U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Supervised Learning

Cancer Data

T05 - G01
Artificial Intelligence

Diogo Costa, 202007770
Fábio Sá, 202007658
João Araújo, 202004293

Specification

This project will apply supervised learning techniques to a real-world problem of malignant cancer detection, developing and comparing the performance of different supervised learning algorithms in classifying cancer cases using a given dataset.

The dataset used in this project was obtained from the Kaggle's Cancer Data Page. It contains 569 instances of cancer biopsies, each with 32 attributes. One of the attributes is the diagnosis, which can be either benign or malignant. The other 30 attributes are numeric-valued laboratory measurements, such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension. The remaining attribute represents the ID number of the instance.

Related Work

Machine learning applications in cancer prognosis and prediction

This paper presents a study on the application of machine learning algorithms for cancer detection. The study compared the performance of various machine learning models, including Artificial Neural Networks, Support Vector Machines (SVM), and Decision Trees, using clinical data from patients.

Supervised Learning-based Cancer Detection

Although the data manipulated by this paper are obtained through images of cancer cells, they subsequently use a layer of machine learning (supervised learning) to classify the types of cancer. The evaluation of the models is similar to the one that will be used in this project.

Tools

Python Libraries

- **Jupyter Notebook** - used to structure all project progress such as source code, algorithms and model evaluations
- **Seaborn** - provides a high-level interface for creating informative and attractive statistical graphics
- **Matplotlib** - to create a wide variety of plots and graphs, including line plots and histograms
- **Scikit-Learn** - provides a range of tools for data preprocessing, modeling, and evaluation
- **Numpy** - for mathematical data modeling
- **Pandas** - data manipulation and analysis tools for working with structured data, such as tables
- **PyCaret** - simple library that allows to run multiple machine learning models at once, and compare themselves

Algorithms

We trained four models that were studied in class:

- **K-Nearest Neighbors**: finds the K nearest data point, but may be slow on this project (large dataset and high-dimensional data)
- **Decision Trees:** prone to overfitting and may not generalize well to new data
- **Support Vector Machine:** a robust and effective classification algorithm and may be computationally expensive on this dataset
- **Neural Networks:** capable of handling complex non-linear relationships and large datasets, but may overfit

In addition to them, we tested others present in the PyCaret.Classification module:

CatBoost Classifier
Extreme Gradient Boosting
Light Gradient Boosting Machine
Random Forest Classifier

Logistic Regression
Extra Trees Classifier
Gradient Boosting Classifier
Ridge Classifier

Linear Discriminant Analysis
Ada Boost Classifier
Quadratic Discriminant Analysis
Dummy Classifier

Data Pre-Processing

Filtering out outliers

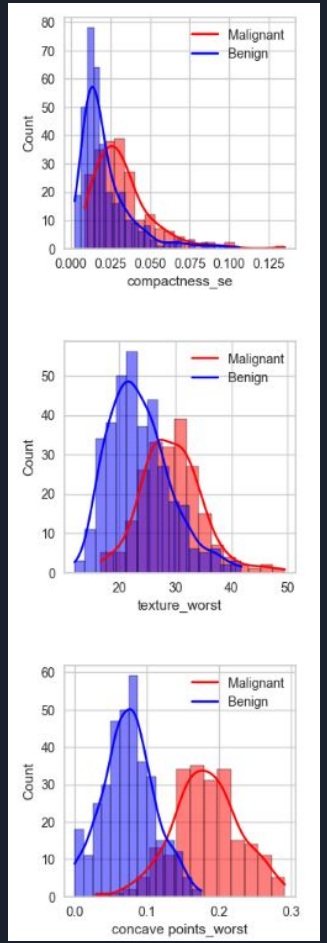
The column **id** is not useful for our analysis, so we can drop them. We need to also drop out the column **32**, since it is full of null values. All other values were uniform and with the same order of magnitude.

Encode the target variable

We need to encode the target variable, so that we can use it in our models, since it is currently object type. The remaining columns are already numeric, so we don't need to encode them.

Feature distribution analysis

Since we have a lot of features, we can't plot all of them, so we will use the Pearson correlation coefficient to find the correlation between the features. we will plot the correlation of the features with the diagnosis. An example can be seen in the figure on the side.



Data Pre-Processing

Resource extraction

We then eliminated some features that had a correlation greater than 95% with each other, to reduce the number of features while maintaining (mostly) the same amount of relevant information for the model.
The figure on the side indicates the 28 features chosen after this step and their correlation with the diagnosis.

Split the dataset into training and testing sets

As the dataset is quite unbalanced (357 benign and 212 malignant) and we have few cases to train the models, we oversampled to ensure that the train set has the same proportion of benign and malignant instances.

Developed Models

From the list of available models, these were the best performers (with default settings applied):

- CatBoost Classifier	- 97.57%	accuracy
- Extreme Gradient Boosting	- 97.14%	accuracy
- Light Gradient Boosting Machine	- 97.13%	accuracy
- Random Forest Classifier	- 96.91%	accuracy
- Logistic Regression	- 96.70%	accuracy

Results comparison (training)

Without feature extraction

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr	Logistic Regression	0.9670	0.9948	0.9412	0.9709	0.9549	0.9289	0.9302	0.537
xgboost	Extreme Gradient Boosting	0.9669	0.9931	0.9412	0.9709	0.9547	0.9286	0.9302	0.136
catboost	CatBoost Classifier	0.9646	0.9931	0.9353	0.9699	0.9519	0.9239	0.9248	2.226
et	Extra Trees Classifier	0.9625	0.9909	0.9294	0.9709	0.9488	0.9192	0.9208	0.164
knn	K Neighbors Classifier	0.9603	0.9885	0.9235	0.9705	0.9452	0.9142	0.9163	0.196
ridge	Ridge Classifier	0.9603	0.0000	0.9059	0.9878	0.9433	0.9130	0.9169	0.083
rf	Random Forest Classifier	0.9602	0.9884	0.9353	0.9583	0.9461	0.9146	0.9154	0.159
ada	Ada Boost Classifier	0.9602	0.9866	0.9353	0.9591	0.9467	0.9149	0.9155	0.139
gbc	Gradient Boosting Classifier	0.9602	0.9909	0.9412	0.9583	0.9479	0.9158	0.9180	0.206
lightgbm	Light Gradient Boosting Machine	0.9602	0.9888	0.9294	0.9657	0.9459	0.9145	0.9163	0.160
svm	SVM - Linear Kernel	0.9581	0.0000	0.9235	0.9643	0.9426	0.9097	0.9112	0.084
lda	Linear Discriminant Analysis	0.9537	0.9884	0.9000	0.9745	0.9346	0.8989	0.9018	0.101
qda	Quadratic Discriminant Analysis	0.9427	0.9798	0.9118	0.9367	0.9223	0.8770	0.8792	0.096
nb	Naive Bayes	0.9406	0.9828	0.9059	0.9356	0.9195	0.8725	0.8739	0.088
dt	Decision Tree Classifier	0.9117	0.9117	0.9118	0.8674	0.8849	0.8136	0.8186	0.089

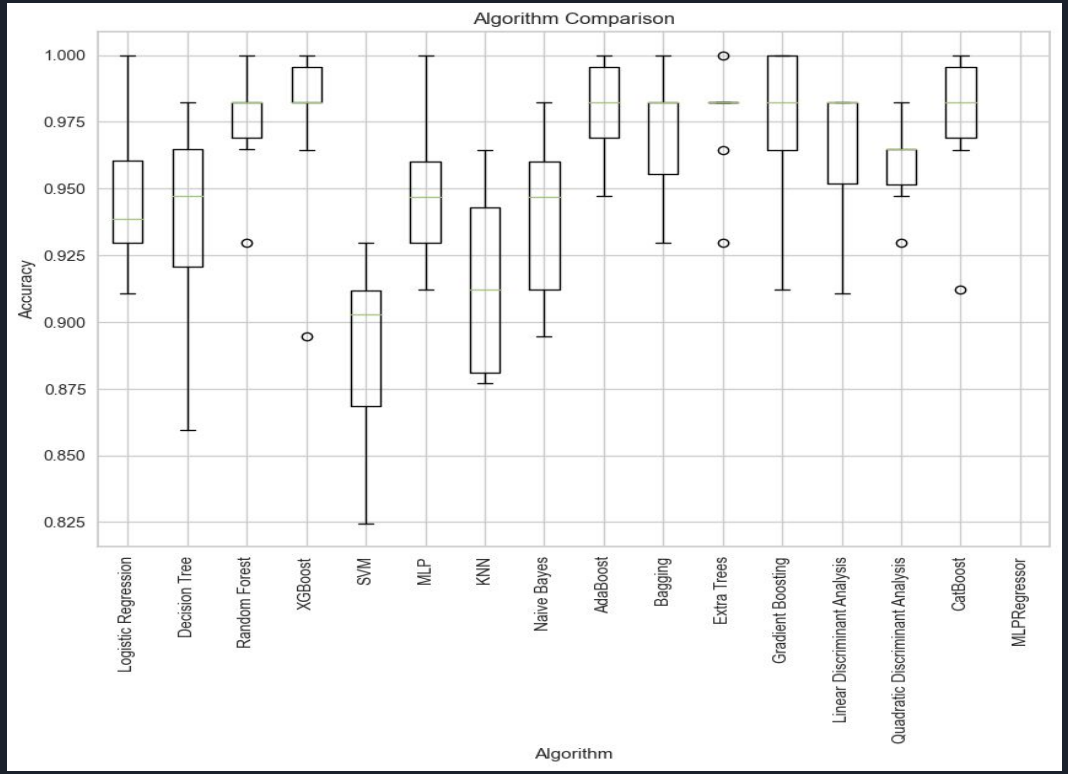
With feature extraction

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
CatBoost Classifier	0.9757	0.9931	0.9529	0.9830	0.9670	0.9478	0.9489	0.196
Extreme Gradient Boosting	0.9714	0.9931	0.9471	0.9761	0.9611	0.9385	0.9390	0.131
Light Gradient Boosting Machine	0.9713	0.9902	0.9471	0.9765	0.9611	0.9384	0.9391	0.125
Random Forest Classifier	0.9691	0.9925	0.9471	0.9701	0.9581	0.9336	0.9342	0.137
Logistic Regression	0.9670	0.9925	0.9412	0.9716	0.9549	0.9289	0.9307	0.523
Extra Trees Classifier	0.9670	0.9933	0.9294	0.9808	0.9538	0.9282	0.9297	0.140
Gradient Boosting Classifier	0.9646	0.9914	0.9412	0.9660	0.9525	0.9244	0.9256	0.126
Ridge Classifier	0.9626	0.0000	0.9176	0.9823	0.9471	0.9183	0.9215	0.111
K Neighbors Classifier	0.9625	0.9856	0.9235	0.9757	0.9481	0.9188	0.9206	0.204
Linear Discriminant Analysis	0.9603	0.9878	0.9059	0.9879	0.9437	0.9133	0.9168	0.116
SVM - Linear Kernel	0.9581	0.0000	0.9471	0.9446	0.9448	0.9111	0.9124	0.109
Ada Boost Classifier	0.9581	0.9920	0.9471	0.9436	0.9447	0.9109	0.9118	0.123
Quadratic Discriminant Analysis	0.9493	0.9741	0.9118	0.9514	0.9300	0.8904	0.8921	0.115
Decision Tree Classifier	0.9362	0.9300	0.9059	0.9253	0.9136	0.8630	0.8653	0.119
Naive Bayes	0.9296	0.9780	0.9176	0.8994	0.9063	0.8501	0.8528	0.109

Results comparison

We can compare the performance of multiple machine learning models using a boxplot graph.

By analysing the graph, we can obtain a visual comparison of algorithm performance and highlight models with potentially higher accuracy and consistency.



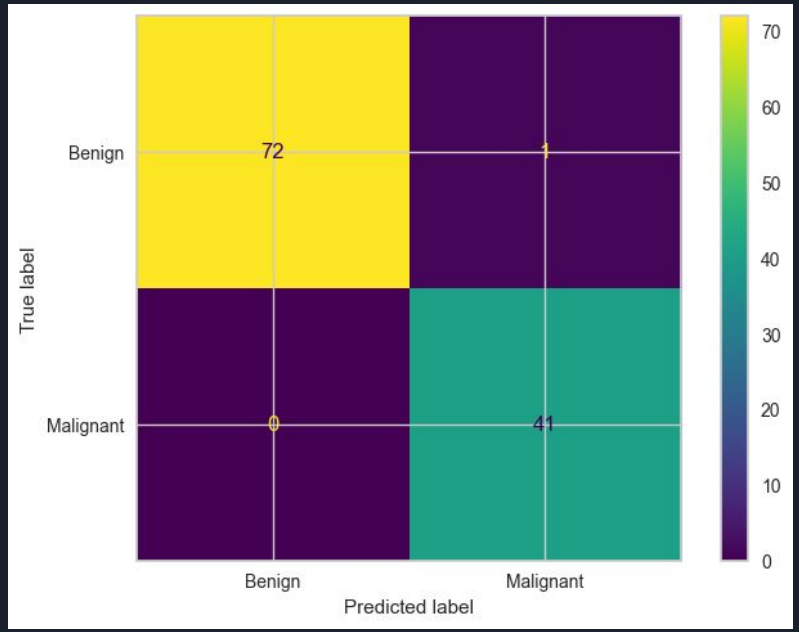
Results comparison

The graph represents predicted and true labels, highlighting correct and incorrect predictions of the CatBoost classifier model

It provides insights into the model's accuracy, biases, and strengths.

A confusion matrix is a valuable tool for assessing a model's ability to classify instances accurately.

The chosen algorithm was mistaken in only 1 case when applied to the test data, where it mistakenly assigned a false positive.



References

- Erdem Tahar et al. Cancer Data. 2023. Accessed on 26 April 2023.
- Konstantina Kourou et al. Machine learning applications in cancer prognosis and prediction. 2015. Accessed on 27 April 2023.
- Juel Sikder et al. Supervised Learning-based Cancer Detection. 2021. Accessed on 27 April 2023.
- Moez Ali et al. PyCaret: An open source, low-code machine learning library in Python. 2020. Accessed on 15 May 2023.
- XGBoost Developers et al. XGBoost: distributed gradient boosting library. 2022. Accessed on 18 May 2023.
- Micaeld et al. Cancer Prediction (99% Acc - 100% Precision). 2022. Accessed on 18 May 2023.

Material Usage

Software:

- Visual Studio Code, Pycharm and Anaconda as the used IDE's;

Languages:

- For the development of this application, we required only the usage of Python (on a Jupyter Notebook);

Python Libraries:

- Seaborn
- Matplotlib
- Scikit-Learn
- Numpy
- Pandas
- CatBoost
- XGBoost
- PyCaret
- Imbalanced-Learning