Final Checkpoint

BREAST CANCER

Francisca Guimarães - up202004229
Inês Oliveira - up202103343
Miguel Santos - up202008450

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Work Specification

This project consists in the application of machine learning models and algorithms related to supervised learning. To train and test the model, we used the data from Kaggle.

For this, we adress the following topics:

- Business Understanding
- Data Understanding
- Data Preparation
- Modeling
- Evaluation

Related Work

To develop our machine learning models, we began by consulting the documentation provided on the Kaggle platform, available at Kaggle. This resource included both the dataset and its detailed description, which were essential for understanding the data we worked with.

For additional insights into the models, we referred to the official scikit-learn documentation, and

For information about outliers and optimize feature selection, we utilized practical guidelines from https://www.geeksforgeeks.org/feature-selection-techniques-in-machine-learning/ and https://www.analyticsvidhya.com/blog/2021/05/feature-engineering-how-to-detect-and-remove-outliers-with-python-code/

In addition to these resources, we also relied on the documentation of several packages used in our project that will be explored in the next section.

Tools and Algorithms

For this project, we used several tools and well-known machine learning libraries:

Pandas: Python library for data manipulation and analysis, particularly suited for working with structured data.
Seaborn: Statistical data visualization library in Python, built on top of Matplotlib, offering high-level interface for creating attractive plots.
Matplotlib: Comprehensive Python library for creating static, animated, and interactive visualizations.
Scikit-learn: Python machine learning library featuring various algorithms for classification, regression, clustering, and more, designed for ease of use.
NumPy: Fundamental package for scientific computing in Python, providing support for multi-dimensional arrays and mathematical functions.

Algorithms:

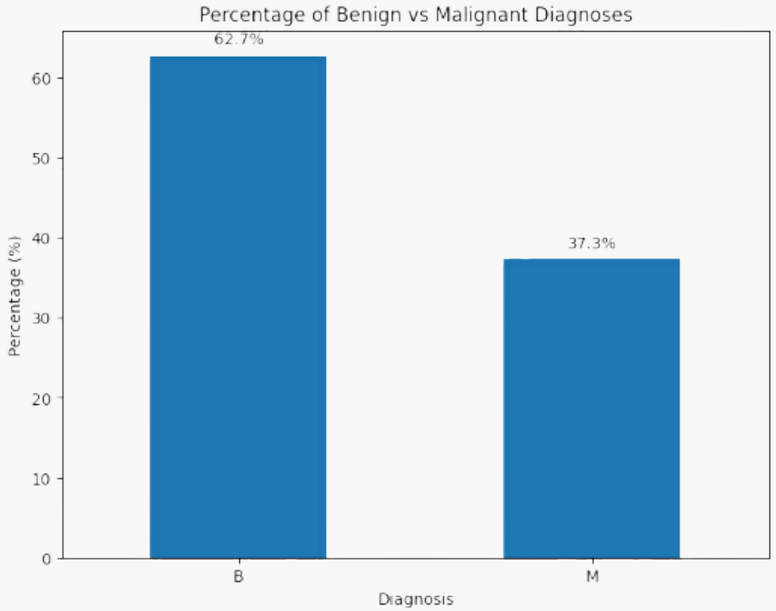
Decision Trees: Tree-based algorithm for classification and regression tasks, splitting data based on features.
Neural Networks: Computational model inspired by the brain's structure, used for various machine learning tasks.
KNN (K-Nearest Neighbors): Simple algorithm for classification and regression, based on the majority vote or mean of nearest neighbors.
EXTRA Support Vector Machines (SVM): Algorithm for classification and regression tasks that finds the optimal hyperplane to separate data points into different classes.
EXTRA Random Forest: Ensemble learning method using multiple decision trees to improve accuracy and control overfitting, suitable for classification and regression tasks.

Business Understanding & Data Understanding

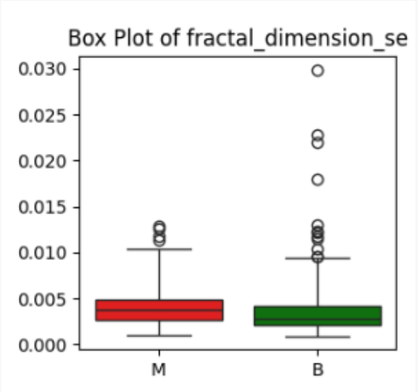
Our goals are to build an accurate model to predict breast cancer and identify important features that help with classification, making the model easier to understand and use in clinical settings.

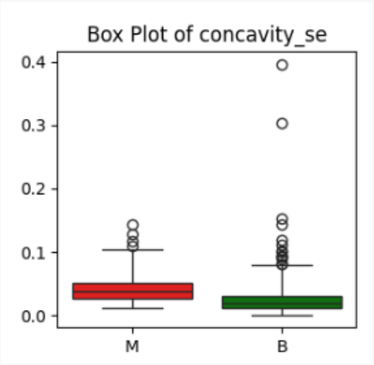
We'll evaluate our algorithms using key metrics: accuracy (overall correctness), precision (correctly identifying malignant cases), recall (capturing all actual malignant cases), and the F1 score (a balance of precision and recall). We'll also use a confusion matrix to provide a detailed breakdown of our model's performance.

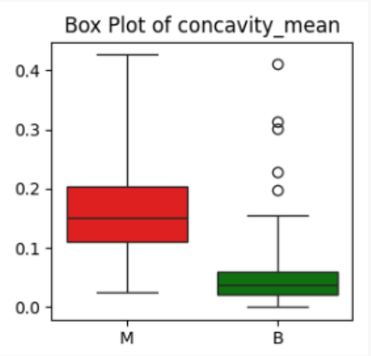
Our dataset contains 569 entries, with 62.7% labeled as Benign (B) and 37.3% as Malignant (M). It includes 32 attributes. We need to balance the data later.



The dataset is clean, with no NULL, NaN, missing, or duplicated values. However, box plots revealed potential outliers that we also need to address.







Data Preparation

Removing Outliers

- Impacts the performance of the models
- How can we be sure it's an outlier and not some rare clinical case?

Implemented two statistical methods: **Z-Score** and the Interquartile Range (**IQR**). Techniques were compared to ensure the most effective identification.

13% of the dataset identified as an outlier. Dataset is quite small, removing 13% of it would significantly impact our study. We replaced the outlier values in the dataset with their respective median values.

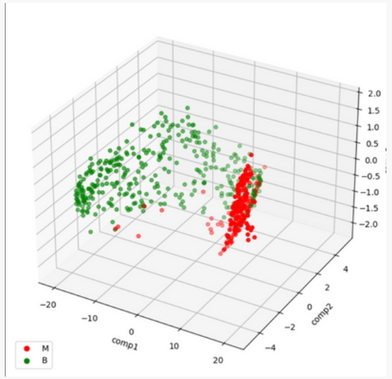


Fig. 1: Representing the data

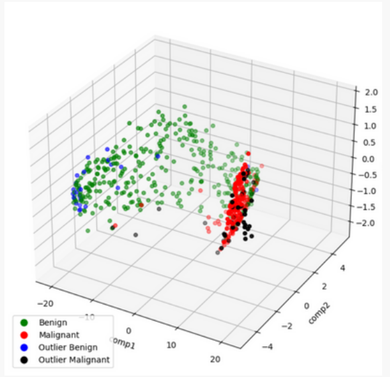


Fig. 2: Representing the outliers (Z-Score)

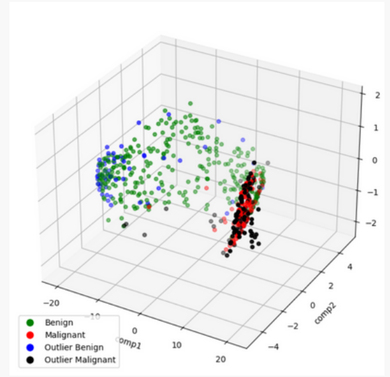


Fig. 3: Representing the outliers (IQR)

Correlation Matrix

- Essential tool in statistical analysis that measures the strength and direction of the linear relationship between pairs of variables. Each cell in the matrix displays the correlation coefficient between two variables.

Several features show a high degree of positive correlation, particularly those related to size (like radius, perimeter, and area), suggesting these may be interdependent. We could manually remove the least significant features but we opted for RFE.

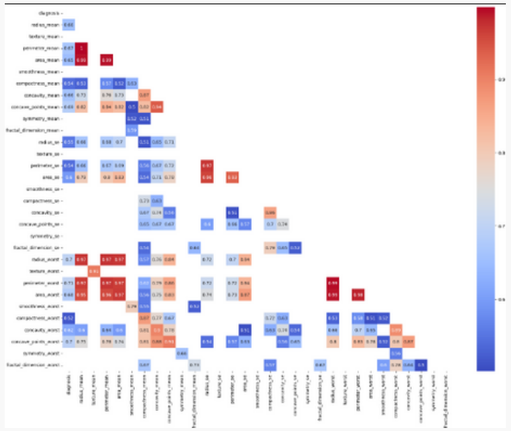


Fig. 4: Correlation Matrix

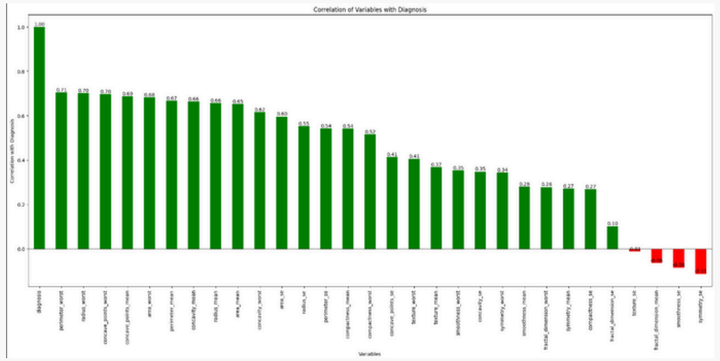


Fig. 5: Feature's relation with diagnosis

Data Preparation

Recursive Feature Elimination

- Feature selection technique to select the most important features

Chosen due to its effectiveness in identifying and removing redundant or less significant features systematically.

This method iteratively constructs a model and removes the weakest feature (or features) until the desired number of features is achieved. This iterative reduction process helps in focusing on the most significant features, improving the model's performance by reducing noise and complexity. We decided to select 10 features.

```
Significant features selected by RFE:
['concave_points_mean', 'radius_se', 'smoothness_se', 'radius_worst', 'texture_worst', 'area_worst', 'smoothness_worst', 'compactness_worst', 'concavity_worst', 'concave_points_worst']
```

Balancing the Data

- Needed because models trained on unbalanced data tend to be biased towards the majority class

```
Original class distribution: Counter({'B': 357, 'M': 212})
Oversampled class distribution: Counter({'M': 357, 'B': 357})
Undersampled class distribution: Counter({'B': 212, 'M': 212})
SMOTE class distribution: Counter({'M': 357, 'B': 357})
```

We explored three different techniques:

- Oversampling: replicates instances of the minority class to balance the class.
- Undersampling: reduces the number of instances from the majority class to match the minority class.
- SMOTE: Generates synthetic samples from the minority class

Oversampling is good for small datasets, but it can cause overfitting. Overfitting happens when a model learns the training data so well, that it performs poorly on new data. On the other hand, undersampling works well for very large datasets because it makes the model run faster, but it means losing some data. Given that our dataset quite small and we want to keep as much information as possible without losing diversity, SMOTE was chosen.

Modeling

Normalization

- Normalization scales all input features to a similar range and distribution.

Normalizing our dataset is important to ensure consistency across features, preventing some from dominating others. Improves the performance of the models.

Several options in sklearn (StandardScaler, MinMaxScaler, RobustScaler).
We use StandardScaler - effective in standardizing features with different scales.

Cross-Validation

- Cross-validation is a technique to evaluate the performance of a model by partitioning the data into subsets.

Ensures that the model's performance is tested on different subsets of data, enhancing its generalizability. Helps in identifying overfitting by ensuring the model performs well on unseen data.

We use Stratified K-Fold - each fold has a similar distribution of the target variable

Parameter Tuning

- Parameter tuning involves adjusting the hyperparameters of a model to optimize its performance.

Helps in finding the best combination of parameters that maximize model accuracy and efficiency. It enhances the model's ability to generalize to unseen data, reducing the risk of overfitting or underfitting and ensures that the model makes the most accurate predictions possible for the given data.

We use Grid Search with Cross-Validation. Chosen for its thoroughness in exploring all possible parameter combinations and its integration with cross-validation to ensure robust model evaluation.

Results and Evaluation

Topics Explored

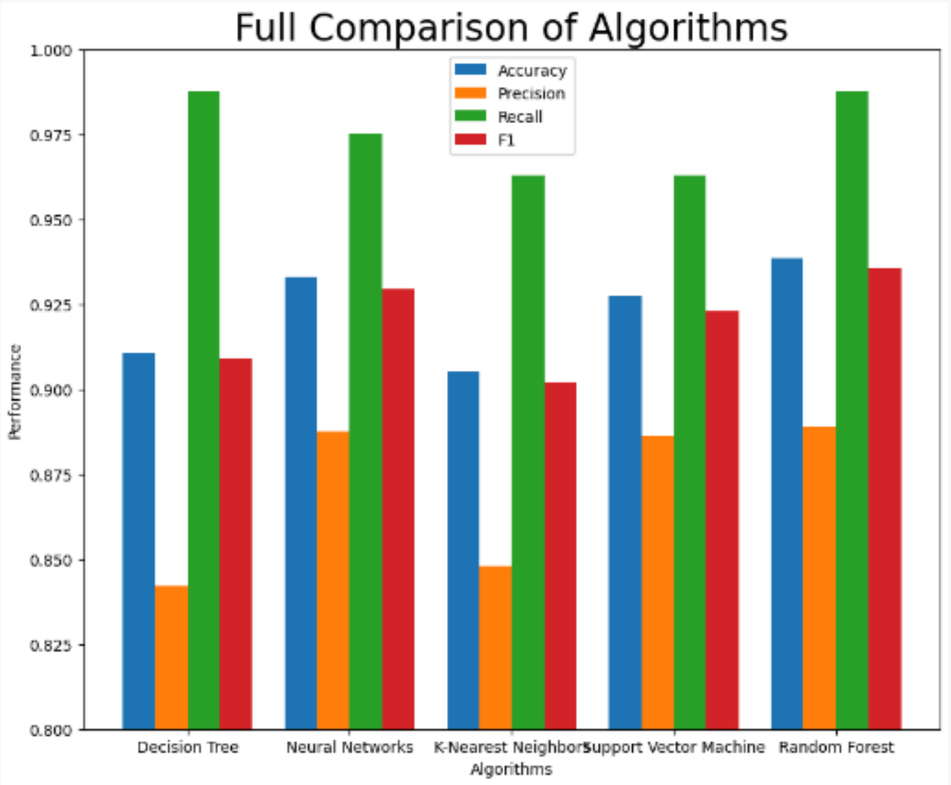
Confusion Matrix, Cleaned vs. Original Dataset, Impact of Normalization, Sampling vs. No Sampling and Different “k” in Cross-Validation

Model Performance Comparison

- Decision Trees: High recall and solid accuracy; lower precision
- Neural Networks: High recall and F1 score;
- K-Nearest Neighbors (KNN): Strong recall and good F1 score; lower accuracy.
- Support Vector Machines (SVM): High accuracy and recall;
- Random Forest: Highest accuracy and recall; strong F1 score and precision; consistently high performance across all metrics.

Optimal Model Choices

- Random Forest and Neural Networks: Best for high accuracy and balanced performance. Random Forest offers robust performance with high recall and precision. Neural Networks provide strong recall and F1 score with high computational cost.
- Decision Trees, KNN, and SVM: Effective based on specific application needs. Decision Trees are quick and reliable, KNN is suitable for high sensitivity to positive cases, and SVM balances high accuracy and recall.



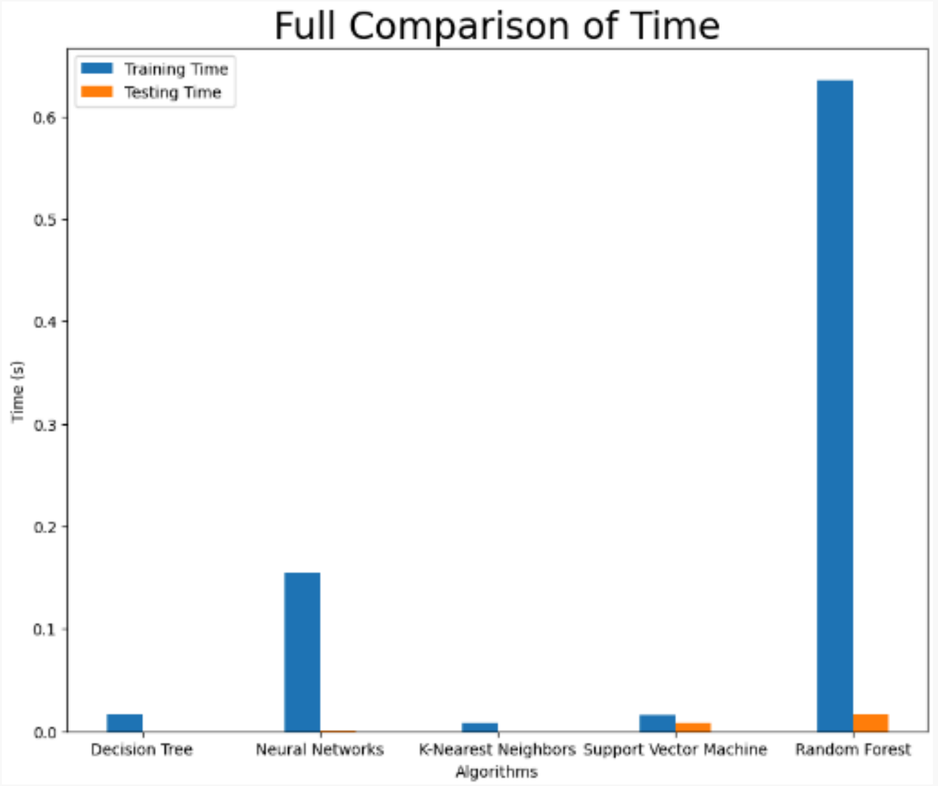
Results and Evaluation

Model Time Comparison

- Decision Trees: Fast training and minimal testing times, perfect for rapid model updates and real-time predictions.
- Neural Networks: Long training times but very low testing times, ideal for infrequent training and frequent predictions.
- K-Nearest Neighbors (KNN): Very short training times but higher testing times, suitable for smaller datasets or scenarios prioritizing training speed.
- Support Vector Machines (SVM): Quick training and minimal testing times, efficient for both training and real-time predictions.
- Random Forest: Longest training times due to multiple decision trees but low testing times, best for periodic training and quick predictions.

Optimal Model Choices

- Decision Trees and SVMs: Excellent for quick training and testing due to computational efficiency.
- Neural Networks: Ideal for applications needing fast predictions after extensive training.
- K-Nearest Neighbors (KNN): Suitable for small datasets with quick training but slower testing.
- Random Forest: Best for high accuracy with periodic training and quick predictions.



Conclusions

This project provided valuable insights into the application of machine learning models for breast cancer mass detection and emphasized the critical role of data normalization, sampling, and preprocessing.

- Optimal Model: Random Forest, chosen for its highest accuracy and recall, ensuring precise and reliable predictions. Crucial in medical diagnostics to minimize false negatives and reduce false positives.
- Alternative Models:
  - Neural Networks: High recall and F1 scores, suitable for infrequent training and frequent predictions.
  - SVM: High accuracy and quick training/testing times, practical for real-time applications.
- Other Models: Decision Trees and K-Nearest Neighbors (KNN) have advantages, but Random Forest, Neural Networks, and SVMs are the top choices for breast cancer mass detection.

It's important to note that our dataset was relatively small, which may limit the generalizability of our findings. In other contexts with larger or different datasets, the performance and suitability of these models might vary.

Despite this, this project significantly enhanced our understanding of machine learning and its applications, providing a solid foundation for future research.