FRI Data Science Project Competition 2023
Interim report

Feature selection on large and sparse datasets

Leon Hvastja, Amadej Pavšič

Abstract

Feature selection is a common problem in machine learning, especially when dealing with sparse datasets. In this project, we explore various out-of-the-box algorithms for feature selection, including those based on simple machine learning models and those derived from the relief algorithm. Our task is to implement, develop, and test these algorithms for smarter feature selection, enabling quicker and more efficient feature selection than using greedy algorithms. We will test the accuracy of our developed approaches against the optimal solution found by the greedy algorithm.

Keywords
feature selection, feature ranking, machine learning, big data, sparse features

Advisors: Jure Demšar, Blaž Mramor, Blaž Škrlj

Introduction

Feature selection is the process of reducing the number of input variables when building a predictive model. The aim is to reduce computational cost and in some cases improve model accuracy.

Our project aims to explore feature selection methods for Outbrain, a web recommendation platform. They provided us with data from their regular operations, which is typically large, sparse and requires time-efficient use. The dataset underwent minimal preprocessing specific to our project. We were also provided with their internally generated feature scores to use as a reference.

We hypothesize that simpler models can produce quick feature selection results, but models that account for feature interactions will be more accurate at the expense of longer computations. Additionally, we anticipate that a heuristic feature pre-selection can improve processing time for some algorithms without sacrificing accuracy.

The main goal of our project is thus the development of an efficient and intelligent feature selection approach given the specifics and constraints of Outbrain's domain of work.

Exploratory Data Analysis

The main purpose of exploratory data analysis (EDA) is to familiarize ourselves with the data and start developing ideas for intelligent preprocessing. The dataset logs ad clicks and comprises over 1.5 million samples with 100 features and a binary label indicating whether the ad was clicked or not. All features are anonymized with numerical labels such as **feature0**. The data contains only categorical features whose values were transformed using feature hashing making them impossible to interpret for us. There were no missing values in the dataset since these were also hashed into integers, but their specific values are unknown.

Cardinality is one of the crucial feature properties we analyzed in our EDA. We have observed that 7 features contain only one unique value, rendering them uninformative. 8 features as well as the label are binary, while 20 features have more than 1000 unique values. However, some features have very high cardinality, which may make them impractical for making predictions.

Distribution of feature values

The label exhibits an 80-20 ratio, indicating a bias toward the ad not being clicked. Note that our dataset was created by undersampling the original data, which had a label ratio closer to 200-1. Additionally, the features have varying value distributions, and we observed that 13 features have one highly prevalent value, occurring in over 90% of the samples. We inferred that these values very likely represent missing values, implying that the corresponding features are highly sparse.

Feature correlation

We utilized Pearson's coefficient (PC) to investigate the connections between features. Our observations revealed a few sets of highly correlated attributes, including those ranging

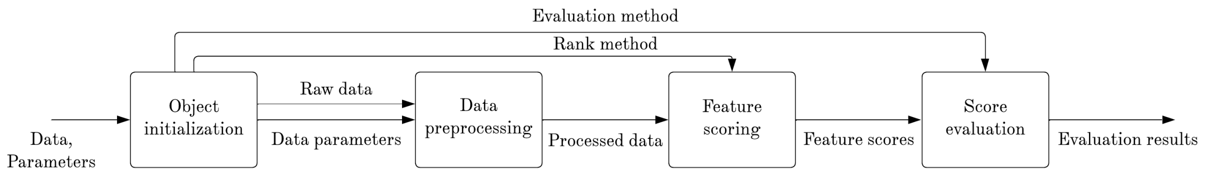

Figure 1. A visualization of our feature selection pipeline.
This figure visualizes the steps required for feature selection in our framework.

from 81 to 89, which are also negatively associated with the label.

It's worth noting that features 98 and 99, have a perfect correlation (PC = 1) with each other and the label, effectively making them a flawless predictor of the label. These characteristics were engineered into the data, so they cannot be used for our feature selection. But they can be useful for initial validation of our approaches as they should be on the top of the feature ranking process.

Rank and Evaluation Pipeline

We have developed a framework for evaluating features that streamlines our algorithm testing process. This pipeline handles data reading, pre-processing, feature score calculations, and evaluations against our ground truth ranking. Its flowchart is shown in Figure 1.

Preprocessing

To reduce the computational load, our initial strategy involves subsampling by rows. This means selecting a random subset of data points to work with. We aim to use this approach to gain insight into the behavior of our feature scoring algorithms with larger datasets, and possibly identify an effective subsampling strategy for very large datasets.

We also perform factorization as a preprocessing step. This involves transforming the original dataset's categories, which are represented by large integers, into a categorical form and re-encoding them as integers from 0 to n, where n represents the number of unique values for a particular feature.

Ranking algorithms

The algorithms we chose to base our approaches on fall into the following two categories: a) **Filter methods** (rank features independently of any machine learning algorithm) and b) **Wrapper methods** (use ML algorithms to implicitly rank features).

Our selection of filter methods includes widely-used approaches that rely on statistical tests, along with a suite of feature ranking algorithms derived from the Relief algorithm. The latter is capable of identifying feature interactions, but its time complexity is significantly higher than that of other methods.

The wrapper methods use some of the most common and simple ML models such as random forests and XGBoost. The aim here is to build simple models and extract the implicit feature importance from the model.

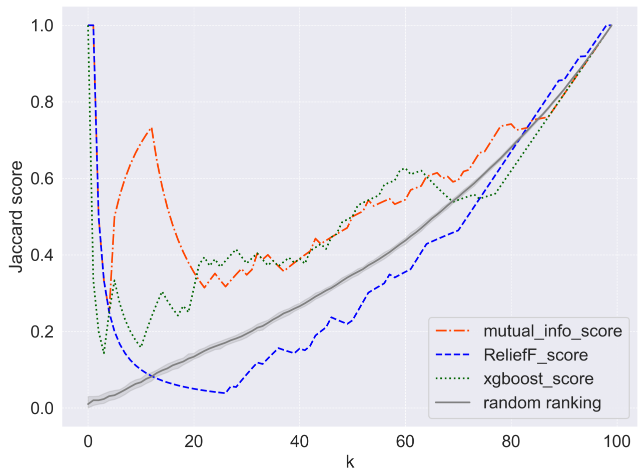

Figure 2. Ranking evaluation comparison. Jaccard score for each top k features for three different ranking algorithms and randomly permuted ranking as a baseline.

Ranking Evaluation

We utilize the Jaccard index to assess our outcomes. Our pipeline output is an array of 100 scores representing all potential top k feature assessments. The score assigned to the i-th position provides insight into the percentage of overlap between the top i features in our generated ranking and the ground truth in our reference set. A visual example of such evaluation is presented on Figure 2.

Results

Our current results have shown us the feasibility of individual algorithms. We have also identified algorithms that appear to perform poorly for our dataset regardless of undersampling rates. We've identified promising algorithms, which will now be analyzed in greater depth.

Discussion

To start our project, we focused on establishing baselines and evaluating various algorithms to develop a strategy for future assessments. Since our computational resources are limited, we had to narrow down our candidates. In the next phase, our attention will be on identifying the optimal feature ranking strategy for our dataset including subsampling rates, feature pre-selection and ranking algorithm choice.