STUDENT PERFORMANCE PREDICTION

Predicting Secondary School Student Performance Using Machine Learning

Lucas Lorenzo Jakin

Mentor: Assoc. Prof. Branko Kavšek, PhD

University of Primorska | August, 2024

OVERVIEW

- Objectives
- Problem
- Data Understanding
- Data Preparation
- Methodology
- Implementation
- Evaluation Methods
- Results
- Discussion
- Conclusion & Future work

OBJECTIVES

- Following the **CRISP-DM** methodology
  - Essential process;
  - Keeping a structured manner;
- Data analysis and preparation
  - Gain insights from data;
- Predict students' final grades
  - **Classification** & **Regression**;
  - Build and evaluate predictive models;

PROBLEM

- **Negative** achievement of Portuguese students
  - High student failure and dropping rates;
- Core subject of *Mathematics*
  - Fundamental knowledge for success;
- Predicting *student performance using* **Data Mining**
  - **Is it possible to predict student performance?**
  - **What are the factors that affect student achievement?**

DATA UNDERSTANDING

- Sources of data:
  - **Kaggle** and **UCI-ML:** consistent *structure* and *information;*
  - Mathematics performance;
- 33 attributes and 395 examples:
  - Attributes: ***demographic***, ***social*** and ***school related;***
- Target attribute: **G3**
  - Representing the **final grade;**
  - *Regression task:* 20-point grading scale;
  - *Discretized G3:* five classes of grades (A to F);

ATTRIBUTES

Variable type: character

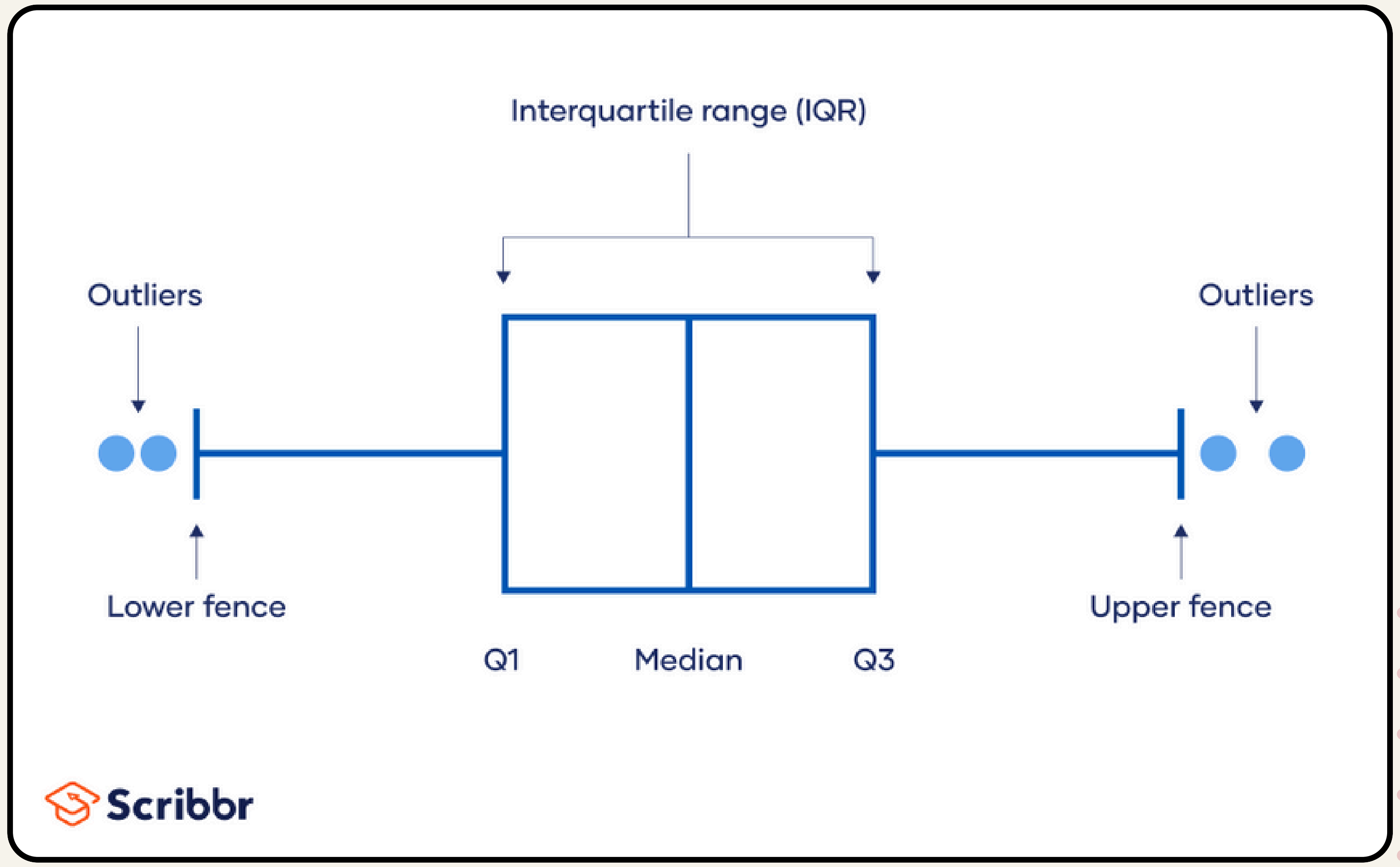
skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
school	0	1	2	2	0	2	0
sex	0	1	1	1	0	2	0
address	0	1	1	1	0	2	0
famsize	0	1	3	3	0	2	0
Pstatus	0	1	1	1	0	2	0
Mjob	0	1	5	8	0	5	0
Fjob	0	1	5	8	0	5	0
reason	0	1	4	10	0	4	0
guardian	0	1	5	6	0	3	0
schoolsup	0	1	2	3	0	2	0
famsup	0	1	2	3	0	2	0
paid	0	1	2	3	0	2	0
activities	0	1	2	3	0	2	0
nursery	0	1	2	3	0	2	0
higher	0	1	2	3	0	2	0
internet	0	1	2	3	0	2	0
romantic	0	1	2	3	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
age	0	1	16.70	1.28	15	16	17	18	22	
Medu	0	1	2.75	1.09	0	2	3	4	4	
Fedu	0	1	2.52	1.09	0	2	2	3	4	
traveltime	0	1	1.45	0.70	1	1	1	2	4	
studytime	0	1	2.04	0.84	1	1	2	2	4	
failures	0	1	0.33	0.74	0	0	0	0	3	
famrel	0	1	3.94	0.90	1	4	4	5	5	
freetime	0	1	3.24	1.00	1	3	3	4	5	
goout	0	1	3.11	1.11	1	2	3	4	5	
Dalc	0	1	1.48	0.89	1	1	1	2	5	
Walc	0	1	2.29	1.29	1	1	2	3	5	
health	0	1	3.55	1.39	1	3	4	5	5	
absences	0	1	5.71	8.00	0	0	4	8	75	
G1	0	1	10.91	3.32	3	8	11	13	19	
G2	0	1	10.71	3.76	0	9	11	13	19	
G3	0	1	10.42	4.58	0	8	11	14	20	

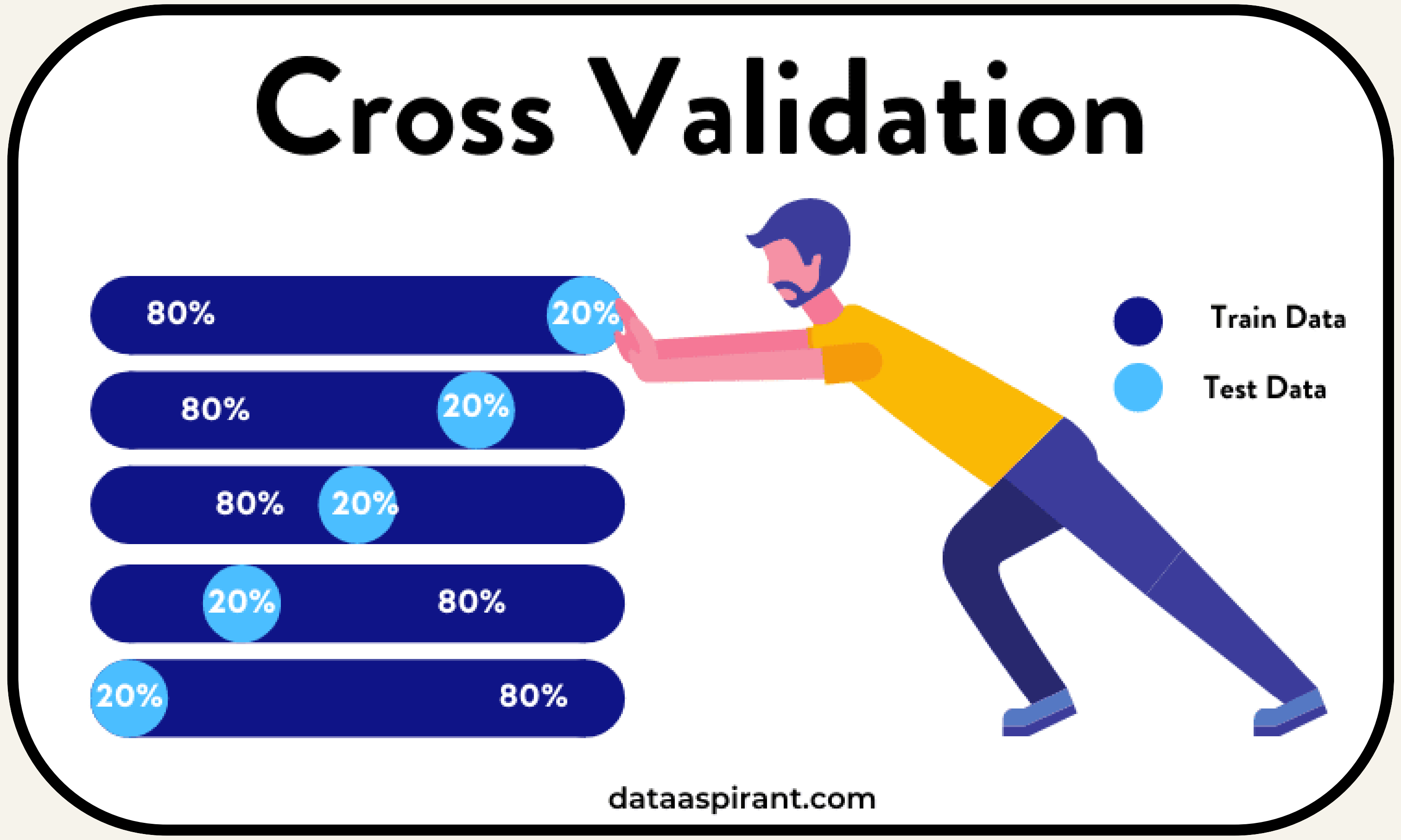
PREPARING THE DATA

- High correlation between G1, G2, and G3
    - **G1** and **G2** removed;
- Regression task:
    - Predicting **G3**;
- Classification task:
    - Predicting **Category** - discretized G3;
- Absence of **missing values**
    - No need for imputation;
- Removing outliers: ***InterQuantile Range method***



PREPARING THE DATA (2)

- <u>Label encoding:</u>
  - Handling **categorical values**;
  - Converting into **numerical**;
- Dataset split into two parts:
  - 75% of data for **training**;
  - 25% of data for **testing**;
- <u>Cross-Validation:</u>
  - **Less biased** than a simple train/test split;
  - **Shuffled 10**-Fold Cross-Validation;



METHODOLOGY

- Application of machine learning algorithms (3 groups):
- ***Baseline algorithms:***

ZeroR

- Majority class classifier
- Benchmark method

OneR

- Best attribute classifier
- Rule with smallest total error

METHODOLOGY (2)

- ***Classification algorithms:***

Random Forest

- Random Forest Classifier
- Supervised learning
- Multiple Decision Trees

k-NN

- k-Nearest Neighbor Classifier
- Grouping data points
- Majority vote on neighbors

SVM

- Support Vector Machine
- Maximum Marginal Hyperplane
- Support Vectors

METHODOLOGY (3)

- ***Regression algorithms:***

Decision Tree

- Three types of nodes
- Constructing **decision rules**
- Decision making problems

k-NN

- k-Nearest Neighbor Regressor
- Dealing with continuous values
- Averaging the k nearest neighbors

Random Forest

- Random Forest Regressor
- Faster and more robust than others

IMPLEMENTATION

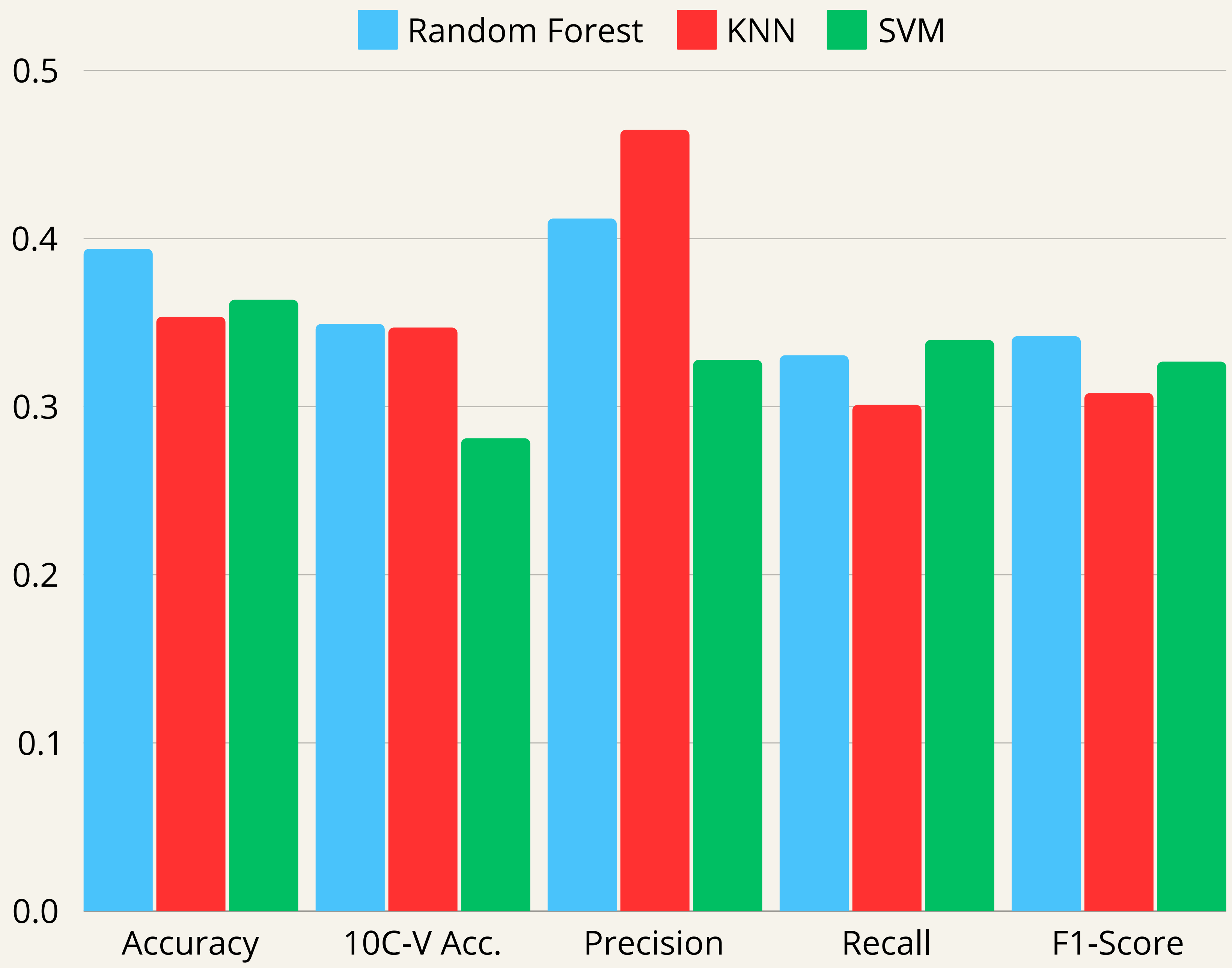
- **ZeroR** implemented “by hand”:
  - Very simple to implement;
- **OneR** implemented in *R programming language:*
  - *Rstudio:* Exploratory Data Analysis;
- All models implemented in **Python**
- **Scikit-learn** library in Python:
  - Open source and commercially usable;
  - Provides unsupervised and supervised learning algorithms;

EVALUATION METHODS

- Models built on the *training set*, tested on the *testing set*
- Classification metrics:
  - *Accuracy;*
  - *Confusion Matrix;*
  - *Precision, Recall and F1-Score;*
- Regression metrics:
  - *MAE;*
  - *MSE* & RMSE;
  - *R-Squared Score;*

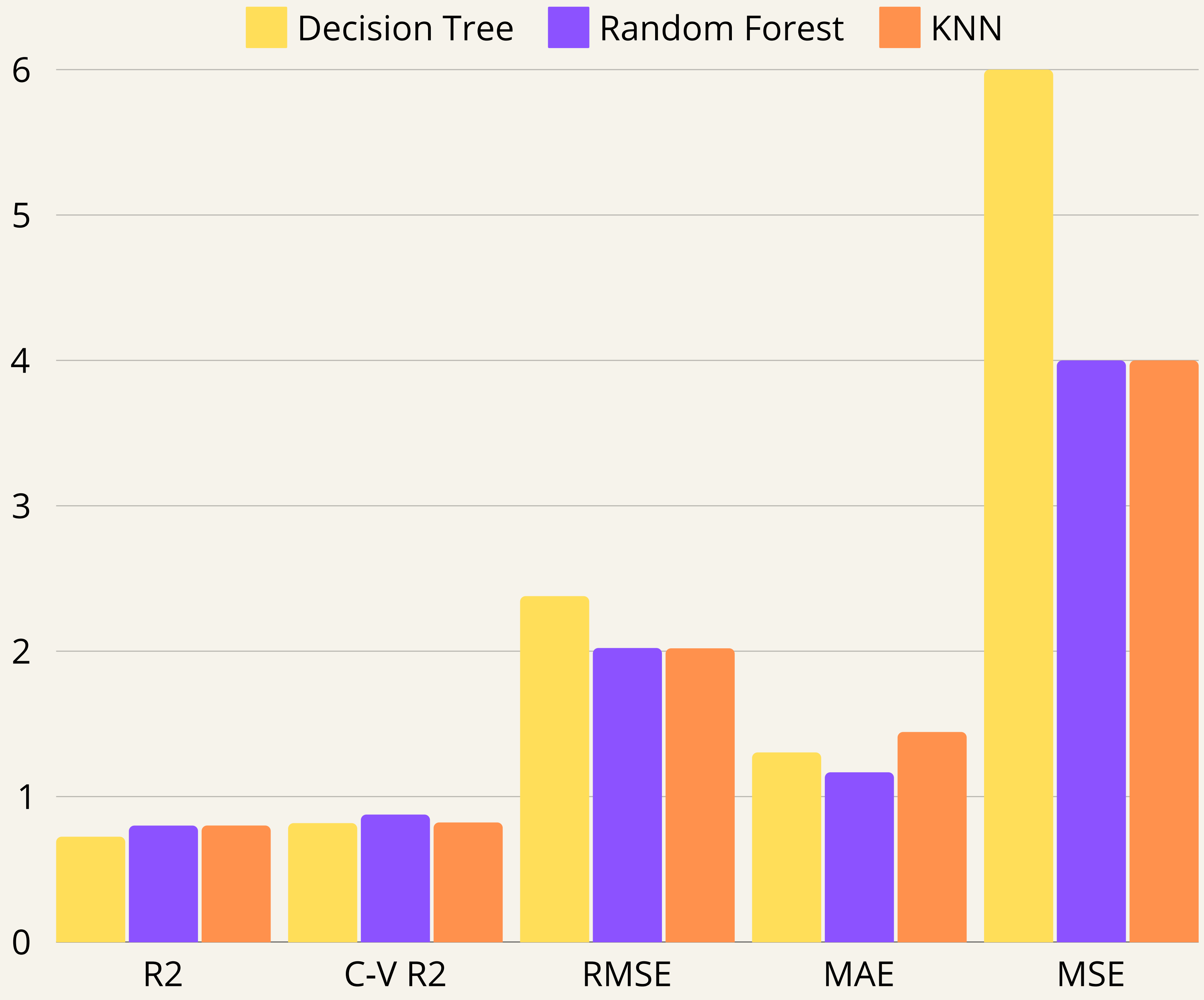
RESULTS

- <u>Classification Scores:</u>
  - **KNN** is most precise but lacks in *recall*;
  - **Random Forest** and **SVM** are more balanced, but **low** in *accuracy;*
  - **SVM** has lowest performance overall;
- Overall struggle with predictions
- Challenging prediction task



RESULTS (2)

- <u>Regression scores:</u>
  - **Random Forest** performs best:
    - Low *RMSE, MAE, MSE;*
  - **KNN:** strong fit and predictive accuracy:
    - Similar to *Random Forest;*
  - **Decision Tree:** slightly worse overall
- Great performance from all models
- Great results overall



DISCUSSION

- Regression models perform much better
- Classification models struggling to make right predictions
- Initial problem more suitable than predicting a **discretized class**

CONCLUSION

- **Data Mining** allows a high level extraction of knowledge from data:
  - Great possibilities in the **education domain**;
  - Enhance school resource management
- Two different **DM goals**
- Six different **DM methods**
- Student achievement highly affected by *previous performances*
- Strong foundation for future exploration

FUTURE WORK

- Model testing on **diverse datasets**
- Tuning model settings and hyperparameters
- Refine techniques to improve model accuracy
- Further study of predictive modeling

University of Primorska | 2024

THANK YOU

Presented By : Lucas Lorenzo Jakin