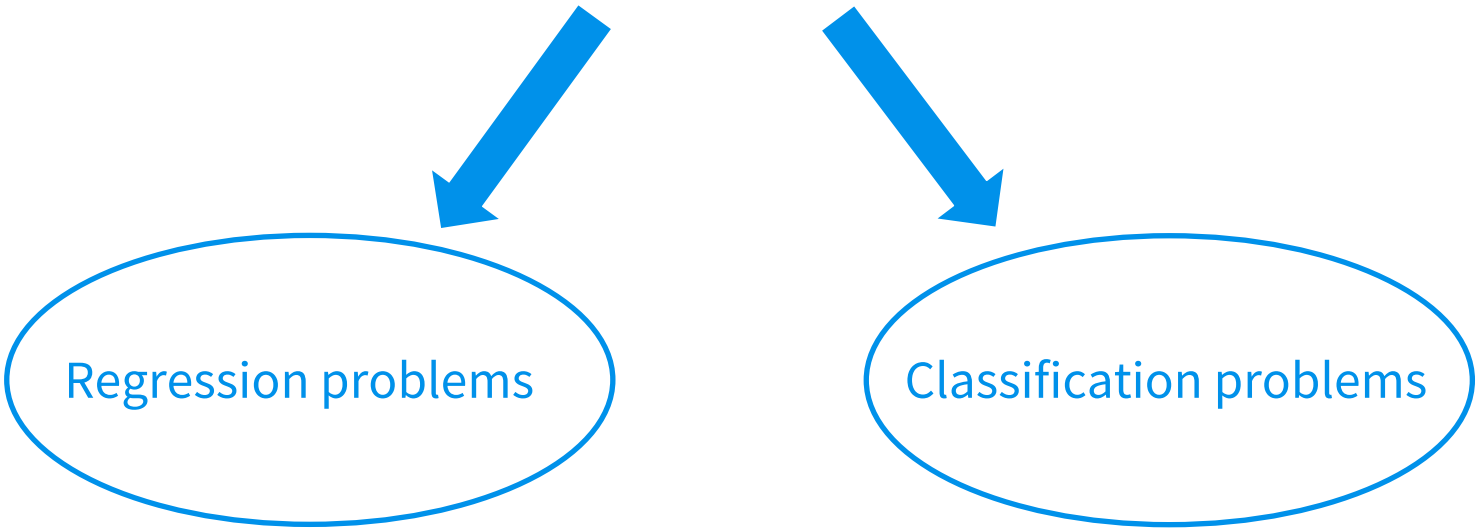
FACULTY
OF ECONOMICS
AND BUSINESS
ADMINISTRATION

## What you will learn

- What is **text classification**? 
- How to create a model which    **predicts** the customer sentiment?
- The **Naïve Bayes** model. 
- Key metrics for model evaluation 
- Practical examples in Python. 

## But…what is “text classification” exactly?

Two main types of machine learning problems:



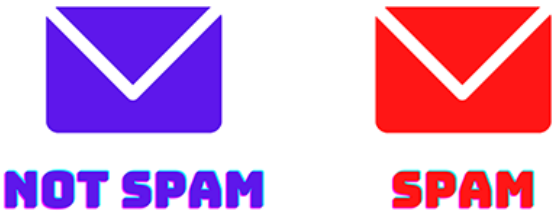
# Regression problems in ML

- Regression - the task of predicting a **continuous target variable**.


Credits: https://propertyinvest.com.tr/factors-that-affect-the-real-estate-prices-in-istanbul/ ; https://www.freepik.com/premium-vector/price-increase-consumer-population-income-schedule-illustration_7825988.htm ; https://www.pinterest.com/pin/507429082998526317/

# Classification problems in ML

- Classification - the task of predicting a **categorical target variable**.



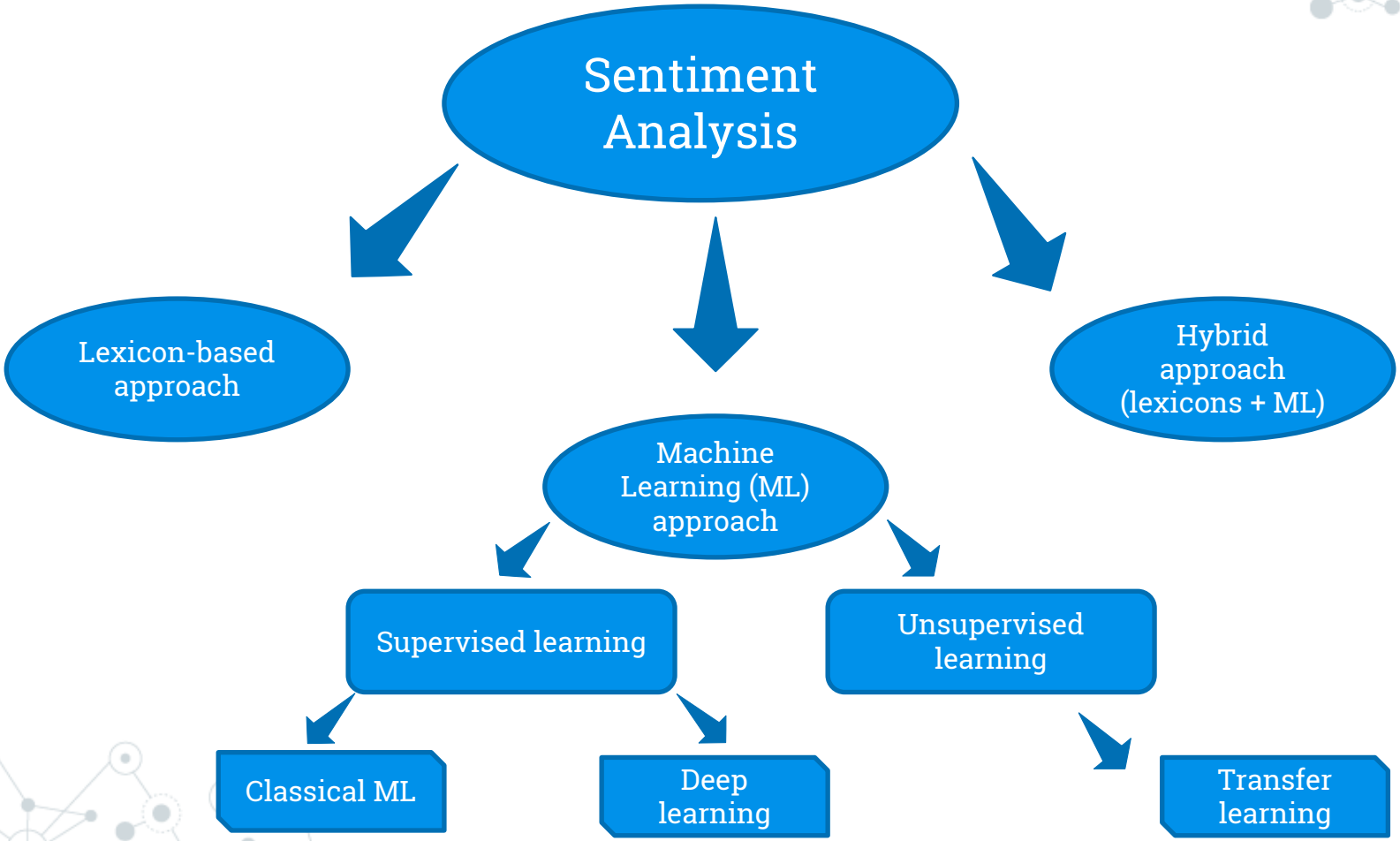


Credits: https://medium.com/analytics-vidhya/what-is-the-confusion-matrix-and-how-to-use-it-6dd63fab33ed; https://capital.com/credit-risk-definition ; https://xpressoai.medium.com/sentiment-analytics-37138aa5c1c0

## Common text classification problems

- **Sentiment**/**emotion** analysis;
- **Customer service request** classification;
- Detection of **toxic behavior** in the internet;
- Classification of **helpful comments**/**reviews** in forums;
- **Fake news** detection;
- Finding opinion **manipulation trolls;**
- Applications in the **medical domain – for example** - Sensors | Free Full-Text | Schizophrenia Detection Using Machine Learning Approach from Social Media Content | HTML (mdpi.com)

# Sentiment analysis as a classification problem



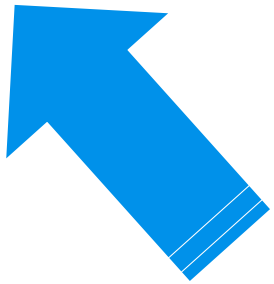
# Supervised vs. Unsupervised learning

## Supervised learning

Predict the value of the target variable based on historical data**. You have training (labeled) data**!

## Unsupervised learning

You don't have any previous knowledge about the target variable values (**no training data available**).



## Sentiment classification – the use case (1)

- The dataset - **customer reviews** for mobile applications (scraped from Google Play).
- The dataset contains the following information:
  - Text with the customer review.
  - Name of the mobile app.
  - Customer service rating (expressed in **5 ☆ scale**).
  - Date
- **Reference**: Android Apps and User Feedback: A Dataset for Software Evolution and Quality Improvement (uzh.ch)

# Sentiment classification – the use case (2)

Google Play


Credits :https://icon-library.com/icon/channels-icon-3.html ; https://gameworldobserver.com/2019/05/14/google-play-store-rating-latest-reviews

# Sentiment classification – the use case (3)

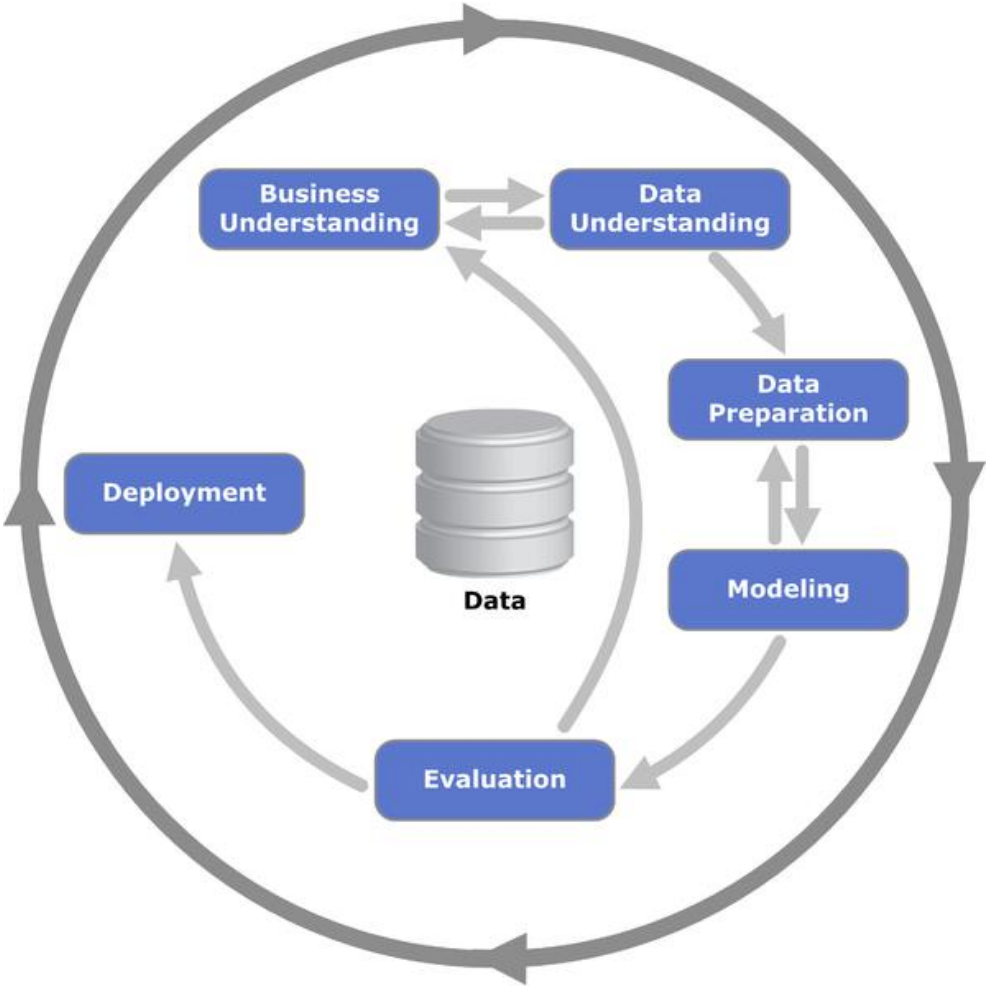
Historical data from Google Play


Create a sentiment classification model using ML

Apply the model on data from other communication channels

Credits: https://gameworldobserver.com/2019/05/14/google-play-store-rating-latest-reviews; https://icon-library.com/icon/omni-channel-icon-23.html ; https://xpressoai.medium.com/sentiment-analytics-37138aa5c1c0

# The CRISP-DM model






Source: https://bg.m.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:CRISP-DM_Process_Diagram.png

## Problem/Business understanding (1)

- Build a **sentiment classification model** based on app reviews posted in Google Play.
- Predict the customer sentiment in its **polarity:**


## Problem/Business understanding (2)

- How to **build the target variable**?
- The answer: use **distant supervision**.

**The negative category:**

**The positive category:**

**The negative category:**


**The positive category:**


- **Possible problems with this approach**:

"Great app….." 

## Data Understanding

- Use **graphical analysis** in order to get familiar with data – word clouds, box plot diagrams and various visualizations.

## Data Preparation

- Text data cleaning and normalization are **crucial in any text mining project**!
- Some important **techniques**:
  - Special characters removal;
  - Case normalization;
  - Removal of URLs;
  - Removal of html tags;
  - Stemming/Lemmatization etc.

## Data Modeling – the feature engineering stage (1)

- Classical ML models require **feature engineering**.
- Main source of explanatory variables – **the text.**
- Usage of $n$**-grams** – unigrams, bigrams etc.
- Be aware: increasing $n$ might lead to **model overfit**!

## Data Modeling – the feature engineering stage (2)

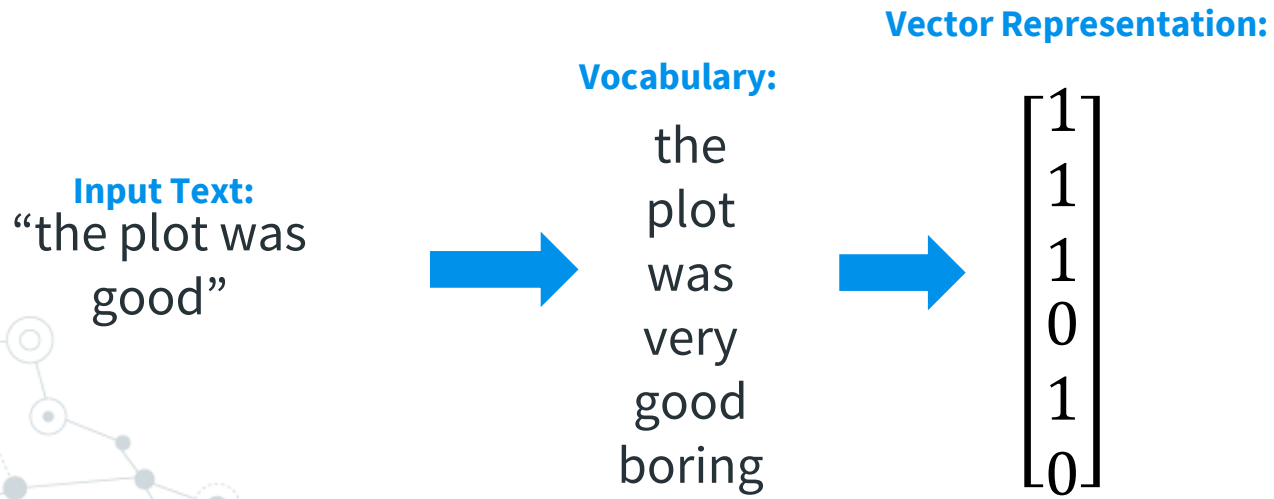
Other potential explanatory variables:

- **Emoticons -**   
- **Punctuation** – “!!!!!”, “?”, “?!” etc.
- **Word capitalization –** “COOL”
- **Available metadata –** time, location etc.
- And other..

**NB:** some features should be extracted before text processing!

## Data Modeling – text vectorization

- The **vector space model**
- Three **main forms of text vectorization**:
  - Binary vectorization
  - Count vectorization
  - TF-IDF vectorization



## Data Modeling – feature selection

- **Feature selection…Why?**
  - Select explanatory variables which have more predictive power compared to others;
  - Dimension reduction;
  - Reduce noise and irrelevant information.
- Different **techniques**:
  - Filtering variables according to frequency.
  - Mutual Information (example with “freeze”).
  - Chi-square test.
  - And other.

## Data Modeling – text classification algorithms

- **Logistic regression** – read more in Gareth, J. et al. An introduction to statistical learning: with applications in R. Spinger, 2013. – **Chapter 4**
- **Support vector classifier (SVC) –** read more in Gareth, J. et al. An introduction to statistical learning: with applications in R. Spinger, 2013. – **Chapter 9**
- **Naïve Bayes Model -** Introduction to Information Retrieval (stanford.edu), Chapter 13

## The Naïve Bayes model (1)

- **Bayes' Theorem:**

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad (1)$$

- $A, B$ – two events.
- $P(A|B)$ - probability of $A$ given $B$.
- $P(B|A)$ - probability of $B$ given $A$.
- $P(A)$ and $P(B)$ – independent probabilities of $A$ and $B, P(B) \neq 0$.
- $P(A)$ is **prior** probability, while $P(A|B)$ is **posterior** probability.

## The Naïve Bayes model (2)

$$P(y|d) = \frac{P(d|y)P(y)}{P(d)} \qquad (2)$$

$d$ – a given client review, $d \in \{1, \ldots, M\}$).

$y$ – the class of the target vairable (positive or negative sentiment),

$y = \{0,1\}$.

- Naïve Bayes Classifier types:
  - Multinomial.
  - Bernoulli.
  - Gaussian.

## The Naïve Bayes model (3)

- The **"naïve" assumption** of the Naïve Bayes model – all explanatory variables are independent.
- This assumption is **rarely true** especially when we work with text data!
- The independence assumption is defined as follows:

$$P(A \cap B) = P(A) \times P(B) \qquad (3)$$

## The Naïve Bayes model (4)

$$P(y|d) = P(w_1|y)P(w_2|y) \dots P(w_v|y)P(y) \qquad (4)$$

$y$ – the class of the target vairable (positive or negative), $y = \{0,1\}$.

$W = (w_1, w_2 \dots w_v)$

$V$ – number of explanatory variables (words, for example).

$$P(y|d) = P(y) \prod_{i=1}^{V} P(w_i|y) \qquad (5)$$

## The Naïve Bayes model (5)

- The **Bernoulli Naïve Bayes** model solves the following optimization task:

$$\hat{y} = \underset{y}{argmax}\, P(y) \prod_{i=1}^{V} P(w_i|y) \qquad (6)$$

## Model validation.. Why?

- **Model validation** – assess model's performance on unseen data.
- **Techniques**:
  - The validation set approach.
  - Leave-one-out cross-validation.
  - K-fold cross-validation.
- Check out **Chapter 5** in Gareth, J. et al. An introduction to statistical learning: with applications in R. Spinger, 2013.

## Model validation – the confusion matrix

- Confusion matrix for 2-class target variable:

| | | Predicted Values | | |
|---|---|---|---|---|
| | | “Positive” category | “Negative” category | |
| Actual Values | “Positive” category | True positive (TP) | False Negative (FN) | Total truly positive (TTP) |
| | “Negative” category | False positive (FP) | True negative (TN) | Total Truly negative (TTN) |

# Model validation – evaluation metrics

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{F1 score} = 2\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

## If you want to learn more... 🤓

- Gareth, J. et al. An introduction to statistical learning: with applications in R. Spinger, 2013. – Chapter 4, 5, 9.
- Introduction to Information Retrieval (stanford.edu), Chapter 13

# Thanks!

## Any questions?

You can find me at:

g.hristova@feb.uni-sofia.bg