Introduction to data science checkpoint

Grupo S
Adriano Machado - 202105352
David Pinto - 201704300
Francisco da Ana - 202108762

Project's Motivation

Our project focuses on detecting potential customers without health insurance through predictive analysis of key factors. This enables us to develop targeted marketing campaigns and offer these individuals insurance plans tailored to their needs.

Work Plan

DATASET SIZE

This dataset contains approximately 72,000 instances and 15 features. Two of these features are unique identifiers and will not be used to train the model. The target variable is binary: **health_ins**

Feature	Data Type
unnamed	Numeric
custid	Text
age	Numeric
sex	Categorical
income	Numeric
health_ins	Boolean
num_vehicles	Numeric
marital_status	Categorical
housing_type	Categorical
is_employed	Boolean
state_of_res	Text
code_column	Numeric
gas_usage	Numeric
rooms	Numeric
recent_move	Boolean

MISSING VALUES AND DUPLICATED ROWS

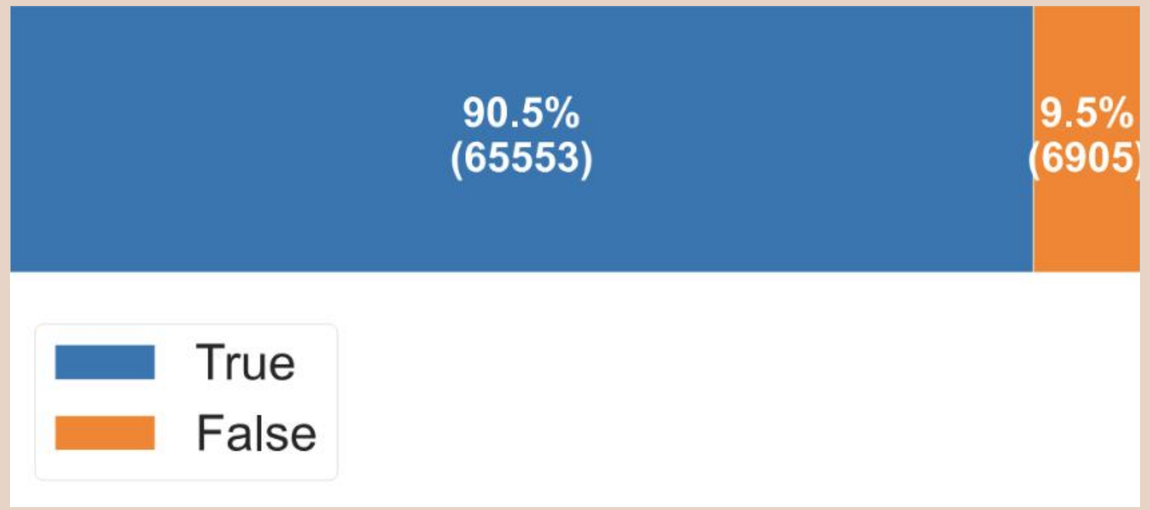
There are **no duplicated** rows in the dataset.
Approximately **2%** of the data contains missing values, all of which occur within the same rows.
Therefore, we have determined that these rows can be **removed** without significant data loss

```
✓ # Missing values by column ···

Unnamed: 0           0
custid               0
sex                  0
income               0
marital_status       0
health_ins           0
housing_type      1686
num_vehicles      1686
age                  0
state_of_res         0
code_column          0
gas_usage         1686
rooms                0
recent_move_b     1687
dtype: int64
```

```
✓ # Number of rows with missing values ···

1687 rows have missing values.
Approx. 2.33% of the orignal dataset.
```

IMBALANCE AND OUTLIERS

The target feature is considerably imbalanced: around **90% of the cases are positive**. balancing techniques may be necessary
Some features like *age*, *income* and *gas_usage* contain extreme values that can affect the model's performance: **outliers** must be handled



Exploratory data analysis methods

- **Data profiling**: Process of examining, summarizing, and analyzing datasets to uncover key data characteristic and patterns often as a preliminary step to data cleaning and preparation

Useful for →

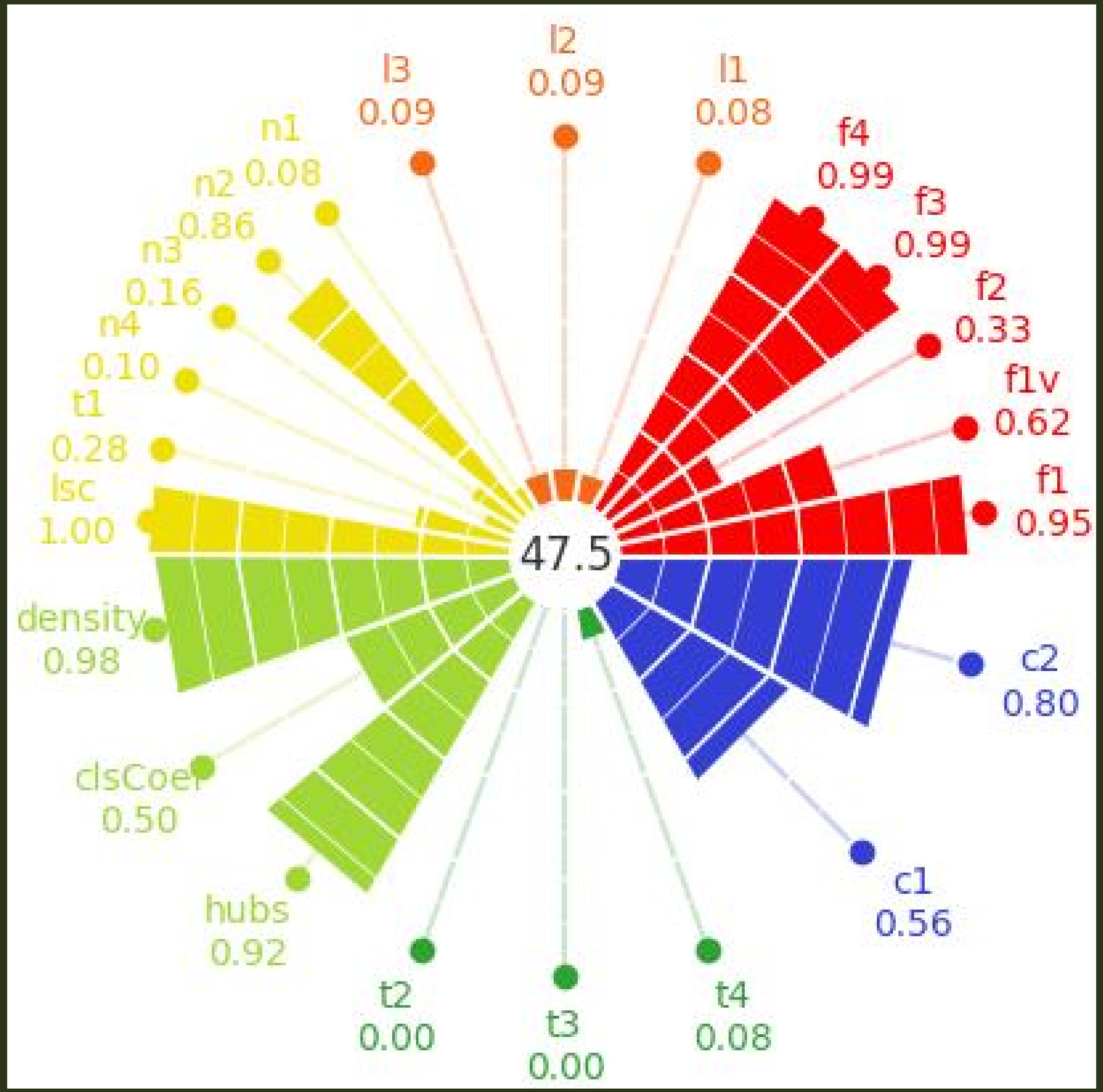
Identifying **missing data**
Detecting **outliers**
Class imbalance detection
Correlations and features **relationships**
Find **irrelevant features**

- **Data complexity analysis**: Data complexity quantifies the characteristics of a dataset that impact the performance and effectiveness of machine learning algorithms, helping to identify potential challenges and guide model selection and optimization

Useful for →

Class separability assessment
Identifying **feature overlap**
Dimensionality reduction potential
Linear separability check
Impact of **class imbalance**
Model selection guidance

Data Complexity Analysis



*Data complexity metrics are presented in the paper: Lorena, A. C., Garcia, L. P., Lehmann, J., Souto, M. C., & Ho, T. K. (2019). How complex is your classification problem? A survey on measuring clas-sification complexity. ACM Computing Surveys (CSUR), 52(5), 1–34. ACM.

We've used the Python library **Problexity** to analyze the complexity of our data. This library provides a easy way to calculate a set of data complexity measures*.

The complexity metrics revealed:

Feature overlapping measures(red):
- High class separabilty
- Moderate feature overlap
- Strong predictive potential

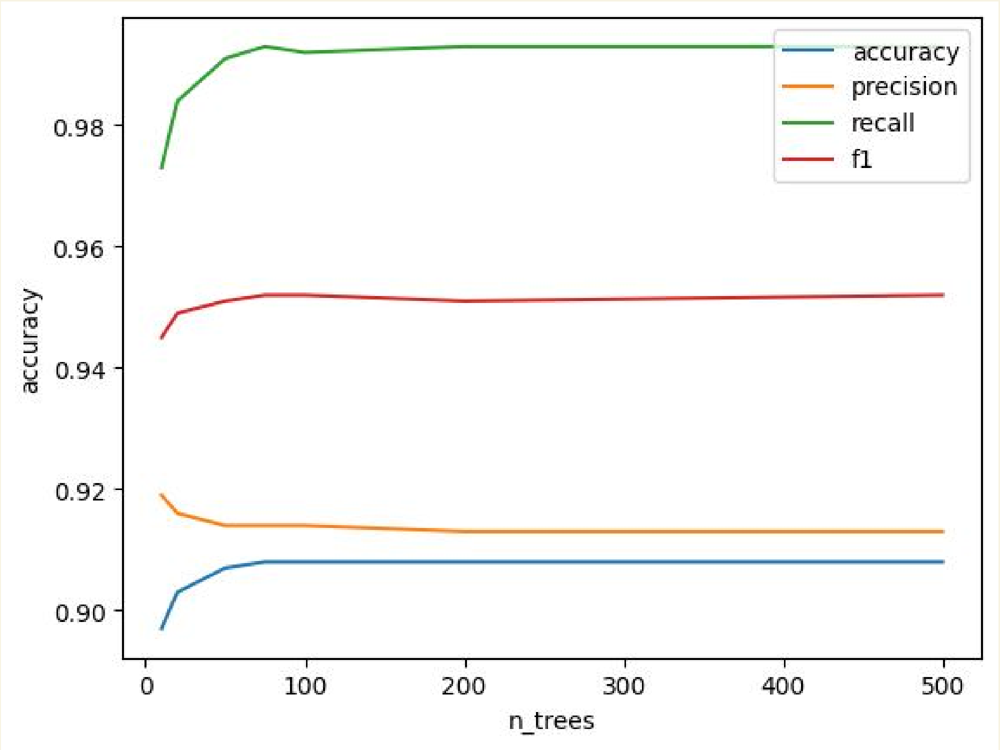
Linearity measures(orange):
- Good linear separability
- Stable decision boundaries

Network measures(green):
- Low dimensionality ratio (PCA revealed that one dimension captures 95% of the data variance

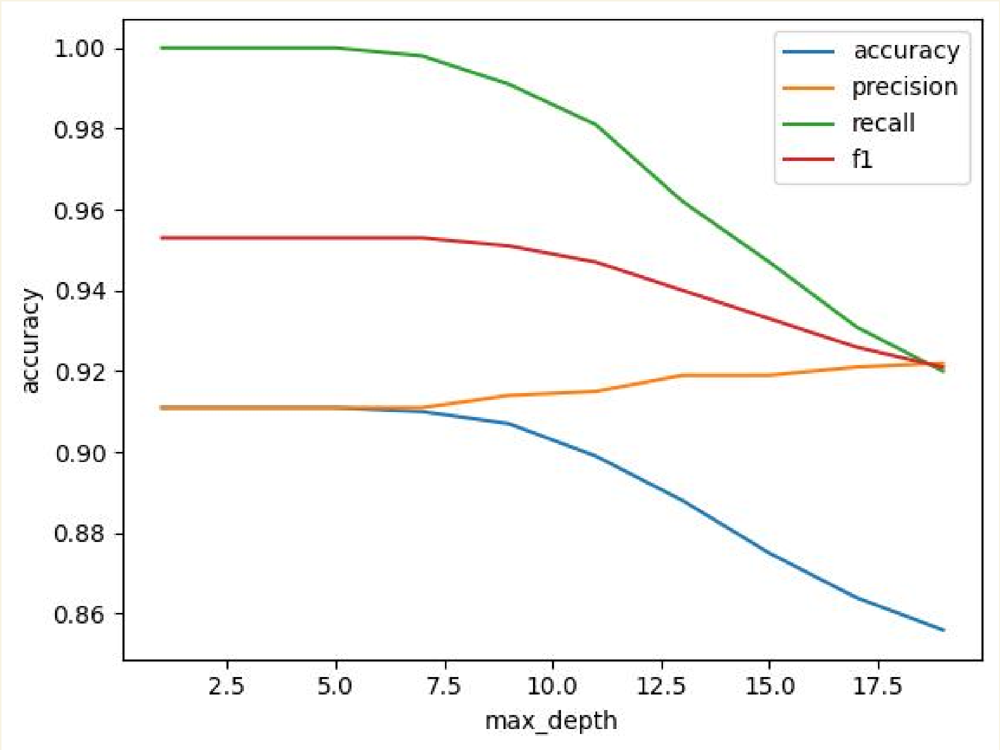
Class balance measures(blue):
- Moderate class imbalance and entropy

Neighborhood measures(yellow):
- Minimal borderline points
- A high ratio of intra/extra class NN distance

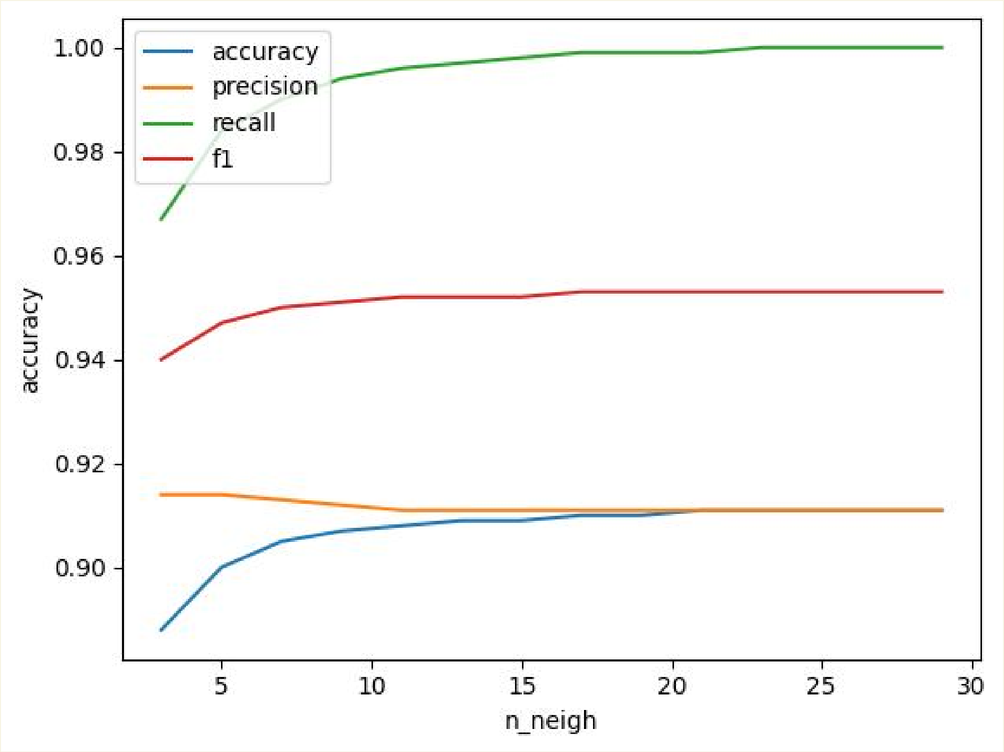
Model Development Methods



Performance Metrics of a **Random Forest** with Varying Numbers of Trees



Performance Metrics of a **Decision tree** with Varying Numbers of depth

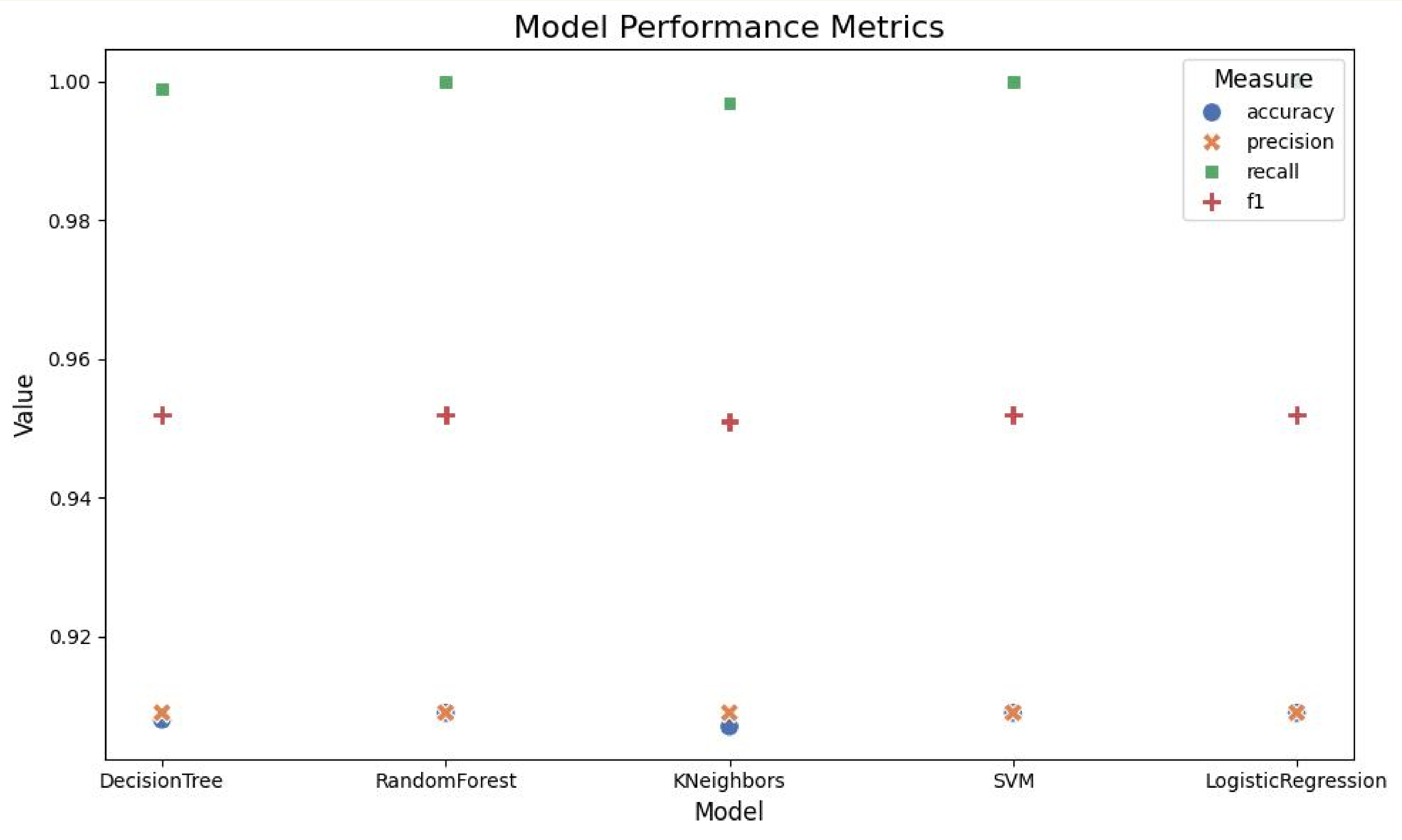


Performance Metrics of a **KNN** with Varying Numbers of Neighbors

Data Profiling and Data Complexity analysis revealed that, besides some evident issues that must be treated (missing values, outliers, …), the dataset exhibits **strong linear characteristics**, **clear class separation**, and **well-clustered instances**.

We have already conducted a quick exploration of some classic ML algorithms from different paradigms to gain an early view of how these models behave when applied to this dataset. We also experimented the impact of certain hyperparameters for a set of models as you can see on the left.

Model Development Methods



model	accuracy	precision	recall	f1	time_s
DecisionTree	0.908	0.909	0.999	0.952	0.108
RandomForest	0.909	0.909	1.000	0.952	1.862
KNeighbors	0.907	0.909	0.997	0.951	3.535
SVM	0.909	0.909	1.000	0.952	66.068
LogisticRegression	0.909	0.909	1.000	0.952	0.348

The results demonstrate that **all models perform similarly** across key metrics. The DecisitionTrees and LogisticRegression, however, are significantly more eficient thus making them the optimal choices for this binary classification task due to their strong balance of performance and efficiency.

Future work

In our upcoming work, we'll focus on two critical areas of improvement:

- **Improve our Data Preprocessing**
  - Compare our current approach of deleting missing values with alternative imputation techniques:
    - Statistical methods (mean/median imputation)
    - Advanced techniques like KNN or regression-based imputation

- **Hyperparameter Optimization**
- **Experiment with ensemble methods**