

**Model for password security**

**Matteo Gianvenuti**

**Donato Bruno**

Summary

[Problem description 3](#_Toc174466345)

[Technologies used 3](#_Toc174466346)

[Dataset 3](#_Toc174466347)

[Accuracy 3](#_Toc174466348)

[Bibliography 4](#_Toc174466349)

# Problem description

Create a classifier model to determine the strength of a password. In practice, the model must establish which class the password belongs to, choosing between five classes: "Too weak", "Weak", "Moderate", "Strong", "Very strong".

## Technologies used

We used the library scikit-learn which is an efficient tool for machine learning in Python. We tested different classifier models from the library to understand which is the most efficient to choose it.

* Gradient Boosting Classifier (ensemble): It is an additive model, which adds decision trees one at a time, trying to improve the model step by step. Based on negative gradient of the loss function.[[1]](#GradientBoostingClassifier) (An ensemble model combines predictions from multiple models to improve overall performance over that achieved with a single model).
* Random Forest Classifier (ensemble): It is based on a set of independent decision trees, trained on different subset of the training set. It uses averaging to improve the predictive accuracy and control over-fitting.[[2]](#RandomForestClassifier)
* K Neighbors Classifier (neighbors): A classifier implementing the k-nearest neighbors vote. The classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point.[[3]](#KNeighborsClassifier)
* Support Vector Classifier (svm): It is a technique based on the Support Vector Machines, which try to find a hyperplane that best separates classes in a dataset.[[4]](#SVC)

## Dataset

## Accuracy

We have tested and evaluated all these models to find the best one for our task.

|  |  |  |
| --- | --- | --- |
| Model | Acc. with stratification | Acc. without stratification |
| Gradient Boosting Classifier | 75% | 68% |
| Random Forest Classifier | 75% | 64% |
| K Neighbors Classifier | 71% |  |
| Support Vector Classifier |  | 66% |

# Bibliography

[1] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html#sklearn.ensemble.GradientBoostingClassifier>.

[2] <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>.

[3] <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>.

[4] <https://scikit-learn.org/stable/modules/svm.html#svm-classification>.