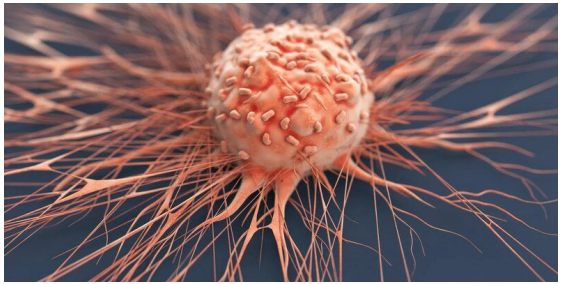
Artificial Intelligence Cancer Data Analysis

João Alves - up202007614@fe.up.pt
José Ribeiro - up202007231@fe.up.pt
Rúben Monteiro - up202006478@fe.up.pt

Problem Description - Cancer Data

Objective: Develop a successful machine learning model that can predict whether or not a cell is benign or malignant.

Dataset: 30 features of 570 different cells, along with the id and the diagnosis of each case. The diagnosis is represented as either B (benign) or M (malignant).



Algorithms and Tools

Python Libraries:

- Pandas
- Matplotlib
- Seaborn
- Imbalanced-learn
- Scikit-learn

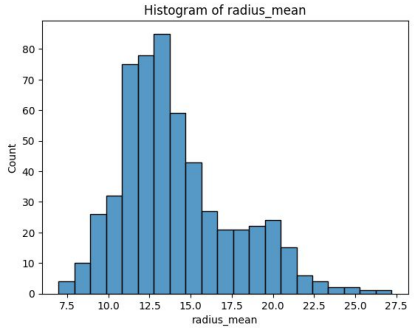
Algorithms used (so far):

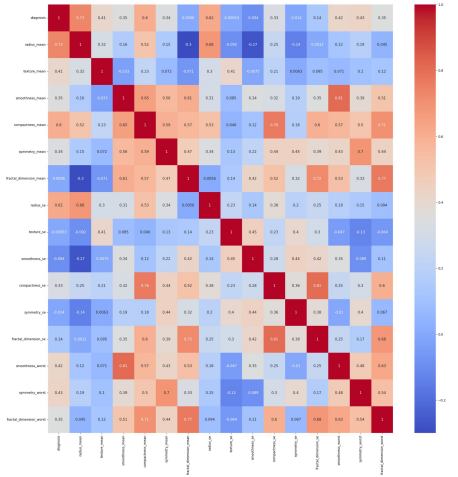
- Decision Tree
- KNN
- SVM
- Neural Network
- Random Forest
- Naive Bayes

Data Pre-processing

Steps:

- Understanding and Visualization of the dataset
- Eliminate unnecessary column id
- Remove outliers from the dataset
- Remove highly correlated features
- Balancing results
    - Oversampling
    - Undersampling



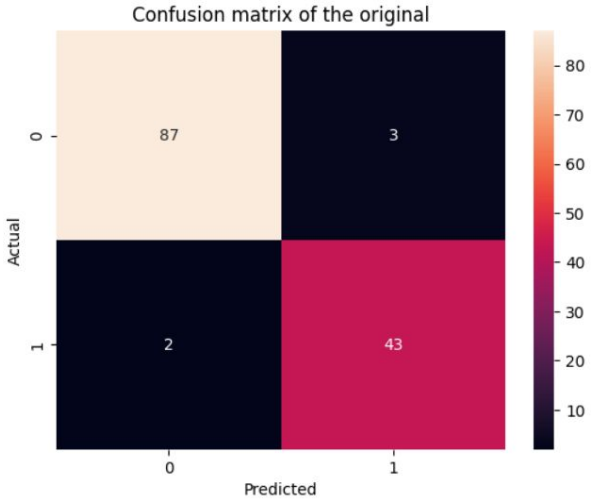


Development of Model

- Use of GridSearchCV to find the best parameters for each of the current algorithms.
- For each algorithm, we get the results for oversampling, undersampling and the original data.
- Analysis of the results in a confusion matrix
- Multiple measures to evaluate the model are used

```
Best accuracy: 0.9554268292682926
Best precision: 0.9573895855924193
Best recall: 0.9554268292682926
Best f1_score: 0.9552898237268609
Best parameters: {'C': 100, 'gamma': 0.01}
```

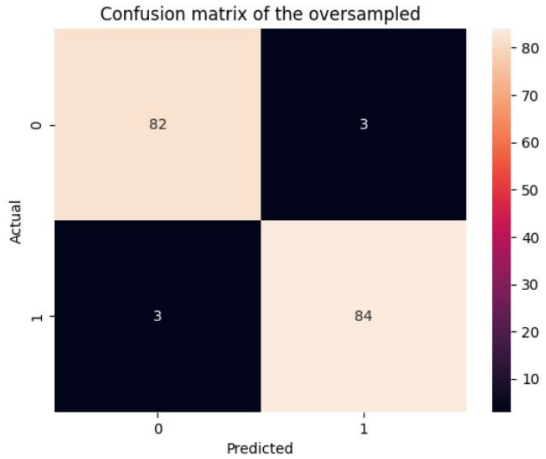
SVM with original dataset



Resultados

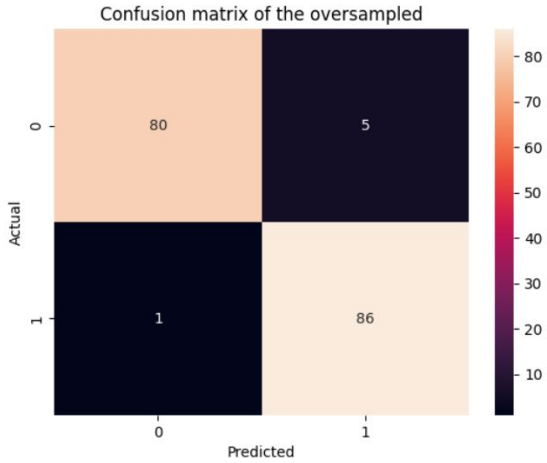
Decision Tree (Oversampling):

```
Best accuracy: 0.9497737556561086
Best precision: 0.9509808430494704
Best recall: 0.9497737556561086
Best f1_score: 0.9497394095196536
Best parameters: {'criterion': 'gini', 'max_depth': 14}
```



Neural Network ():

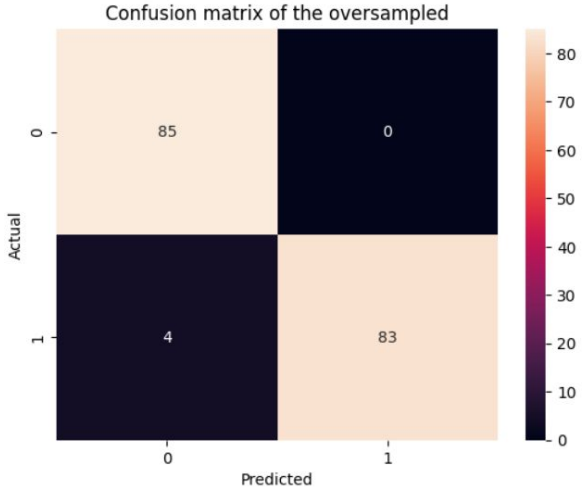
```
Best accuracy: 0.9632730015082955
Best precision: 0.9651144205004922
Best recall: 0.9632730015082955
Best f1_score: 0.9632271832601624
Best parameters: {'activation': 'logistic', 'alpha': 0.001,
'hidden_layer_sizes': (100, 100), 'max_iter': 500}
```



Resultados

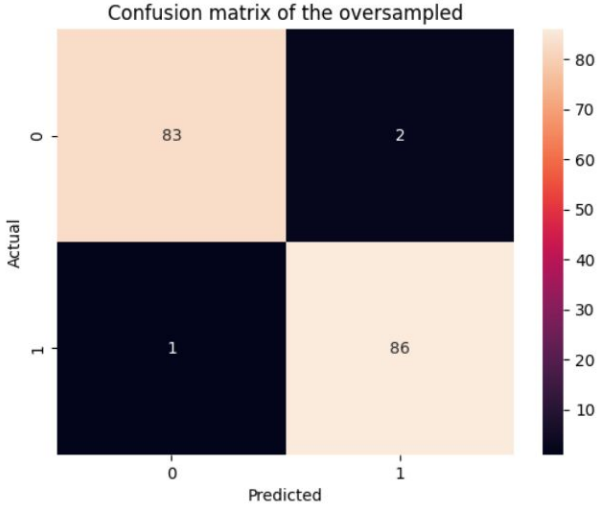
SVM (Oversampling):

```
Best accuracy: 0.9651960784313726
Best precision: 0.9667082473032439
Best recall: 0.9651960784313726
Best f1_score: 0.9651623993808867
Best parameters: {'C': 10, 'gamma': 0.01, 'kernel': 'poly'}
```



KNN (Oversampling):

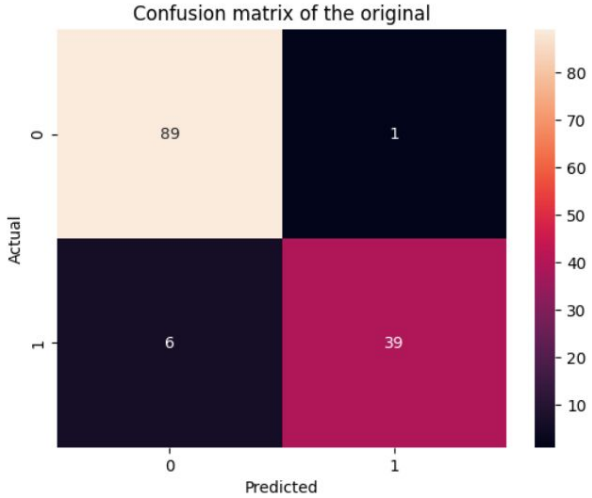
```
Best accuracy: 0.9555429864253394
Best precision: 0.9577050433430845
Best recall: 0.9555429864253394
Best f1_score: 0.955468105748112
Best parameters: {'metric': 'manhattan', 'n_neighbors': 9, 'weights': 'distance'}
```



Resultados

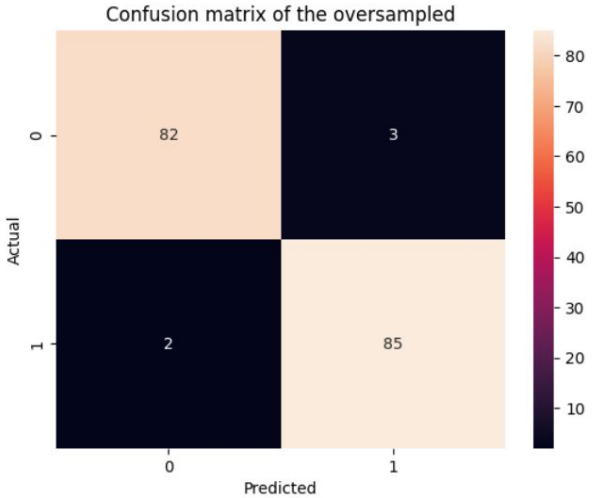
Naive Bayes (Original):

```
Best accuracy: 0.9429268292682927
Best precision: 0.9466820437983993
Best recall: 0.9429268292682927
Best f1_score: 0.9427206631781535
Best parameters: {'var_smoothing': 1e-05}
```



Random Forest (Oversampling):

```
Best accuracy: 0.9845776772247362
Best precision: 0.985416217769159
Best recall: 0.9845776772247362
Best f1_score: 0.9845641468327372
Best parameters: {'criterion': 'gini', 'max_depth': 9, 'n_estimators': 300}
```



Results

- We noticed an increase in performance of our models after some more data pre-processing after our first checkpoint.
- Most algorithms performed better with oversampled data.
- The best results we obtained were from the Random Forest, SVM and Neural Network. However, all algorithms performed relatively well.
- Overall it was a successful project, where we were able to achieve our goals

References

Scikit-learn documentation: https://scikit-learn.org/stable/index.html

Theoretical Class Slides