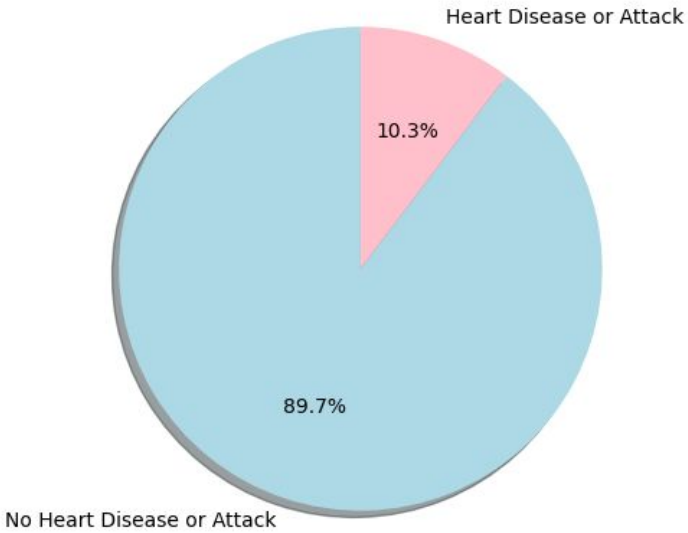
Application of ML techniques in the medical field, prediction of heart diseases

Elisa Ancarani

MSc of Artificial Intelligence, Unibo

The Dataset

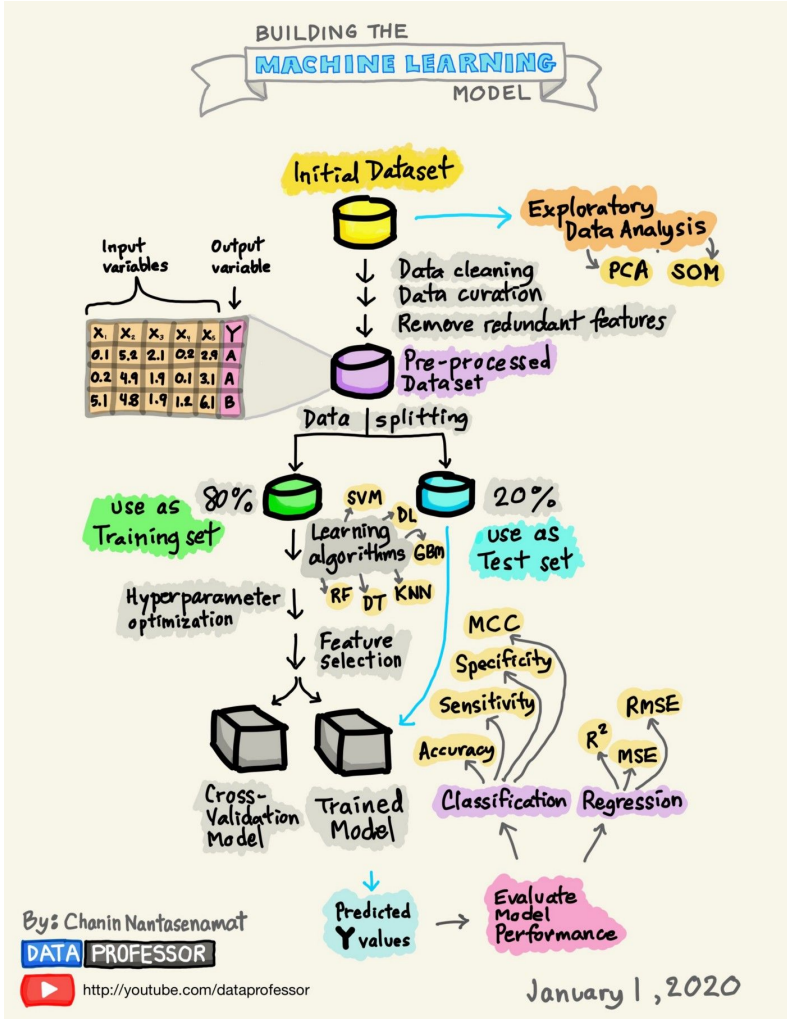
- 22 features, 253680 records
- Target Feature (binary): Heart Disease or Attack
- **Main Challenge**: Heavily Imbalanced!!



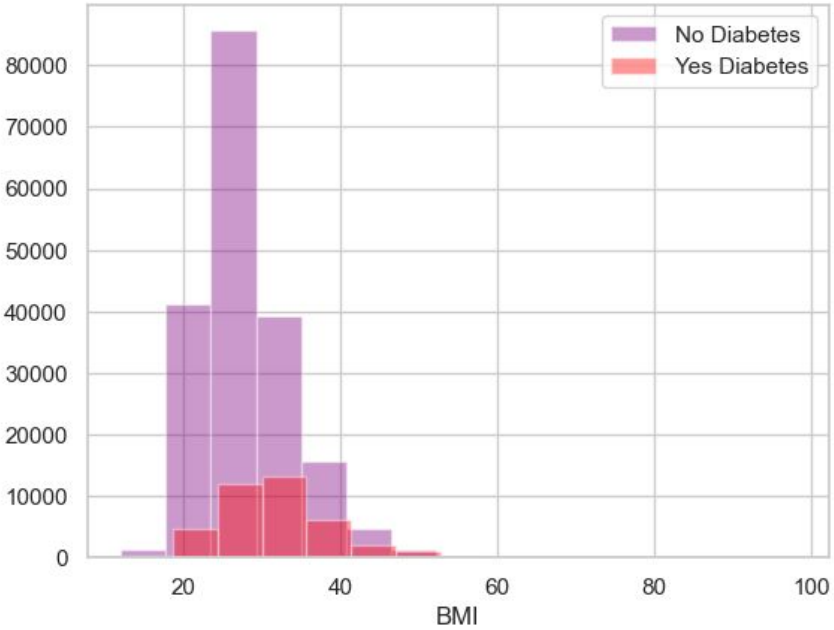
Link to the dataset here

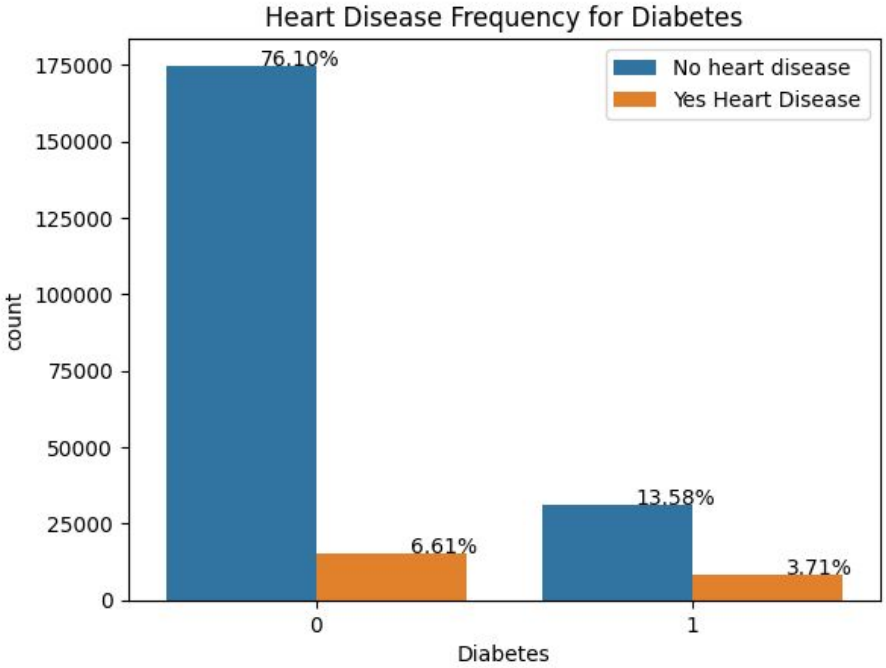
Overall Pipeline

- Exploratory Data Analysis (EDA)
- Pre processing
  - data cleaning
  - outlier detection
  - data transformation
  - duplicate values
- Handling Imbalance Problem
- Application of ML Algorithm
  - KNN, Decision Tree, Random Forest, XGCBoost
- Evaluation of the models (F1 Score, Accuracy, Precision)



EDA

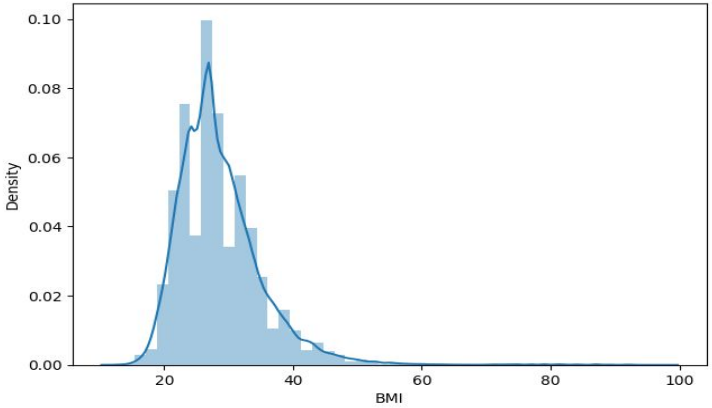


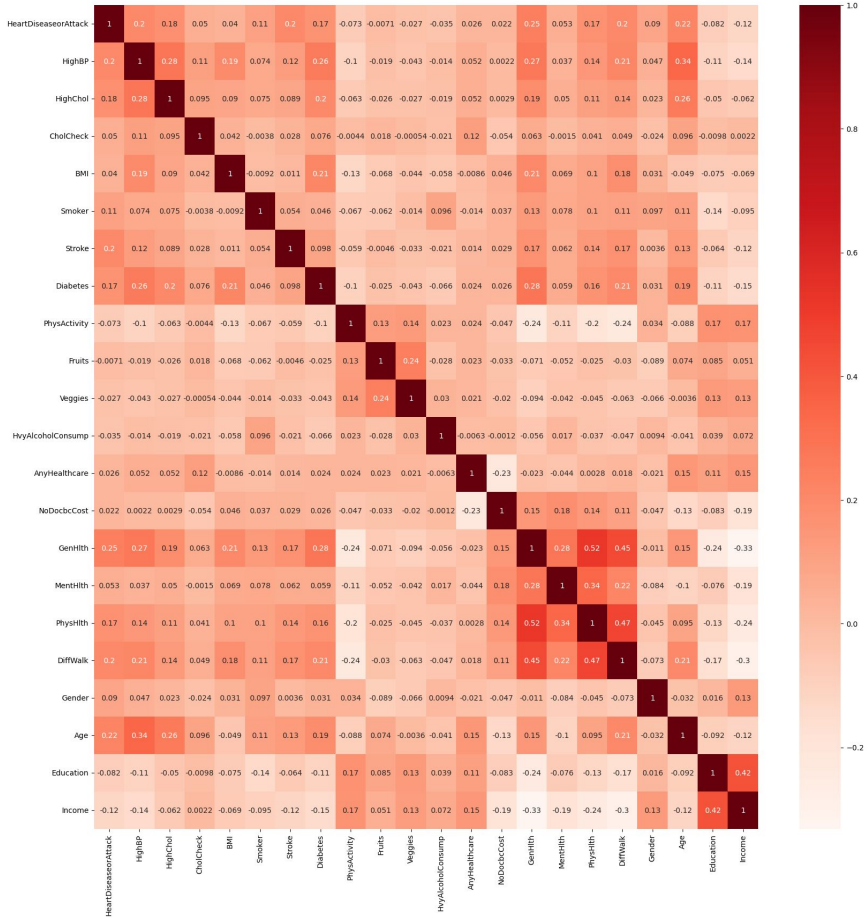


HeartDiseaseorAttack	0	1
GenHlth		
1	33898	1009
2	73499	4037
3	65873	7841
4	24822	6724
5	7972	4106

Pre Processing

- Not particularly high correlated features, features are all kept
- Removing outliers would mean, in our case, removing relevant data.
- Drop duplicate values
- Features from float64 to int32





Sampling Method

Methods:

- Random Oversampling, Adasyn
- SMOTE (5 versions)
  - standard
  - Hybrid: SMOTE + Tomek Links
  - Hybrid: SMOTEENN
    - Last two points are implemented using two different resampling strategies
- Random Undersampling, NearMiss
- Hybrid: Random Oversampling + Random Oversampling
- Hybrid: SMOTE + Random Undersampling

Models

- K-Nearest Neighbors
- Decision Tree
- Random Forest
- XGBoost - Classifier

Metrics

- Models are evaluated focusing on true positives (sick patients) w.r.t to true negatives (healthy patients).
- Evaluation considers a trade off between accuracy and f1-score
- Also Precision has an important role in the choice of the best model

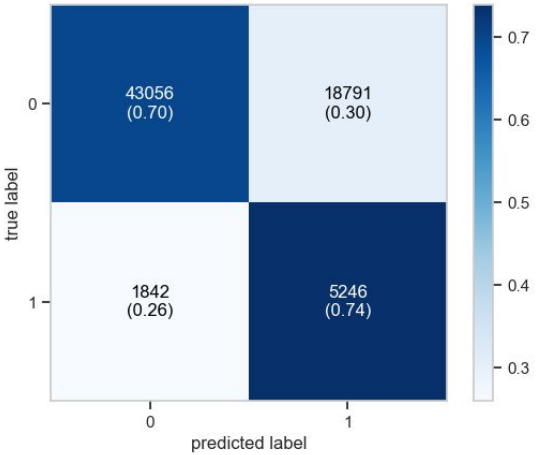
The Dataset

- Dataset Division: 70% to train the model, 30% to test it
- All models are trained with Cross Validation with cv=5 to help mitigating overfitting issue.
  - Except for KNN -> main issue is that it was really slow and the results were worse compared to other models

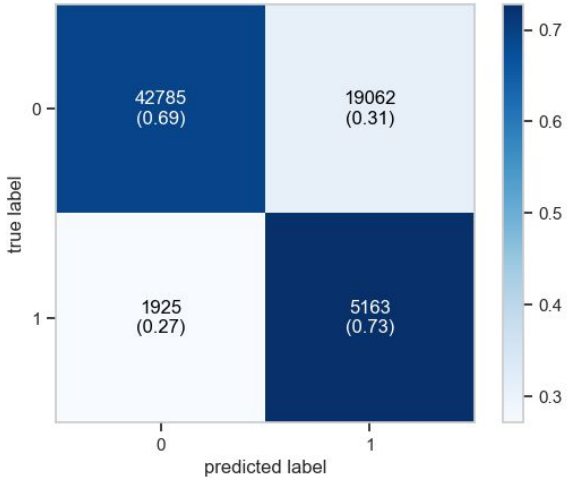
Results

Model	Accuracy	F1-Score	Precision
Random Forest	70%	(0) 81 (1) 34	(0) 96% (1) 22%
XGBC	70%	(0) 80 (1) 33	(0) 96% (1) 21%

Random Forest



XGBClassifier



Comments

- High Accuracy doesn’t mean necessarily good results
  - an high accuracy was given also when the number of true positives was very low, those cases were not taken into account
- With some sampling strategies such as random oversampling and random oversampling I obtained apparently good results, those strategies have two main drawbacks:
    - ROS - randomly duplicates samples in the minority class → model is not able to generalise well
    - RUS - randomly removes samples from the majority class → looses lot of info
- SMOOTEN with edited nearest neighbour is not that bad, but it was not as good as the other two versions (lower accuracy and f1-score).
- Precision is not as high as hoped but the number of samples classified but the number of false positives samples is much higher compared to the number of true positives.

Thanks for the attention!