The aim with this course is to develop a deeper understanding of a subarea in computer science. We choose machine learning and how it can be used in real life scenarios.

Introduction – 0.5 min

Fredrik, Julian.

We will explain what Kaggle.com is for a website, what our main problem is, the most common machine learning algorithm we used and the workflow to climb the leaderboards.

We will present our results with an interesting twist and then give you our conclusions.

Kaggle.com – 1 min

A website where users from all over the world compete to produce the best machine learning models. The website also provides all kinds of interesting datasets in a non-competitive environment.

Users range from new amateur programmers to PhD students.

Many organizations don't have access to advanced machine learning that provides the maximum predictive power from their data. Kaggle offers companies a cost-effective way for this problem, and at the same time give data scientists and statisticians real-world data to develop their techniques.

Company or organizations can make competitions and provide money for example the top 3 best entries. In our competition Expedia there is $25,000 distributed for the top 3.

You compete as a team. Teams consist of 1 or more team mates.

Submissions are made in .csv (comma separated value) files with specific columns and the submission is scored according to a known scoring function. The score is public as part of a leaderboard.

Some competitions, as the one we are part of, only allows up to 5 submissions per day.

Random Forest Classifier – 2 min

[1]
Supervised learning classifier. The algorithm combines random decision trees with bagging, where each model votes with equal weight, to achieve very high classification accuracy.

Random sub-samples are taken from the data. Some data can overlap in different sub-samples. Output is classification for each sample. [pekar i bilden]

The algorithm generates decision trees on each sub-sample.

[2]
The algorithm then summarize all the decision trees combining it into a forest, hence the name. The classifier will consist of many different decision trees to predict an output.

Mistakes are taken care of.

One good thing with Random Forest is that it avoids over fitting the data. A model that has been over fit has poor predictive performance, as it overreacts to minor fluctuations in the training data.

[3]
So how can Random Forest be so effective? This is what makes Random Forest work.

1. Most trees provide correct prediction for the most part of the data.
2. Tree make mistake at different place.

Given an input, we want to classify it. Since most of the decision trees classify C1, this must be the correct one [pekar i bilden].

Results – 1 min

[1]
Random Forest classifier: 0.18584
most popular local hotels: 0.30090

Leak – 2 min

Some weeks ago a leak became public. Leak is when predictions can be made by looking at training data to help to predict test data. Competition Admin confirmed the leak. The leak affects 1/3rd of the testing data.

You can find *hotel_clusters* for the *affected rows* by matching rows from the *train* dataset based on the following columns:

user_location_country, user_location_region, user_location_city, hotel_market and orig_destination_distance

However, this will not be 100% accurate because hotels can change cluster assignments, hotels popularity and price have seasonal characteristics.

Now the competition score is a mix, 33% of a puzzle scoring and the rest of a true model.

Results – Leak

[1]
Using a more advanced approach with most popular hotels and leakage we got:
0.50050

Conclusion

[1]
Machine learning in real-life situations to optimize

It is important to not leak training examples into test set. Especially not when training a model to use in real-life situations.

As of now, the best model will have to find the leak (1/3) and train itself to catch the rest of the holdout data (2/3).