10ACADEMY

DATA SCIENCE TRAINING

WEEK 5

Competitions and Data Challenge Tricks

By:

Gedeon MUHAWENAYO

August 11, 2020

1 Outlines

- General introduction
- Understanding the Competition
- Visualizing the data
- Data Pre-processing
- Creating a Baseline model
- Improving the model
- Hyper-parameters
- Overfitting
- Model Regularization
- Cross-Validation
- Kernel
- Transfer Learning
- Reproducibility
- Robust model(win)

2 General Introduction

Data competitions serve many purposes. One of the radical benefits is that they are the perfect place to learn best practices, accrue feedback on your work, and augment your skills. They can also serve as a channel for problem-solving and brainstorming by probing the multitude of crowdsourced solutions to big problems.

Moreover, data competitions are an opportunity to push boundaries and encourage creativity among the best and the brightest in a variety of data-related fields.

The experience you get is invaluable in preparing you to understand what goes into finding feasible solutions for big data.

Data science competition platforms like Zindi and Kaggle are commonly used to tackle social problems in different sectors such as Agriculture, Healthcare, Environment and normal Business decisions. We will end our sessions with a data science competition.

Skip this steps if you already have account on Zindi and Kaggle
Click here to sign up on Zindi
Click here to sign up on Kaggle

Note: Data challenges and competitions are one of the best way to learn, and to show your potential to the World. Always remember to tweet (or LinkedIn post) your ranking within a data science competition.Furthermore, you can make money from the competitions

3 Understanding the Competition

The most popular tasks in data challenges are classification and regression. Fundamentally, classification is about predicting a label and regression is about predicting a quantity. Classification is the problem of predicting a discrete class label output for an example. Regression is the problem of predicting a continuous quantity output for an example

Before starting any data science competition, try to understand the challenge very well. The first step is to go to the competition platform and read the description of the competition.

The **description** gives an introduction into the competition's objective and the sponsor's goal in hosting it.
The **data** tab is where you can download and learn more about the data used in

the competition. You'll use a training set to train models and a test set for which you'll need to make your predictions.
The **evaluation** section describes how to format your submission file and how your submissions will be evaluated. Each competition employs a metric that serves as the objective measure for how competitors are ranked on the leaderboard.
The **timeline** has detailed information on the competition timeline. Most Kaggle Competitions include, at a minimum, two deadlines: a rules acceptance deadline (after which point no new teams can join or merge in the competition), and a submission deadline (after which no new submissions will be accepted). It is very, very important to keep these deadlines in mind.

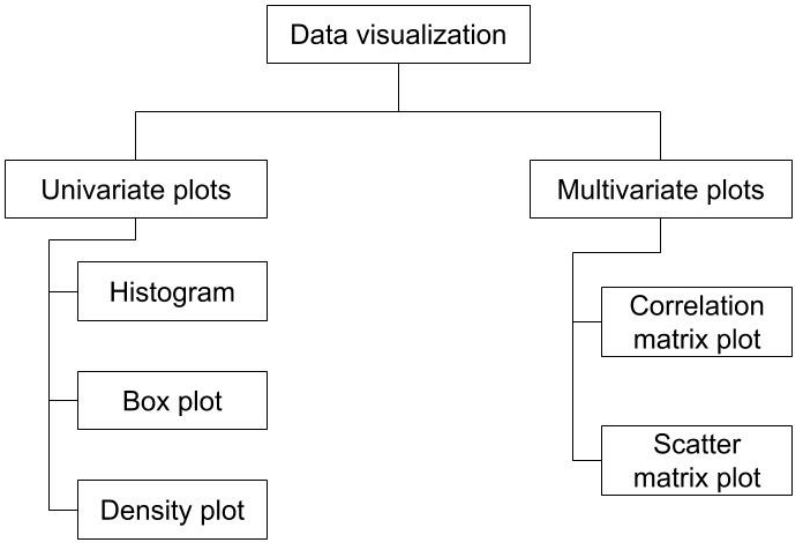
TASK 1: Here are the popular evaluation metrics: Log loss, Accuracy, root mean square error, mean absolute error and F1 score. Your task is to write a summary on how they differ from each other and how they are related to the leaderboard positions (the higher the better or the lower the better, etc.)
The following links would be very helpful LINK1, LINK2
TASK 2: For the following data challenges, explain the possible evaluation metrics that could be used.

- Predicting wine price from customers review
- Churn prediction (whether a customer will go or stay)
- House price prediction
- Climate forecasting
- Predicting traffic jam
- Predicting plantation diseases from drone images

Note: Both TASK1 and TASK2 should be in a single report of 2pages maximum.

4 Visualizing the data

Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.



Univariate plots: These are the simplest type of data visualization. With the help of univariate visualization we can understand each attribute of our dataset independently.

- Histograms group the data in bins and is the fastest way to get idea about the distribution of reach attribute in dataset.
- Density plot is another quick and easy technique for getting each attribute's distribution
- Box plot, Box and Whisker plot is a useful technique to review the distribution of each attributes.

Multivariate plots: These are data visualization techniques that help us to understand the interaction between multiple attributes of our dataset.

- Correlation matrix plot, helps to understand the changes between two variables. Far example it can indicate what happens to the price of wine if the

bottle size increase.

- Scatter matrix, shows how much one variable is affected by another or the relationship between them with the help of dots in two dimensions.

TASK 3: For the following data visualization objectives, explain clearly the type of visualization that could be be used, use examples.

- Understanding if you have class imbalance in the dataset
- understanding the attributes that affect the target
- Checking if your dataset is linearly separable

5 Data pre-processing

In machine learning and data science, we always need to feed the right data i.e the data in correct scale, format, and containing meaningful features, for the problem we want to model.
This makes data preparation the most important step in ML process. Data preparation may be defined as the procedure that makes our dataset more appropriate for ML process.
In a broad sense, data pre-processing will convert the selected features into a form we can work with or can feed to the learning algorithms.

Data pre-processing techniques:

- **Scaling**: Data recaling makes sure that the attributes are at same scale. Generally, attributes are rescaled into the range of 0 and 1. We can rescale the data with the help of **MinMaxScaler** class of **scikit-learn** Python library.
- **Normalization**: Normalization is use to rescale each row of data to have a lenght of 1. It is mainly used in sparse dataset where we have lots of zeros. We can use **Normalizer** class of **scikit-learn** Python library.

- **Standardization**: This is used to transform the data attributes with a Gaussian distribution. It differs the mean and standard Gaussian distribution with a mean of 0 and a standard deviation of 1. We can standardize our date(mean=0, SD = 1) with the help of **StandardScaler** class of **scikit-learn** Python library.

- There exists a lot of data preprocessing techniques, click here for more.

Note: In many data challenges the given dataset would be unstructured. Wikipeadia defines unstructured text data as data is not organised in a predefined manner. Unstructured information has text , dates , numbers and facts. These make for irregularities and ambiguity which makes it difficult for the machine to understand the raw text. So before applying some data preprocessing you have to create the numerical representation for the text data, you could use tokenization, or word embeddings, tf-idf, or bag of words. Check this LINK

6 Creating a Baseline model

In some competitions they will provide a starter notebook, that include how you could load the data, model steps and creating submission.
However it is really good to create your own baseline, the following are the steps that most people use:

- Imports: Iclude the packages that you need, in most cases these are Numpy, torch, Pandas, sklearn etc. Note that in most competitions you are only allowed to use publicly available libraries.

- Set the seed and args parameters: The **seed()** method is used to initialize the random number generator. The random number generator needs a number to start with (a seed value), to be able to generate a random number. By default the random number generator uses the current system time. args parameters will help you to control hyper-parameters.

- Data pre-processing: Load the data and do basic pre-processing, you will

spend too much time on features processing later, at this step do basic processing.

- Simple model first: Start with simple model, such as Logistic regression, Linear Regression or Naive Bayes. Note that the model depends up on the task.
- Make prediction, after training and evaluating your model make prediction with the provided test set. To evaluate your model, use cross-validation.
- submit your prediction: This step helps to be sure that the format that you are using is the right one and your submissions will be accepted.

Data challenge 1: Download the dataset here, and try to do the following tasks:

- Create a notebook(.ipynb)
- Read the given data, both train, test and sample submission. Remember to use packages like pandas and numpy
- Preprocess data, using the techniques we described, preprocess the given data (convert text to numbers, scale data, normalize data and standardize them).
- Visualize your data, for at-least 10 features visualize their univariate plot as they related to the target(wine price). Make a correlation plot to see the features which correlate more with the target, this is an important step in all data competition as it helps in feature selection. Note: Remember to handle the NAN (empty cell) you can replace them by column mean.
- Split the given train data into training set and validation set
- From sklearn select an appropriate model, make sure that your model is very simple. Use your training set to train the model and the validation set for evaluation, remember to use cross-validation.
- Predict on the given test set and from your predictions create a submission file, make sure your submission file is in the same format as the given sample submission.

- Using markdown in your notebook answer the following questions:

    1. List top 3 features which correlate more with the target.
    2. Write your validation accuracy
    3. Explain if your model overfits the training set or whether it underfits
    4. Suggest what could be the next step in order to improve your model.s