Machine Learning - Project

Alessandro Pina
Blekinge Tekniska Hogskola
Karlskrona, Sweden
10486221@polimi.it

I. Abstract

The aim of this project is to independently explore and report on the application of machine learning to a real-world challenge.
In this project we try to predict if a person has been **a smoker during an extended period of his life**.
We use **supervised learning** techniques. More precisely, we are talking about a **classification** task.
After an initial phase of **data analysis** and **data preprocessing**, we compare different models, trying to identify the one that gives the best results.

II. Introduction

I started my work by browsing **Kaggle**[1] in order to find a dataset and a challenge suitable for this kind of project.
This task has not been as easy as expected, because I wanted to find something challenging that had not been already explored over and over by other people.
For this reason I tried to avoid popular competitions and I settled on the **"Young People Survey" dataset**, that contains data from a survey taken from over a thousand people in Slovakia during 2013.
This dataset contains answers to 150 different questions, ranging from music and movie preferences to phobias and demographics.
In the following pages I will often write 'smokers' instead of 'people that have smoked during their lifetime'. This is done for the sake of brevity.

III. Dataset

The "Young People Survey" dataset[2] contains **1010 rows** (data points), corresponding to the number of people that participated in the survey, and **150 columns** (features), corresponding to the number of questions asked, of which 139 were ordinal and 11 were categorical.
The numerical answers usually range from 1 to 5, with 1 indicating low importance (or disagreement) and 5 indicating high importance (or agreement).
All the participants of the survey were of Slovakian nationality, aged from 15 to 30 years old.
A survey as a source of data is extremely unreliable. People lie, and even if they don't, the majority of the answers are subjective and dependant on the interpretation of the question by the person answering it.
Another aspect to consider is the amount of questions that the people are asked to answer to. 150 questions is a lot, and people may be tempted to answer randomly just to complete the survey in a quick way.
Missing values are also contained in the dataset.

IV. Implementation

A. Programming language

I chose **Python** as the programming language for this project, given its wide popularity in the Data Science community.
I also used **Jupyter Notebook** to show and execute the code. This very useful tool can contain both computer code and rich text elements, making it an excellent choice for the task, given that it can be used as human-readable document for analysis and as executable document for the code.

B. How to run

To run the notebooks provided many roads can be taken. Two of them are:

- having Jupyter Notebook installed. The easiest way to get it is to download **Anaconda**[3], a free and open-source distribution for scientific computing. After having installed it, it is sufficient to run Jupyter Notebook and open the desired .ipynb file.
- using a cloud service like **Colaboratory**[4], a Google research project created to help disseminate machine learning education and research. It's a Jupyter Notebook environment that requires no setup to use and runs entirely in the cloud. It is sufficient to upload the .ipynb file and to modify the string that indicates the location of the data. After this, the code can be run very easily.

C. Files provided

Two files are provided:

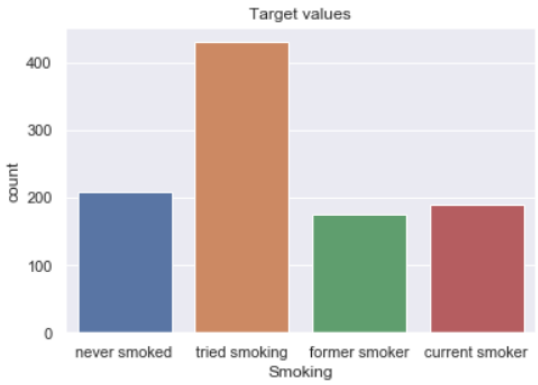
- *data_analysis.ipynb* needs to be run first. It contains the data preprocessing and data visualization steps.
- *modeling.ipynb* needs to be run second. It contains the modeling steps and the visualization of the final results.

[1] https://www.kaggle.com
[2] https://www.kaggle.com/miroslavsabo/young-people-survey
[3] https://www.anaconda.com/distribution/
[4] https://colab.research.google.com/

V. Data Preprocessing

A. Target variable

The target variable of this dataset is '**Smoking**' and it normally contains 4 different values:



For the purpose of this project, I grouped them into two categories:

- **'No'** for 'never smoked' and 'tried smoking'.
- **'Yes'** for 'current smoker' and 'former smoker'.

B. Encoding of categorical variables

I transformed all the categorical variables in numerical ones by encoding them with integer numbers.
This is done because some models necessitate features to be in a numerical form in order to properly work.

C. Outliers detection

A brief phase of outliers detection has been carried out. A wrong height value has been corrected.

D. Missing values

In order to make the data suitable for the training of the models, I identified and removed rows that had a missing value for the target variable.
For what it concerns the remaining missing values, I tried to **impute** them either by looking at correlated features or by inserting the most frequent value.

E. Binning of continuous variables

Binning refers to dividing a list of continuous variables into groups.
In order to improve the accuracy of the models, we binned features *'Age'*, *'Weight'* and *'Height'*.

F. Training and test set

It is important to remember to keep a part of the data separated, in order to be able to evaluate the accuracy of the models on unknown data.
In our case the data was shuffled and split in this way: **80% training**, **20% test**.
Stratification has been used to avoid obtaining unbalanced datasets.

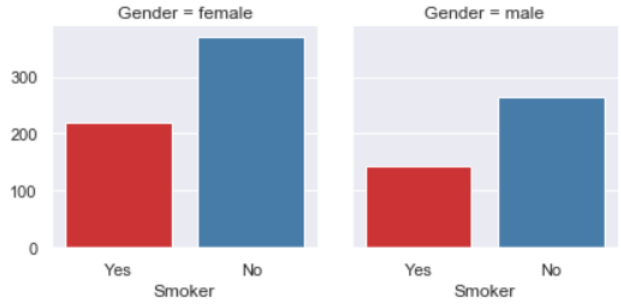
VI. Data visualization

A. What is it

Data visualization is the graphical representation of information and data. In this project I used visual elements like graphs and charts to better see and understand patterns in data. Python provides many libraries that greatly help the creation of these visual tools, making it easy to generate them with just a few lines of code.

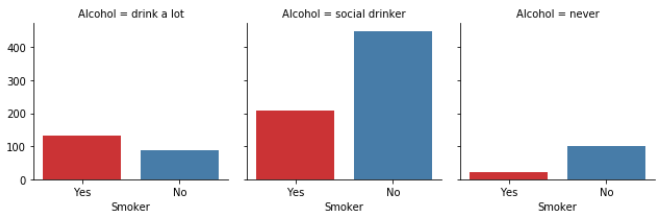
B. Insights gained

1) Gender: I computed all the **correlation coefficients** between features and plotted them, in order to try to gain some insights that would have been useful to improve the models.
One particular, very interesting, discovery was that, in this dataset, **gender** doesn't seem to influence whether a person is a smoker or not. Actually, females are slightly more likely than men to have been smokers during their life.



This is very strange, because by looking at national statistics about people of Slovakia, it can be noticed that men are more than twice as likely to be smokers than women.
This inconsistency may be due to the limited amount of samples available or to the fact that people were not being honest while answering this survey.

2) Alcohol: Another interesting correlation can be noticed between the consumption of alcohol and smoking.



Alcoholics are more likely to be smokers than the average drinker. People that do not drink usually do not like smoking.

VII. Baseline

It is important to establish the **baseline performance** on a predictive modeling problem.
A baseline provides a point of comparison for the other models that will be evaluated later. It also provides context on how good a given method actually is.
In this project I have used a **dummy classifier** that classifies everything as the most frequent label appearing in the training dataset.

Therefore, every test sample is classified as 'No' (not a smoker).
The accuracy of this model is **63.7%**.

VIII. Models

Four different supervised classification learning algorithms have been used:

A. Support Vector Machine

An **SVM model**[5] is a representation of the samples as points in space, mapped so that samples of separate categories are divided by a clear gap that is as wide as possible. New samples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

B. Random Forest

Random forest[6] is an ensemble learning method that operates by constructing a multitude of decision trees, each one with a different training dataset, at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

C. AdaBoost

AdaBoost[7], short for Adaptive Boosting, is a machine learning meta-algorithm that can be used in conjunction with many other types of learning algorithms to improve performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier.
AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. It is sensitive to noisy data and outliers. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing, the final model can be proven to converge to a strong learner.

D. Logistic Regression

Logistic regression[8] predicts the probability of an outcome that can only have two values. The prediction is based on the use of one or several predictors (numerical and categorical).
A logistic regression produces a logistic curve, which is limited to values between 0 and 1. This makes it suitable for classification problems. Logistic regression is similar to a linear regression, but the curve is constructed using the natural logarithm of the odds of the target variable, rather than the probability.

[5]https://en.wikipedia.org/wiki/Support-vector_machine
[6]https://en.wikipedia.org/wiki/Random_forest
[7]https://en.wikipedia.org/wiki/AdaBoost
[8]https://en.wikipedia.org/wiki/Logistic_regression

E. Discarded models

At first I also tried using other models, like **Nearest Neighbors**, **Naive Bayes** and **MLPClassifier**. The results were not particularly good, probably because of the low amount of samples available or the high correlation between some features.
For this reason I decided to focus only on the best models, keeping only the best four in order to have a more thorough comparison between them.
I left the code containing the discarded models commented, in case someone wanted to see how they performed.

IX. Hyperparameters tuning

A. What is it

In machine learning, **hyperparameter optimization**[9] (or tuning) is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A **hyperparameter** is a parameter whose value is used to control the learning process. By contrast, the values of other parameters are learned.
The same kind of machine learning model can require different constraints, weights or learning rates to generalize different data patterns.
Hyperparameter optimization finds a **tuple of hyperparameters** that yields an optimal model, which minimizes a predefined loss function on given independent data.

B. Implementation

To tune the hyperparameters of the models I used **GridSearchCV**, a very useful tool provided by the library **sklearn**. After the definition of a parameter grid, the algorithm performs **cross-validation** to select the combination of parameters that gives the best result.
Of course this procedure could be quite **computationally intensive** (depending on the dimension of the hyperparameter space), so in the code I have already specified in the definiton of the models the optimal parameters that I have found during this step.
This means that if the code is run without any modification, the tuning will not be performed (because all the hyperparameters are already tuned) in order to reduce the execution time of the notebook.
To see this process at work, it is enough to uncomment the lines of code contained in the 'parameters' array. This structure contains one dictionary for each model, where every key indicates the name of the parameter and every value is an array containing the values that are going to be tried.

X. What is to be expected

A. Accuracy improvement

Knowing the nature of the dataset and the limited amount of the samples available, I expect that it will be very hard to achieve more than a **10% improvement** over the baseline accuracy.

[9]https://en.wikipedia.org/wiki/Hyperparameter_optimization

B. Best model

Given that we have only a limited amount of data points, there are two main issues to take into consideration:

- **Overfitting**, with only a few data points, there is a higher risk to overfit the model.
- **Outliers**, with a lot of data, a couple of outliers will not be a problem. But with only a few data points, like in our case, they may cause some problems.

To avoid overfitting, we need to **avoid complexity**. As a result, we need a simpler model with a limited number of parameters. **Logistic regression** could be what we need. **SVM** may also work well.

For what concerns outliers, in our case we should be fairly shielded by their effect, given that the majority of the answers to the survey ranges from 1 to 5. As a precaution, I also grouped numerical features like age, height and weight in 5 groups, depending on their value.

XI. Results

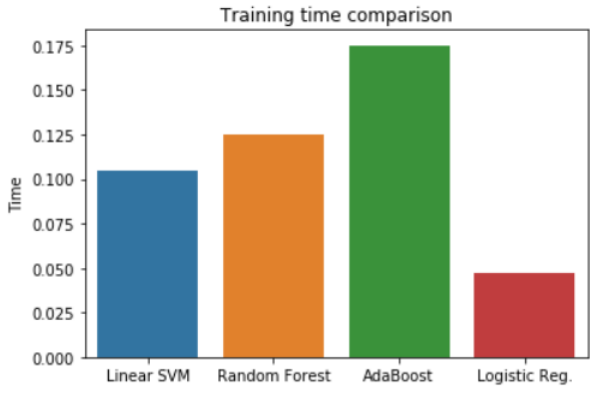
A. Ranking models

	SVM	RF	AdaB	LogReg
Time	0.12 (2)	0.13 (3)	0.17 (4)	**0.05** (1)
Accuracy	67.2 (4)	67.7 (3)	68.2 (2)	**70.1** (1)
Precision	68.9 (3)	67.0 (4)	70.8 (2)	**72.4** (1)
Recall	88.3 (2)	**96.9** (1)	85.2 (4)	85.9 (3)
F1 score	77.4 (3)	**79.2** (1)	77.3 (4)	78.6 (2)
MCC	22.9 (4)	23.7 (3)	26.7 (2)	**31.8** (1)

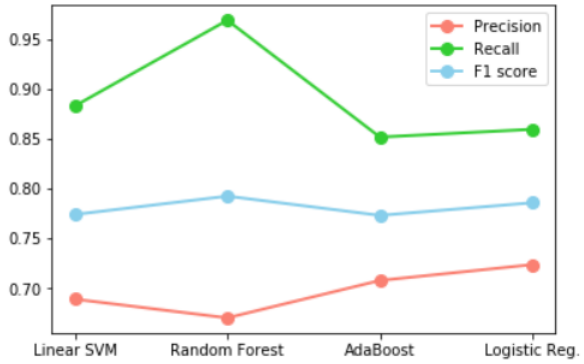
B. Metrics

I used these different metrics to evaluate the models:

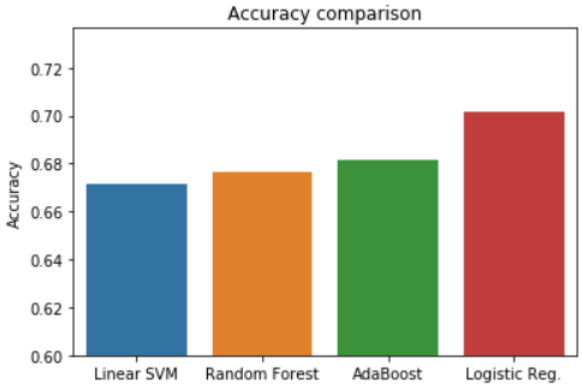
- **Time**: the time (in seconds) spent to train the model.



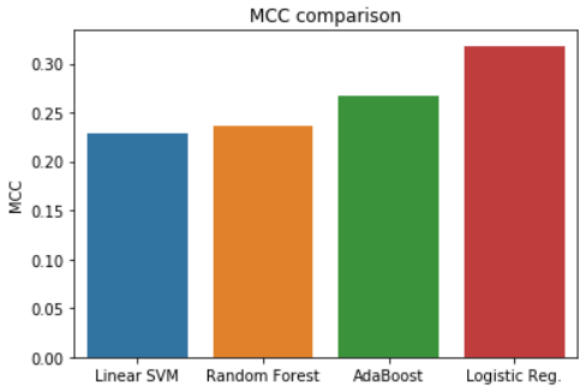
- **Precision**: the number of correct positive results divided by the number of positive results predicted.
- **Recall**: the number of correct positive results divided by the number of all samples that should have been identified as positives.
- **F1 score**: the harmonic mean between precision and recall, ranging from 0 to 1. It tells how precise the classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).



- **Accuracy**: the ratio of number of correct predictions to the total number of input samples.



- **MCC**[10] **(Matthews Correlation Coefficient)**: a coefficient that takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. It returns a value between 1 and +1.
  A coefficient of +1 represents a perfect prediction, 0 indicates random prediction and 1 indicates total disagreement between prediction and observation.



[10] https://en.wikipedia.org/wiki/Matthews_correlation_coefficient

XII. Conclusions

A. Accuracy improvement

As expected, we did not manage to improve the baseline classifier accuracy by a lot (+6.4%), due to the conditions previously considered.

B. Best model

I would say that best model between the four considered is, by a small margin, the **Logistic Regression** one.
It reports an accuracy of **70.1%** over the test data, 6.4% better than the baseline model (63.7%) set at the start of the project.
On top on this, it's also the **fastest** model to train (0.05 sec) and the **most precise** (72.4%).
Its **F1 score** is second only to the Random Forest one, due to RF's very high recall.
I consider **MCC** as the most reliable metrics in this case, because it also takes into account the true negatives. Logistic Regression has the best MCC between the four models considered (+31.8%).

C. Information acquired

What can be deduced from what I have found out during this project?
The main issue of this dataset is that, as stated before, the data is not good. No model can perform well if the input data is flawed. 'Garbage in, garbage out' applies in this case.
Nevertheless, this project could be a start towards the construction of a better survey.
In a next improved version of this survey, it may be a better idea to remove questions that were discovered to be redundant, i.e. had a very high correlation with other questions.
In this way the number of questions may be reduced, with the direct consequence of an higher focus of the participants and better answers on the remaining questions.
Additionally, an higher distribution of the survey would help gathering more data, allowing the development of a better model that could aid an eventual targeted campaign to reduce smoking between young people.

D. Final considerations

I'm very pleased with how this assignment turned out to be. By working on this project, I grew accustomed to a variety of instruments that will certainly be an important part of my Data Science toolbox.
Even if the results do not look exceptional, I feel that, given the constraints "imposed" by the nature of the dataset, I have been able to reach a satisfying result.