Capstone Proposal

Machine Learning Engineer Nanodegree

Andrea Balzano

10th April 2017

The Search for New Earths

Project overview

This project analyses the data observed by the NASA Kepler space telescope searching for exoplanets using the transit technique.
planets themselves do not emit light, but the stars that they orbit do. If said star is watched over several months or years, there may be a regular 'dimming' of the flux (the light intensity). This is evidence that there may be an orbiting body around the star; such a star could be considered to be a 'candidate' system.
NASA itself utilises python to interpret the data and has created PyKE, a library for data reduction to help with extraction and preprocessing of the light curve images, however this project analyses only FLUX data, not pictures.
Some of the machine learning techniques already been used by developers are 1-D CNN, XGBoosting, PCA.

Problem Statement

The goal is to create an agent able to classify candidate systems.
At the time this dataset was prepared Campaign-3 was unlikely to contain any undiscovered exoplanet-stars. Therefore, all stars (i.e. all rows) which were not confirmed to host an exoplanet, were labelled with a 0. This is over 99% of the observations.
In total, there are 42 observations of confirmed exoplanets labelled with a 1, 5 in the test set and 37 in the train set.
The task involved are the following:

1 Download the dataset.
2 remove outliers and normalise the data.
3 correct for class imbalance
4 run preprocessing analysis.
5 run classification algorithm.

The output of the classifier is a binary class where a system not containing exoplanets is labelled 0, and a candidate for more investigations is labelled 1.

Datasets and Inputs

The dataset I will use is available on Kaggle.
https://www.kaggle.com/keplersmachines/kepler-labelled-time-series-data

In details, the Trainset includes:
- 5087 rows or observations.
- 3198 columns or features.
- Column 1 is the label vector. Columns 2 - 3198 are the flux values over time.
- 37 confirmed exoplanet-stars and 5050 non-exoplanet-stars.

The Testset includes:
- 570 rows or observations.
- 3198 columns or features.
- Column 1 is the label vector. Columns 2 - 3198 are the flux values over time.
- 5 confirmed exoplanet-stars and 565 non-exoplanet-stars.

I will utilise part of the trainset to do my validations before evaluating the performances with the test set.

Solution Statement

This dataset has very high dimensionality represented by the time series of the FLUX measurements, I will first analyse the data looking for outliers that can be due to noise or sensors anomalies, I will approach the detection using Tukey's Method for identifying outliers, where an outlier step is calculated as 1.5 times the interquartile range (IQR).
I Choose his method because it isn't dependent on distributional assumptions, and It also ignores the mean and standard deviation, making it resistant to being influenced by the extreme values in the range.
I will filter all the stars with measurements under/over the outlier step looking for periodicity on these values to try isolate noise and decide on a case basis if to keep the record, remove it or replace the value of the outlier with the median value for that column.
I will address the class imbalance by applying SMOTE + Tomek links to try to balance the tradeoff of over/undersampling.
The SMOTE technique increases the size of the minority class by creating samples in the neighbours, correcting for the tendency to overfit to the samples common during oversampling.
The Tomek links is a redundancy driven technique to reduce sampling size, it looks for Tomek-links which consist of points that are each other's closest neighbours, but do not share the same class label.
For a binary class, the number of possible combinations grows exponentially with the number of dimensions, 2^n where n is the number of features, this is also known as the curse of dimensionality.
I will apply features transformation with PCA to capture the delta changes over time, reducing the number of features and compensating for the curse of dimensionality.

I will try a number of classifiers to get a benchmark, then I will explore an algorithm and fine tune it using cross-validation on the training set.
Ones the score on the validation set is acceptable I will evaluate the performance on the testset.

Benchmark

This project could be defined as 'A Search for a Needle in a Haystack' therefore I wouldn't expect an algorithm that can classify systems with high accuracy, I will consider a precision on class 1 of 30 % a good benchmark for this project.

Evaluation Metrics

class 0 includes 99% of samples, due to this high class label imbalance, the metric used will be the precision and recall scores for class 1.
Precision is the fraction of predicted observation that is correct.
The recall is the fraction of the true observations that are successfully predicted.
In this example, precision is how many predicted candidate system are actually a candidate and recall is how many candidate systems we are correctly identifying in the full set of candidates.

Project Design

1. Outliers detection using Tukey's Method for identifying outliers.
2. Class imbalance correction by applying SMOTE + Tomek links.
3. Features transformation with PCA.
4. Creation of a benchmark classifier, I will implement a simple classifier in naïve Bayes, SVM, decision tree and use the best model as benchmark.
5. Develop a Neural Network classifier using Tensorflow.
6. Cross validation on the training set as validation.
7. Classify testset.