Master’s Degree
in
Data Science

MACHINE LEARNING OPTIMISATION OF SOFTWARE DEVELOPMENT PROCESSES: A REAL-CASE APPLICATION OVER TICKET ISSUING DATA

Supervisors
IVANO BISON
DOTT. FABIO CELLI

Candidate
LEONARDO PAJER

Academic Year
2021/2022

Structure of the presentation

- Key concepts
- Case-study
- Limits and further research

Key concepts

- **Information flow**: exchange of information among people, processes, and systems within an organisation.
- **Ticket issuing system**: software which assigns to business processes labels to control the workflow
- **Artificial Intelligence**: the theory and development of computer systems able to perform tasks normally requiring human intelligence (Oxford Languages)
- **Machine Learning**: the process of computers changing the way they carry out tasks by learning from new data, without a human being needing to give instructions in the form of a program (Cambridge Dictionary)

CASE-STUDY

Objectives

- **«Hard» objective**: adopting Artificial Intelligence with the goal of predicting the probability that a ticket issue, related to software development tasks, will be solved within the assigned deadline given at the time of its creation

- **«Soft» objectives**: improving information flow and investigate the dynamics underlying software development processes

Project’s Framework

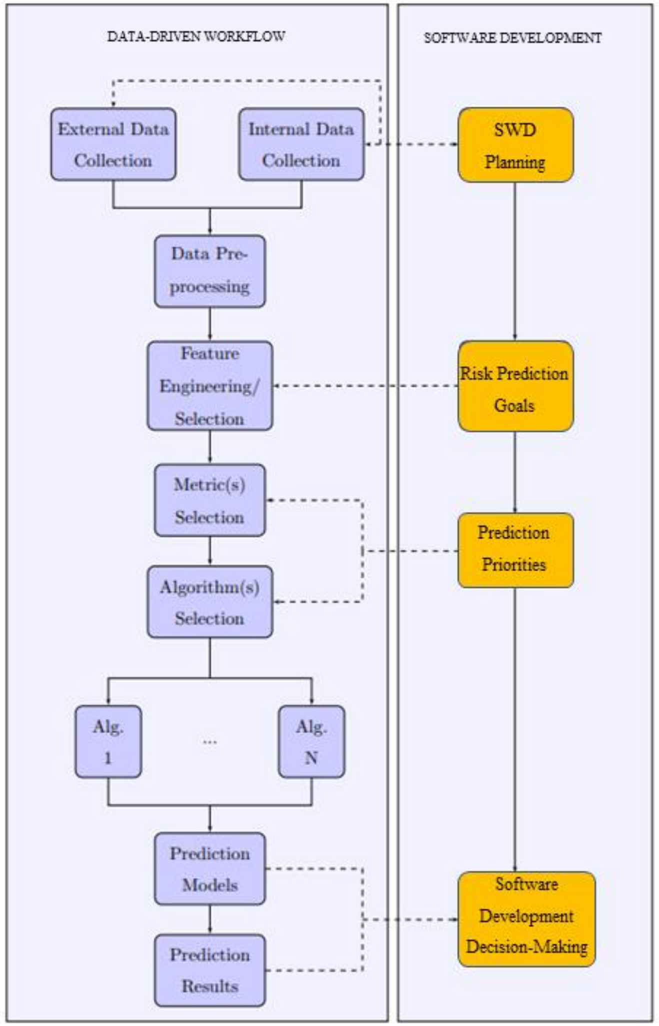


Figure 1. Workflow pipeline

Dataset 1/2

- Shape: 32296 rows and 7 columns.
- Oversampling approaches were used to deal with class imbalance in the target feature (94.5%; 4.5%)
- The dataset's columns indicate the final attributes “fed” to the models, rows corresponds to Software Development ticket issues
- Data were collected from January 1st, 2020 to February 4th, 2021.
- Train/test split: 0.7/0.3

Dataset 2/2: target and features

- **Target (KPI)**: binary variable, it describes wheather a ticket issue has been solved within the desired deadline.

- **Features:** Priority, Month_creation, Azienda, Risk, Area, Project_category.

Method

- Oversampling (SMOTE)
- Feature Engineering: creation of Risk from issues type
- Cross-Validation for hyperparameters optimisation according to Recall metric
- Validated models: Random Forest Classifier, Logistic Regression, XGBoost

Results 1/2: «Hard» goals

= best params
= recall score

mean_test_precision_score	mean_test_recall_score	mean_test_accuracy_score	param_max_depth	param_gamma	param_n_estimators
0.825	0.923	0.864	25	1	300
0.825	0.923	0.864	25	1	100
0.825	0.922	0.863	15	1	300
0.825	0.922	0.863	15	1	100
0.826	0.922	0.864	25	0.3	100

Table 1. Top 5 scores XGBoost

mean_test_precision_score	mean_test_recall_score	mean_test_accuracy_score	param_max_depth	param_max_features	param_min_samples_split	param_n_estimators
0.826	0.923	0.864	15	5	5	100
0.826	0.923	0.864	15	3	3	100
0.826	0.922	0.864	15	3	5	100
0.826	0.922	0.864	15	2	5	100
0.825	0.922	0.863	15	5	3	100

Table 2: top 5 scores Random Forest

mean_test_precision_score	mean_test_recall_score	mean_test_accuracy_score	param_max_iter	param_C	param_penalty
0.769	0.788	0.776	100	0.5	l2
0.769	0.788	0.776	150	0.5	l2
0.769	0.788	0.776	100	1	l2
0.769	0.788	0.776	150	1	l2
0.769	0.788	0.776	100	0.3	l2

Table 3 Top 5 scores Logistic Regression

Results 2/2: «Soft» goals

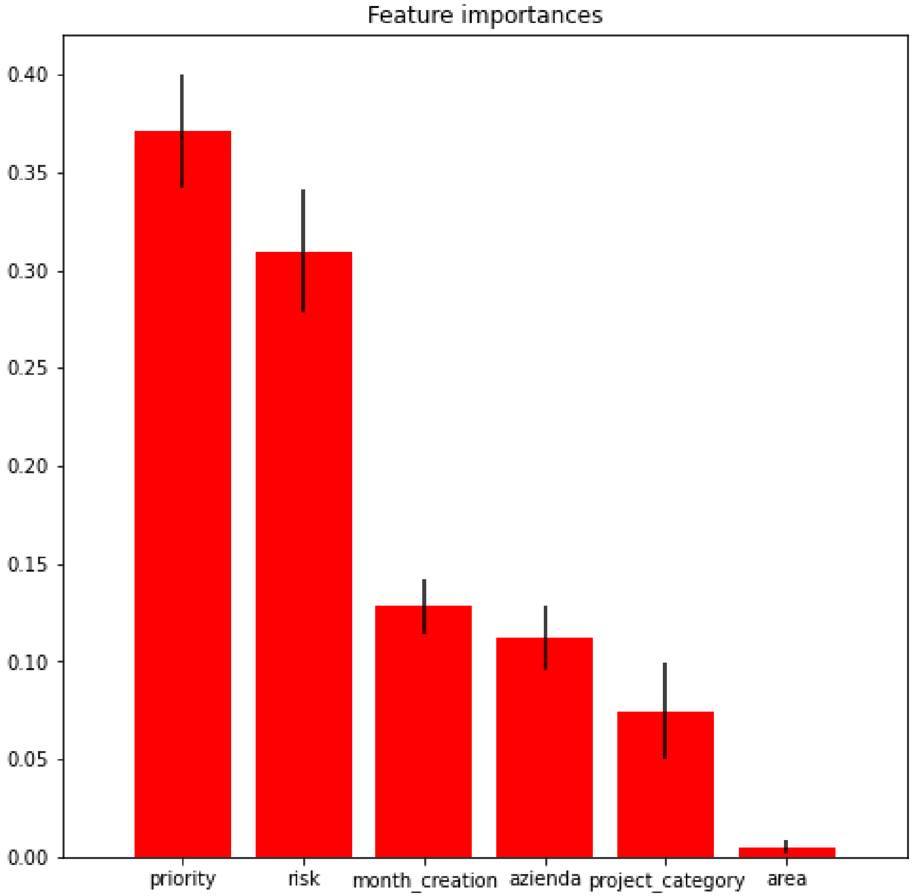


Figure 1. Random forest feature importance

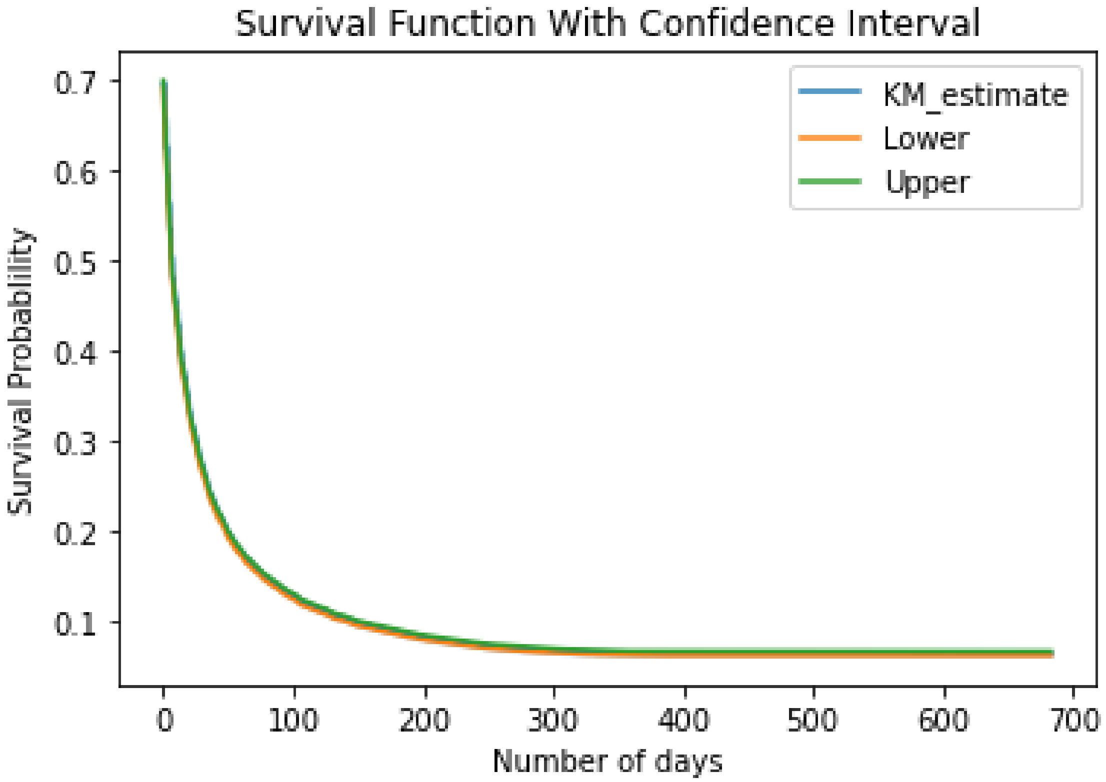


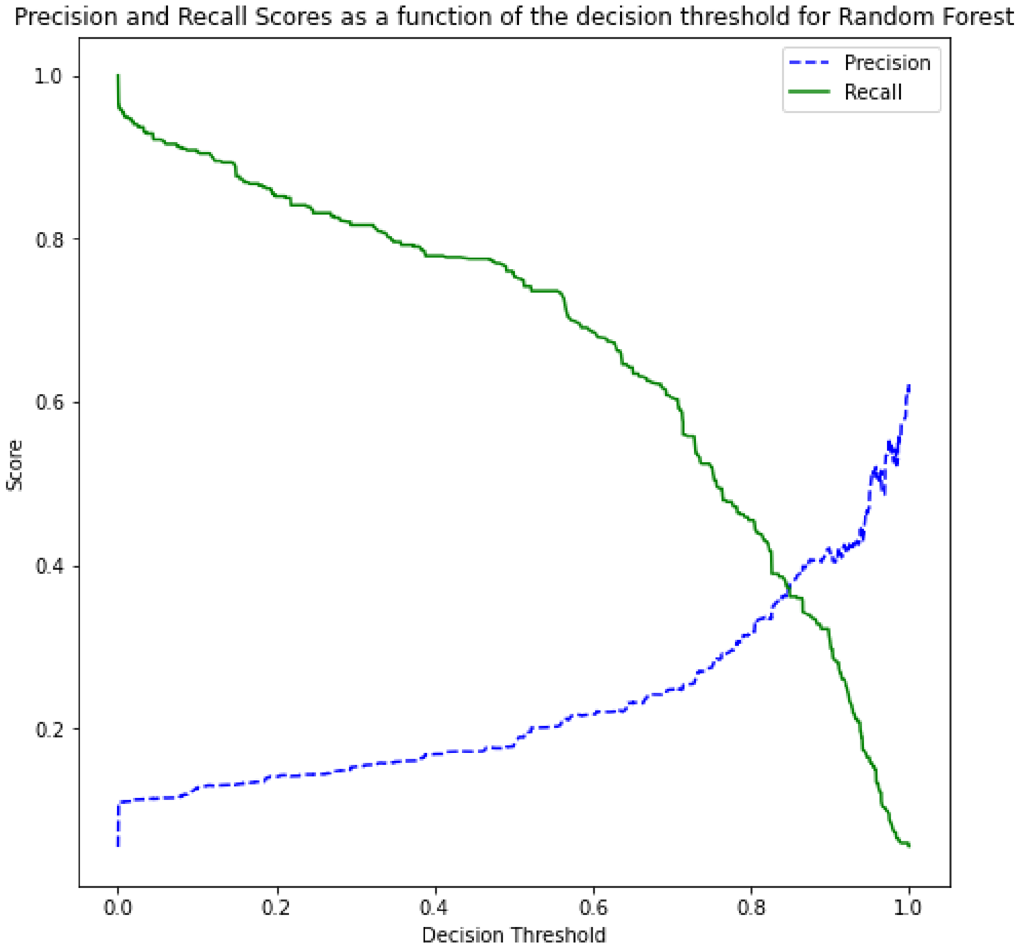
Figure 2. Kaplan-Meier estimated survival probability function.

Limits and future research

- Choice of the algorithm: the objective was double, forcing the choice of the algorithms
- Data availability: history rebuilt only from 2020
- Target inbalance: SMOTE not ideal
- Data integration: feature extracted from text with NLP might improve performance

Thank you for your attention

APPENDIX



Precision and Recall Scores as a function of the decision threshold for Random Forest
Precision
Recall
1.0
0.8
0.6
0.4
0.2
Score
0.0
0.2
0.4
0.6
0.8
1.0
Decision Threshold