Machine Learning

100 QUESTIONS

THOMPSON WONG

Introduction

MACHINE LEARNING

THOMPSON WONG

This book is written during the learning of machine learning, it summarized the frequent questions asked by junior or the major points that make people confuse. The original plan is 100 questions, it may be adjusted depends on the practice. Hope it will help you in machine learning field. Enjoy !!!

ABOUT AUTHOR

Thompson Wong
Data Scientists | Machine Learning | Startup
https://kinshing0930.wixsite.com/portfolio

Thompson, experienced with data analysis such as equipment optimization, fault detection and diagnosis, performance evaluation; software development includes analytics framework development, machine learning pipelines and application, RESTful API, optimization algorithm design; research and development for next generation technology; dashboard solution using Grafana.

Benefiting from the various working experience in different scale company, a solid experience and good soft skill is developed such as engineering mindset for problem solving, design thinking for the product development, efficiency and effectiveness in software features delivery, leadership to coordinate the works from different stakeholders, communication between client request and team capability.

Communication to digest the needs, design thinking to make a innovation and breakthrough, engineering to solve problem step by step, leadership to coordinate the colleagues and make everyone satisfied in their job assignment.
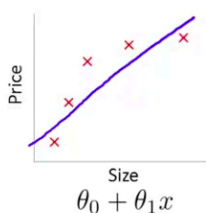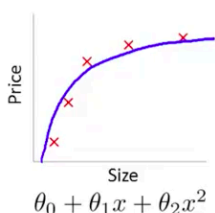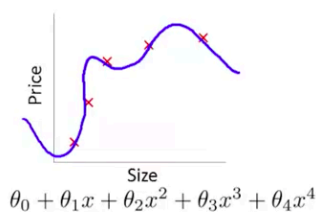
QUESTION 1

BIAS AND VARIANCE

Bias and variance is a basic concept to illustrate the machine learning model performance, like overfitting and under-fitting. High variance also means high uncertainty.

$\theta_0 + \theta_1 x$

$\theta_0 + \theta_1 x + \theta_2 x^2$

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High bias (underfit)

d=1

"Just right"

d=2

High variance (overfit)

d=4

The bias usually represent the underfitting, as insufficient features makes prediction bias in training set. The variance usually represent the overfitting, too much features make machine learning model only perform well in training set, and get fail in newly unseen dataset or test set or validation set. You can comprehend as not generic model.
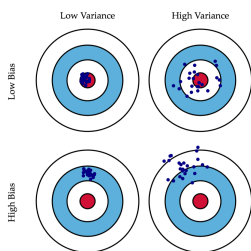
QUESTION 2

REGULARIZATION

Regularization is an effective things that control the model fitting performance. It is divided by L1 and L2 regularization. The formula is generally written in:

L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda\sum_{j=0}^{M}|W_j|$$

L2 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda\sum_{j=0}^{M}W_j^2$$

Loss function   Regularization Term

The L1 regularization is usually used in feature selection, L2 regularization is usually used in avoiding overfitting in machine learning. L1 regularization reduce model complexity via feature selection (reducing coefficients parameters/ features amount), L2 regularization reduce complexity by controlling coefficients parameters values.

QUESTION 3

DATA-CENTRIC OR MODEL-CENTRIC?

Data quality is always the major factors affecting the machine learning model performance. Garbage in, garbage out. Spending time for controlling data quality is much better than model algorithm. The existing algorithm is good enough, we all standing on the giant's shoulders.

When you have good enough data quality and quantity, the modelling result will be close to each others in different machine learning algorithms. So don't worry on selecting algorithm or developing new algorithm. Focus on improving data quality, Data-centric AI is greater than model-centric AI. The below figure illustrate the relationship between data quantity, quality and its performance
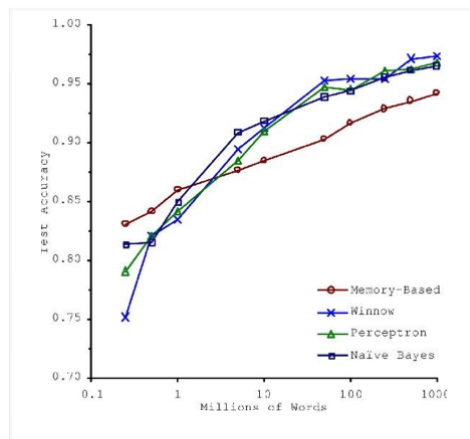
QUESTION 4

MODEL-BASED VS INSTANCE-BASED LEARNING

Model-based usually knew as supervised learning such as SVM (support vector machine), instance-based learning usually knew as unsupervised learning such as kNN.

In model-based learning, once model is built, the training data can be discarded (e.g. learned weights and bias value in SVM: y=mx+c) while instance-based learning use the entire dataset as the model, and looks at the close neighbourhood of the input example in the space of feature vectors and outputs the label that it saw the most often in this close neighbourhood.

Fundamentally, the model-based and instance-based learning has no direct relationship with the terms of data-centric and model-centric. Don't get confused even the terms are similar.

QUESTION 5

DIFFERENCES BETWEEN RF, XGBOOST, LIGHTGBM, GBDT

XGBoost, lightGBM, GBDT, RF are attributed as **ensemble learning***, and boosting as major techniques to improve model performance while random forest adopts bagging method to improve performance.

Generally, the relationship is similar to the iteration of technologies. Therefore, RF -> GBDT -> XGBoost -> LightGBM. I shall create new question to detailedly explain each algorithms.

QUESTION 6

ENSEMBLE LEARNING

The purpose of ensemble learning is that combining multiple base-estimator's prediction result to improve single estimator's robustness and generalization.

The current ensemble learning is divided by two classes. One is strong relationship, serialization is required for result generation. Another one is no strong relationship between estimators, use parallel method to create multiple estimators. The representative of the former is Boosting, and the representative of the latter is Bagging.

Some terms can also be comprehended as ensemble learning as well such as blending, stacking. People may claim this is different. But no worries, it is the same. Just bear in mind is ok, no need to argue for a long time.

QUESTION 7

DECISION TREE AND RANDOM FOREST

Random forest and decision will be discussed together as they are complementary to each other.

Terms in tree models: root node, leaf node, decision node, leaf cutting, splitting, branch, parent node, child node...
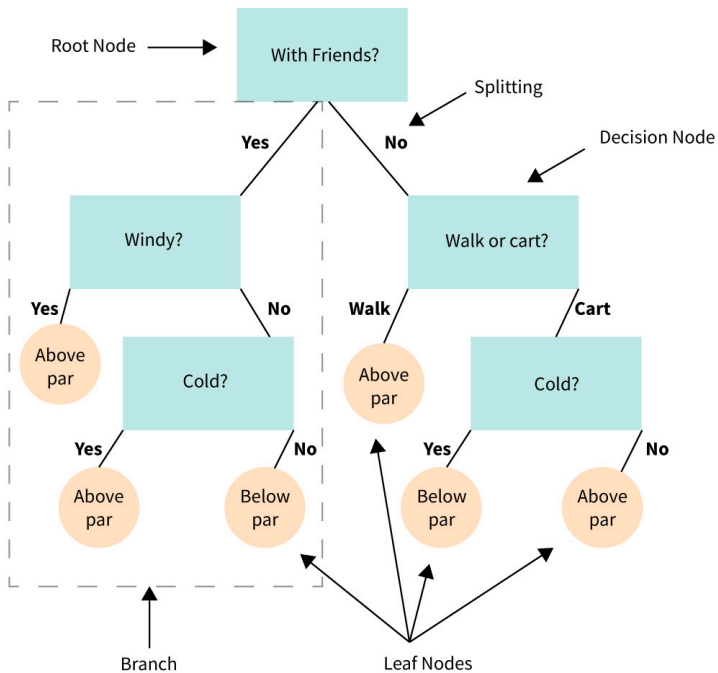
Splitting: process of splitting the root node into multiple node
Leaf node: the node can't be further split
Decision node: a child node will be split for another child node. It is called decision node
Pruning: the process of removing the child node under decision node. It is used to avoid over fitting

Classification and regression tree (CART), the major classes of decision tree model. Classification tree will use "mode" as prediction output. Regression tree will use "average" as prediction output.

The tree model usually use the following method as the evaluation of the splitting performance.
Classification) gini impurity, information gain. The ultimate goal is reducing the uncertainty of the prediction, ensure the probability.
Regression) variance. Lower variance will be selected as the decision node values. It is because lower variance means prediction has lower fluctuant prediction, or relative stable prediction, or not widely distributed results.

The splitting process will take the lowest variance as the decision value in the node.

QUESTION 8

GRADIENT BOOSTING DECISION TREE

Gradient boosting decision tree also named as multi additive regression tree. (It is useful, additive telling you it is model use summation.)

The fundamental of gradient boosting decision tree is residuals based learning (fitting the negative gradient). Assuming you have dataset (x_1, y_1), (x_2, y_2), ...(x_n, y_n). Then you use a model (linear or whatever like $F(x)$) to fit these data and minimise the loss function. After model fitted, you found its fitting effect is good but potential to advance. (e.g. residuals examples: 0.8 vs 0.9 | 1.4 vs 1.3). The easier way is creating a new model to fit the residuals of $F(x)$, which equals to $y_i - F(x_i)$.

The training dataset will be $(x_1, y_1 - F(x_1))$, $(x_2, y_2 - F(x_2))$, ...$(x_n, y_n - F(x_n))$. The target of next estimator is changed to fit the residuals of former estimator. While next estimator can predict the residuals, and you sum it up, the model is accurate. Allow me to reference a picture for easy illustration.
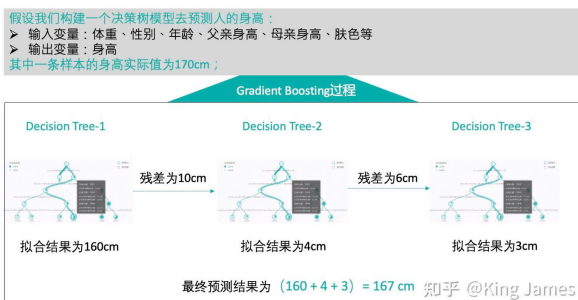
After several iterations, the model will converge finally. Thats why boosting is serialization method, and not support parallel like random forest.

Some may confuse, why GBDT use negative gradient. Loss function shall be minimum when ytrue=ypredicted . If you remember, we have regularization in the cost function. Therefore, when model adopts regularization, it is not able to use residuals for fitting, we shall calculate the gradient of loss function. In most of situation, if just fitting the residuals, the loss go smaller but higher regularization. The cost function may not be reduced. The objective function can be expressed by the gradient, uniform unit scale.

(More data, will have more precise decision node as node splitting based on the data and its variance after splitting)

QUESTION 9

XGBOOST

QUESTION 10

LIGHTGBM

QUESTION 11

COST FUNCTION, LOSS FUNCTION, OBJECTIVE FUNCTION

Loss function is usually a function defined by different metrics. It is about the difference between y_{true}, $y_{predicted}$.

$$residuals = loss = y_{true} - y_{predicted}$$

Cost function is more general. It is the summation of loss function over the training set plus regularization (or penalty)

$$cost\ function = \frac{1}{2m} \sum_{i=1}^{n} (y_{true} - y_{predicted})^2 + \frac{1}{2m} \lambda |x|$$

Objective function is the most general term for any function that optimize during training. It is customized. Like a score of specific training.

Loss function is a part of cost function which is a type of an objective function. So you can know it as any function you want to minimize or maximize called objective function.

Don't get confused in the terms definition, it is interchanged depends on the context, background... all the purpose is for gradient descent. Right?

QUESTION 12

PREPROCESSING IN MACHINE LEARNING

QUESTION 13

CROSS VALIDATION