

## Pre-concepts

### 1. What is machine learning ?

Machine learning is the study of algorithms and models that perform a specific task without using explicit instructions. A big aspect of machine learning is “pattern discovery” or “data mining”.

### 2. What is good data mining ?

Good data mining refers to the process of extracting meaningful and valuable information from large amounts of data through techniques such as machine learning. The goal of data mining is to summarize high dimensional data in such a way that we can relate it to models of interest.

### 3. What is big data ?

Big data refers to extremely large datasets that are big in both number of observations and number of variables. The datasets often require new technologies such as Hadoop, Spark, and NoSQL databases, for processing and analysis. Big data can be used to uncover patterns, trends, and associations and lead to more informed decision-making and improved outcomes.

### 4. What are three pillars of data visualization ?

They are statistics, design and language. Statistics aims to reduce the dimension of the data to a few rich variables for comparison. Design aims for effective communication with shapes, space and color for a give set of variable observations. Language makes it easy to move from statistics to design.

## Hypothesis testing

### 1. What is $\Pr(>|t|)$ ? Why the \*\*\* ?

In the output of a linear regression analysis, the term “ $\Pr(>|t|)$ ” refers to the p-value associated with each coefficient. The p-value represents the probability that the relationship between the predictor variable and the response variable is due to chance, given the null hypothesis that the true coefficient is zero.

A small p-value indicates strong evidence against the null hypothesis and supports the alternative hypothesis that the coefficient is not zero and there is a significant relationship between the predictor and response variables. Conversely, a large p-value suggests that the relationship is not statistically significant and that the predictor variable does not have a meaningful impact on the response variable.

The asterisks in the linear regression output indicate the level of statistical significance of predictor variables based on the p-value. \*\*\* suggests a p-value less than 0.01.

### 2. How do you perform hypothesis testing in linear regression ?

1). Formulate null and alternative hypotheses. Null hypothesis states that there is no relationship between the predictor and response variables; alternative hypothesis states that there is a relationship.

- 2). Choose a significance level  $\alpha$ . A significance level is the probability of rejecting the null hypothesis when it is true. (probability of making type I error)
- 3). Perform t-test statistics.

T-statistic measures how many standard errors a coefficient is away from zero, indicating whether or not it is statistically significant. A large t-statistic value indicates that the corresponding coefficient is far from zero and that the predictor variable has a strong effect on the response variable, while a small t-statistic value indicates that the effect is weak or absent.

- 4). Calculate the p-value. The p-value is the probability of observing a t-statistic as extreme as the one calculated, given that null hypothesis is true.

5). Make a decision. If the p-value is less than the significance level, we reject the null hypothesis and we consider there is statistically significant relationship between the predictor and response variables.

**3. Say there are 100 variables in the linear regression, why is it true that we cannot trust the output, in terms of statistical significance, at this point of time ?**

High dimensionality (Including too many variables in model) can increase the risk of false discovery because it leads to an increase in the number of hypothesis tests being performed. When many tests are performed, the probability of observing at least one false positive result by chance increases, even if all of the null hypothesis are true.

**4. What is (How do you calculate) false discovery proportion/rate ?**

False discovery proportion (FDP) is the ratio of number of false positives to number of tests called significant. FDP is a property of fitted models, therefore, unknown. False discovery rate (FDR) is the expectation of FDP.

**5. How does change in alpha, significance level, influence FDR ?**

As the significance level is the threshold for accepting or rejecting a null hypothesis in a hypothesis test. When the significance level is lower, the threshold of accepting a null hypothesis is higher, and fewer null hypotheses are rejected. A lower significance level reduces the risk of false positive results, and fewer false positive results will be included in the total number of significant results or rejected null hypotheses. As a result, a lower significance level (alpha) will result in a lower FDR.

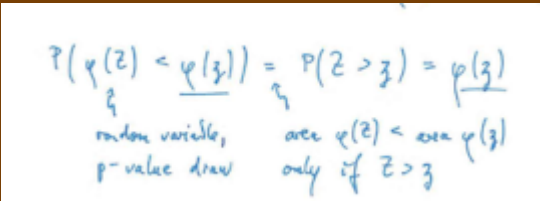
**6. Assume the significance level for a hypothesis test is  $= 0.05$ . Consider the following statement: "If the p-value is 0.001, the conclusion is to reject the null hypothesis." Is this statement correct? Explain. (2 points, 4 sentences max)**

Yes. The statement is correct. In a hypothesis test, the p-value is the probability of observing a test statistic as extreme given the null hypothesis is true. If the p-value we calculate in the test is lower than the significance level we set, it provides strong evidence that the null hypothesis is false in favor of the alternative hypothesis and we have to reject the null hypothesis.

**7. Why are p-values distributed uniformly under null hypothesis ? What does this implicate ?**

The reason for this can be inferred from the definition of  $\alpha$  as the probability of a type I error. We want the probability of rejecting a true null hypothesis to be alpha, we reject when the observed p-value  $< \alpha$ . the only way this happens for any value of  $\alpha$  is when the p-value comes from a uniform distribution.

The probability of  $F(z) < F(z^*)$  is exactly the probability of  $z > z^*$ , which is equal to  $F(z^*)$ . this distribution has the property of uniform distribution. As in uniform distribution, the probability of  $a < b$  equals to  $b$ .



$$P(\underbrace{p(z)}_{\substack{\text{random variable,} \\ \text{p-value draw}}} < \underbrace{p(z^*)}_{\substack{\text{area } p(z) < \text{area } p(z^*) \\ \text{only if } z > z^*}}) = P(z > z^*) = p(z^*)$$

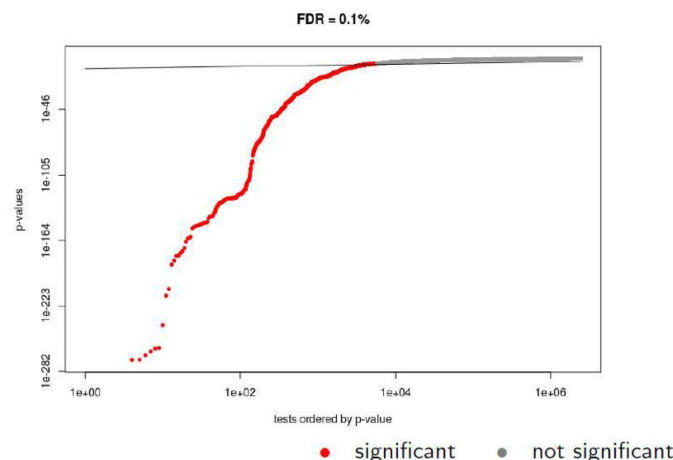
From this, we know that the probability of all p-values are going to be identical, which gives us a uniform distribution.

## 8. What is Benjamini-Hochberg (BH) procedure ? and how does it work ?

The BH procedure is a multiple hypothesis testing correction method used to control the false discovery rate at the desired level by adjusting the p-values for multiple tests.

The BH procedure works as follows :

- 1). Calculate the p-value for each hypothesis test
  - 2). Rank p-values in ascending order
  - 3). Compute the FDR threshold,  $q$ , by multiplying the desired FDR level (e.g., 0.05) by the rank of the test divided by the total number of tests (e.g.,  $q = 0.05 \cdot i/m$ , where  $i$  is the rank of the test and  $m$  is the total number of tests).
  - 4). For each test, compare its p-value with the FDR threshold. If the p-value is less than or equal to the FDR threshold, we reject the null hypothesis for the test.
- Draw the FDR-cut line with slope of  $q / \#$  [of variables] and find the max point where p-value cross this line and use that point to set your rejection region.*



# Regression

## 1. Why does it often make sense to look at the Log of a variable instead of the variable itself?

There are generally two situations where we want to look at the log of a variable. One situation is when we use log-transformation to normalize the distribution of the variable, which can be skewed or have outliers. The other situation is to interpret multiplicative change rather than additive change of a variable as log-transforming a variable makes it easier to understand percent change of the variable.

## 2. How does elasticity relate to the following regression model: $\log y = \alpha + \beta \log x + \varepsilon$ ? Explain.

Elasticity refers to the percentage change in one variable in response to a percentage change in another variable. In this regression model, we can interpret the coefficient  $\beta$  in the way of elasticity, which is defined as the % change in y in response to a 1% change in x, as follows : Elasticity =  $(\beta * 100\%) / (100\% + \beta * 100\%)$ .

## 3. What is the interaction term in regression ?

An interaction term in regression is a product of two or more independent variables to determine whether the relationship between the dependent variable and an independent variable is different for different levels of another independent variable.

## 4. What is logistic regression? How to interpret coefficients of logistic regression?

Logistic regression is a type of regression analysis to predict a binary outcome (ex. 1 as success and 0 as failure) from multiple predictor variables.

The coefficient in a logistic regression model represents the change in the log odds of the dependent variable given a one-unit change in the corresponding independent variable, holding all other independent variables constant.

For example, if we have a logistic regression :  $y = 0.34x + e$ , this suggests that having one unit increase of x multiplies odds of y by  $\exp(0.34)$ .

## 5. What is maximum likelihood estimation? What is the relationship between deviance and likelihood of regression ?

Maximum likelihood estimation is a statistical method used to estimate the parameters of a model that describe the relationship between a set of predictor variables and the outcome. The goal of the MLE method is to find a set of parameters that maximize likelihood of observing the data given the model and variables.

The deviance measures the distance between the actual data and fitted value, which we would like to make it as small as possible. Likelihood is the probability of data given parameters, which we hope to maximize. Both of them can be used to minimize the sum of squares of errors and estimate the parameters based on the previous result.

**6. What is the degree of freedom ? Can you tell me the number of variables and observations from the following R output of the logistic regression ?**

Number of observations - df, df is the number of coefficients estimated in the model.

```
> summary(spammy)...\n(Dispersion parameter for binomial family taken to be 1)\n  Null deviance: 6170.2  on 4600  degrees of freedom\nResidual deviance: 1815.8  on 4543  degrees of freedom\nAIC: 1931.8
```

We can tell that there are 4601 observations in the dataset, and there are  $4601 - 4543 = 58$  variables in the model.

**7. What are null models and standard models ?**

A null model is a baseline model that contains no predictors or independent variables. It estimates the response variable based on the mean value. It is used as a reference to compare the results of more complex models.

A standard model is the model that includes one or more predictor or independent variables.

The standard model is fit to the data to capture the relationship between the response variable and the predictors.

The standard model is compared to the null model to determine if the addition of predictors improves the fit of the model and provides additional information about the relationship between the response variable and the predictors.

**8. What is the dispersion parameter in a regression model ?**

The dispersion parameter is a quantity that characterizes the spread or variability of a probability distribution. It measures the amount of deviation in data relative to the mean value. Common examples of dispersion parameters include variance and standard deviation in the case of normal distribution.

**9. Based on 1988 census data for the 50 States in the United States, the correlation between the number of churches per State and the number of violent crimes per State was 0.85. Consider the following statement: “We can conclude that there is a causal relationship between the number of churches and the number of violent crimes committed in a city.” Is this statement correct? Explain. (3 points, 6 sentences max.)**

No. First of all, we don't have any information about the significance level of this coefficient, so it can be possible that this correlation relationship between the number of churches per State and the number of violent crimes per State is not even statistically significant. Even if the correlation is statistically significant, correlation does not mean causation. Especially in this case, a casual relationship doesn't make sense. The highly correlated relationship can be because the increase in the population correlates with both.

**10. Consider the following statement: “In a simple linear regression analysis we assume that the errors are independent and normally distributed with variance 0**

and constant mean.” Is this statement correct? Explain. (2 points, 4 sentences max)

The statement is partially correct. In a simple linear regression model, it is assumed that the error terms or residuals are independent and normally distributed. However, the assumption of zero variance(感覺這都不是個東西) and constant mean is not necessary though it is common to follow this assumption.

11. How does linear regression output change when we multiply by 100, or when we multiply x by 100, or when we multiple x and y by 100 ?

**Effects of Data Scaling**

Dependent Independent	y	cy	y
$x_1$	$\beta_1 (se_1)$	$c\beta_1 (c*se_1)$	---
$dx_1$	---	---	$\beta_1/d (se_1/d)$
$x_2$	$\beta_2 (se_2)$	$c\beta_2 (c*se_2)$	$\beta_2 (se_2)$
Intercept	$\beta_0 (se_0)$	$c\beta_0 (c*se_0)$	$\beta_0 (se_0)$
R-squared	$R^2$	$R^2$	$R^2$
SSR	SSR	$c^2*SSR$	SSR

Standard errors in parentheses

Econometrics 7

## Model Selection

1. What is  $R^2$  ? How to calculate it ? Does it mean that the larger  $R^2$  the better the model is ? What can be the limitations of using  $R^2$  to judge the goodness of fit of a model ? What is the difference between in-sample R-squared and OOS R-squared ?

R-squared measures the proportion of variation in the response variable that is explained by the predictors in a regression model.

R-squared =  $1 - (\text{Residual Sum of Squares} / \text{Total Sum of Squares})$ , where RSS is the sum of squared differences between the actual response values and predicted response values, TSS is the sum of squared difference between the actual response values and the mean of the response values.

In general, a larger R-squared indicates a better fit of the model. The limitations of using R-squared to judge the goodness of fit of a model include :

1). Does not take the number of predictors in the model into consideration: A model with many predictors can have a high R-squared even if the predictors are not significantly related to the response variable.

2). Does not penalize overfitting : A model can have a high R-squared even if it fits noise in the data, leading to overfitting.

3). Does not account for residual variability : R-squared only measures the explained variance and does not account for the residual variability. When a model has large residual variability, it means that there is still a significant proportion of variation in the response variable that is not accounted for by the predictors. Therefore, even if it has a high R-squared, it may not be a good fit for the data.

In-sample R-squared is calculated using the data fitted in the model. It represents the proportion of the outcome that could be explained by the predictors based on the samples used to fit the model. Out-of-sample r-squared is calculated using the data not used to fit the model. Instead, it represents the proportion of the outcome that could be explained by the predictors based on the data that was not used in the model. Out-of-sample r-squared provides a more realistic measure to evaluate the performance of the model as it considers the ability of generalization of the model.

## **2. What is “overfitting”? (3 points, 6 sentences max) How to detect it ?**

Overfitting is the problem that occurs when the model performs well on the training set but poorly on the new data.

Overfitting happens when the training data is too small or cannot represent the whole population, when the model contains too many unrelated variables, or when we use the same data to train the model too many times.

Here are several symptoms of overfitting :

- 1). There is a low prediction error (high prediction accuracy) on the training data but a high prediction error (low prediction accuracy) on the test data
- 2). A model that is too complex and has many parameters that don't have a significant effect on the response variable
- 3). The values of coefficients are extremely high or flip radically when excluding each additional variable.

## **3. What is the problem of using the FDR method as a model selection tool ?**

FDR method can be used to identify a subset of significant predictors from a large set of candidate predictors. However, the problem with using FDR as a model selection tool is that it assumes that the predictors are independent of each other, which is not always the case in practice. When there are highly correlated predictors, we will then get big p-values and neither variable is significant. In conclusion, in this case, the FDR method can lead to an overestimation of the number of false discoveries.

## **4. What are forward stepwise procedure, backward stepwise procedure, subset selection and problems of each ?**

Forward stepwise procedures : this method starts with a null model and adds variables one at a time based on a criterion (such as highest correlation or lowest AIC/BIC).



Backward stepwise procedures: this method starts with a model that includes all variables and removes variables one at a time, based on a criterion (such as lowest correlation or highest AIC/BIC).

Both forward and backward stepwise procedures are greedy algorithms, meaning that they find the best solution at each step that may not lead to the globally optimal solution.

Subset Selection: This is a method where all possible combinations of variables are considered, and a model is selected based on a criterion (such as minimum AIC/BIC or maximum R-squared).

Subset selection is computationally expensive and can become intractable with a large number of variables.

**5. What is regularized regression/regularization ? Why do we need to perform this type of regression ? In your own words (no math), how does a lasso regression work? (5 points, 10 sentences max) What are pros and cons of it ?**

Regularized regression/Regularization is a technique for feature selection. It adds a penalty term to the loss function that the model optimizes to shrink coefficients of some variables to be zero . As a result, we can get a sparse model with few coefficients and avoid model overfitting.

There are different types of penalty functions, lasso uses absolute value of coefficients as penalty function while ridge uses squared coefficients.

Here are steps to perform lasso regression :

1). Choose a linear regression model as the base model

2). Modify the original loss function to include penalty term :

$$\text{Loss} = \text{Mean Squared Error (MSE)} + \lambda * (\text{sum of the absolute values of coefficients})$$
$$= \sum (y_i - \hat{y}_i)^2 + \lambda \sum |\beta_j|.$$

3). Find the optimal  $\lambda$  using Cross Validation.

Set a sequence of penalties :  $\lambda_1, \dots, \lambda_t$ . For each of  $k = 1, \dots, k$  fold, fit the path  $\beta_{1, \dots, \beta_t}$  on all data of the fold  $k$  and get the fitted deviance on the data that has been left out. This will give us  $k$  draws of OOS deviance for each  $\lambda_t$ .

4). Choose the  $\lambda$  that minimizes average OOS deviance ( $\lambda_{\min}$ ) or the biggest  $\lambda$  with average OOS deviance no more than 1SD away from the minimum ( $\lambda_{1se}$ ).  $\lambda_{1se}$  will balance predicting against false discovery, while  $\lambda_{\min}$  purely focuses on predictive performances.

Pros :

1). It avoids model overfitting.

2.) It does feature selection more efficiently than stepwise selection (forward or backward selection) as it's automatic.

3). It has been found that Lasso produces the highest model prediction accuracy under the widest range of circumstances compared to stepwise selection. ("Extended Comparisons of Best Subset Selection, Forward Stepwise Selection, and the Lasso" by Hastie et al, 2017)



Cons :

- 1). It's hard to interpret the selection result of a model, for example, why does lasso select some features instead of other features?
- 2). When there are features highly correlated with each other, lasso may randomly select one of them into a model.
- 3). Sensitivity to Scale: Lasso Regression is sensitive to the scale of the features, and it is important to normalize or standardize the features before training the model. We can multiply  $\beta_j$  by  $sd(x_j)$  in the cost function to standardize, therefore,  $\beta_j$ 's penalty is now calculated per effect of 1SD change in  $x_j$ .

**6. Explain cross validation. (5 points, 10 sentences max) What are potential problems of cv ?**

Cross validation is a model evaluation technique used to assess the performance of a model by splitting the original dataset into multiple folds and training the model on different subsets while evaluating the model with the remaining part of the data. The process is repeated multiple times with different subsets of the data in order to get the most robust estimate of the mode.

Here's how k-fold cross validation works:

- 1). Split the dataset into k equally sized folds
- 2). For each iteration, use k-1 folds for training and the remaining fold for validation.
- 3). Repeat the process k times, each time using a different fold for validation.
- 4). Average the model's performance over the k iterations to obtain an estimate of its performance. (use error rate, we usually measure error rate as deviance, but alternatively, we can also use MSE, misclassification rate, or error quantiles.)

Potential problems :

- 1). it's not a perfect out-of-sample model evaluation method. As the fold of dataset that has been left out as test set will be put back to training set and used again, the performance of model measured based on the this setup cannot be seen as fully out-of-sample performance
- 2). Time-consuming and computationally expensive : cross-validation requires training and evaluating the model multiple times, which can be computationally expensive, especially for complex models and large datasets.
- 3). It's unstable. The performance of cross validation can be influenced by the random partition of the data into folds, leading to different results each time the process is repeated. This randomness can also make it difficult to compare the performance of different models.

**7. What are the information criteria ? Why do we use AICc instead of AIC ? Are AIC and CV related concepts? Explain. (5 points, 10 sentences max)**

Information criteria are used to compare and select models based on the balance between the goodness of fit and the complexity of the model.

- 1). AIC, though a in-sample information criteria, is trying to estimate OOS, as it resembles OOS deviance (= In-sample deviance +  $2df$ )

AIC (Akaike Information Criterion) = residual deviance + 2 \* df (number of non-zero variables for lasso and MLE).

2). AICc

AIC is only good for big n (number of observations) / df, when there are large number of parameters. df roughly equals to n (number of observations). AIC will lead to overfitting. In this case, we use AICc instead. AIC works like a CV curve.

AICc = residual deviance + 2 \* df \* n / n - df - 1

3). BIC

BIC roughly equals to  $-\log(p(M_b|data))$ , where  $p(M_b|data)$  is the probability that a model is true before you saw any data. BIC works more like a 1se CV curve.

BIC = residual deviance +  $\log(n) * df$

AIC and CV are related concepts in the sense that they are both methods used for model selection. Both of them can be used to approximate OOS performance of the model. AIC is a measure of the goodness of fit of a model with a penalty for the number of parameters in the model, which aims to balance the goodness of fit with the complexity of the model. It's an in-sample information criteria.

Cross-validation is a technique for estimating the performance of a model on unseen data by splitting data into validation sets and training sets. The model is trained on the training set and evaluated on the validation set. It measures the OOS performance of the model. It's always better to use CV compared to AIC.

## Treatment Effect

### 1. What is a treatment effect? (3 points, 6 sentences max)

Treatment effect refers to the effect of a treatment on outcome on a response of interest independent of any other factors that may influence the outcome.

### 2. What is active learning ? (大概率不考)

Active learning is to use existing observations to guide where you sample next. The goal of active learning is to improve the performance of a model by focusing on the most informative instances, rather than using all of the data for training.

### 3. How to control for confounders in causal inference ?

Remove from the true treatment effect of any other influences that are correlated with d, the treatment dummy, by including them in regression.

### 4. Why does lasso regression cannot help us remove confounding effects ? If we still want to use lasso, how should we remove confounders ?

Since lasso regression assigns any additional nonzero variables in the model penalty, when there is a confounding variable, it will choose to just collapse the effect of confounder into the treatment effect. To guarantee that confounding effects are removed from gamma (the treatment effect), we need to include them in the model without penalty.

We can apply two-stage lasso regression :

1. Estimate  $d(x)^{\wedge}$  with lasso regression of  $d$  on  $x$ .
2. Do a lasso of  $y$  on  $[d, d(x)^{\wedge}, x]$  including  $d(x)^{\wedge}$  unpenalized to ensure that confounder effects on  $d$  have been removed

At this point of time,  $\gamma^{\wedge}$  measures the real treatment effect.

## RDD

### 1. What is Regression Discontinuity Design (RDD)?

Regression Discontinuity Design (RDD) is a quasi-experimental research design that is used to estimate the causal effect of a treatment or intervention. It is used when assignment to the treatment group is based on a predetermined threshold (running variable), such as a score on a test or a certain age. The idea is that individuals just above and just below the threshold are similar in all other respects, except for their treatment status. By comparing outcomes for these two groups, researchers can estimate the causal effect of the treatment.

### 2. What are the limitations of Regression Discontinuity Design (RDD)?

- 1). Assignment Bias: RDD requires a clear and credible assignment mechanism for treatment and control groups, which may not always be present.
- 2). Local Average Treatment Effect: RDD is only capable of estimating the average treatment effect within the bandwidth surrounding the cutoff, but may not generalize to other populations.
- 3). Non-Random Assignment: RDD assumes that assignment to treatment is based on the score near the cutoff, but other factors could influence assignment, leading to bias.

### 3. What might be the effect of choosing a smaller bandwidth? What if we chose the maximum bandwidth in regression discontinuity design ?

In the context of Regression Discontinuity Design (RDD), bandwidth refers to the width of the interval or window used to identify the discontinuity in the treatment assignment.

Choosing a smaller bandwidth in RDD can result in more precision in the estimates of the treatment effect near the discontinuity, but may also result in a larger bias due to the limited sample size. This is because a smaller bandwidth reduces the number of observations included in the analysis, which can lead to increased variability and potential bias in the estimation of the treatment effect.

On the other hand, choosing the maximum bandwidth in RDD can result in a larger sample size and less bias in the estimation of the treatment effect, but may also result in less precision in the estimates near the discontinuity. This is because a wider bandwidth includes more observations in the analysis, which can lead to reduced variability but may also result in a dilution of the treatment effect due to the inclusion of observations that are less affected by the treatment.

In general, the optimal bandwidth size in RDD depends on the specific context and the underlying data generating process. Researchers may need to balance the trade-offs between bias and precision in the estimation of the treatment effect, and choose a bandwidth size that

provides an appropriate balance between these factors. It is also important to conduct sensitivity analyses to assess the robustness of the results to different bandwidth sizes and to test the sensitivity of the findings to other model specifications.

## Bootstrap

### 1. Use your language to explain what is bootstrap and how to do bootstrapping ?

Bootstrapping is a statistical method to estimate the distribution of statistics of interest by resampling data with replacement. We treat the sample data as population under the assumption that they can perfectly represent population. We draw samples that have the same number of observations as the original sample with replacement. With replacement means that each draw is put back in the “bucket”, so it is possible to sample the same observation multiple times.

### 2. How to get 95% CI for $\gamma^{\wedge}$ (a statistic of interest) using bootstrapping ?

- 1). Draw multiple samples with replacement from the original sample, creating a set of bootstrap samples. The size of each bootstrap sample should be equal to the size of the original sample.
- 2). Calculate the desired statistic (e.g.,  $\gamma^{\wedge}$ ) for each bootstrap sample.
- 3). Repeat steps 1). and 2), many times, usually a hundred, to obtain a large number of bootstrap estimates of  $\gamma^{\wedge}$ .
- 4). Sort the bootstrapped estimate of  $\gamma^{\wedge}$  in ascending order.
- 5). Obtain the 2.5th and 97.5th percentiles of the bootstrapped estimate of  $\gamma^{\wedge}$  to construct the 95% CI for  $\gamma^{\wedge}$ .

### 3. What are some scenarios where bootstrapping may fail ?

- 1). Extremely small sample size. When the sample size is too small, the bootstrapped estimates may not accurately reflect the true distribution of the population. In this case, the variability of the bootstrapped estimates may be too large to provide meaningful results.
- 2). When the empirical data distribution is a poor substitute for the population distribution.
- 3). When the data is high-dimensional. When data is high-dimensional, bootstrapping may fail because the number of samples required to accurately represent the population grows exponentially with the number of dimensions. In other words, as the number of dimensions increases, the data becomes increasingly sparse, making it difficult to obtain accurate estimates.

## KNN: what is the most common class around x ?

### 1. Use your own words to describe what is KNN and how to perform KNN ?

K-nearest Neighbors (KNN) is a classification algorithm based on the the idea of assigning a new observation to the class or label that is most frequent among its “K” nearest neighbors.

To perform KNN :

- 1). Choose the number of nearest neighbors “k”
- 2). Calculate the Euclidean distance between the new observations and all the training observations in the feature space.

- 3). Sort the distances in ascending order and pick the top “k” nearest neighbors.
- 4). Determine the most common class or label among the “k” neighbors.
- 5). Assign the new observation to that class or label.

## 2. What are the limitations of KNN ?

- 1). Need for proper scaling : The KNN algorithm is sensitive to the scale of features and it's necessary to scale the features to bring them to the same range. (more like note when performing KNN )
- 2). No good way to select K. The choice of the value of K is subjective and it's important to choose the right value to balance between underfitting and overfitting. Classification is sensitive to k.
- 3). Sensitive to selection of distance metric. The algorithm is highly dependent on the choice of distance metric, and the performance of the algorithm can vary significantly depending on the choice of distance metric. In addition, the algorithm may not perform well if the distance metric is not appropriate for the type of data being analyzed.
- 4). May not work well with imbalanced datasets. In such cases, the algorithm may be biased towards the majority class and may not be able to correctly classify instances of the minority class.
- 5). KNN algorithms may suffer from the curse of dimensionality. The performance of the algorithm deteriorates as the number of dimensions increases. This is because the distance between points becomes less meaningful in high-dimensional spaces, and the algorithm may not be able to effectively find the K-nearest neighbors.

## 3. What is multinomial logistic regression ? How to interpret the result of multinomial logistic regression?

Multinomial Logistic Regression (MLR) is a type of generalized linear model that is used to model multiclass classification problems, where the target variable has more than two categories. It is an extension of binary logistic regression, where instead of modeling the target variable as a binary outcome, it models the target variable as a categorical outcome with multiple classes.

Multinomial Logistic regression good website:

<https://www.mygreatlearning.com/blog/multinomial-logistic-regression/>

To interpret the results of MLR, we can look at the following aspects:

Coefficients : The coefficient of the independent variables represent the change in the log-odds of the target variable for each one-unit increase in the independent variable, holding all other variables constant. A positive coefficient indicates that an increase in the independent variable is associated with an increase in the log-odds of the target class, while a negative coefficient indicates that an increase in the independent variable is associated with a decrease in the log-odds of the target class.

For example,

If  $\exp(\beta_{mg\_A} - \beta_{mg\_B}) = 0.6$ , then Odds of class A over class B drop by 40%

If  $\exp(\beta_{mg\_A} - \beta_{mg\_B}) = 1.2$ , then Odds of class A over class B increase by 20%

## K-Means

### 1. What is the difference between supervised and unsupervised learning?

Supervised learning involves training a machine learning model on a labeled dataset, where the labels are provided by a human expert. The goal of supervised learning is to predict the correct output for new, unseen input data based on the patterns and relationships learned from the labeled data.

Unsupervised learning, on the other hand, involves training a model on an unlabeled dataset, where the algorithm has to discover the underlying patterns and relationships in the data without any guidance from a human expert. The goal of unsupervised learning is to uncover hidden structures, clusters, or associations in the data.

In summary, the key difference between supervised and unsupervised learning is that in supervised learning, the model is trained on labeled data, while in unsupervised learning, the model is trained on unlabeled data, and has to discover patterns and relationships on its own.

### 2. Use your own words to describe what K-means algorithm is and how to perform K-means?

K-means is a clustering algorithm used in unsupervised machine learning to group similar data points into clusters. The algorithm works by iteratively assigning each data point to the cluster whose mean is closest to the data point.

The following is a step-by-step description of how to perform K-means:

1. Choose the number of clusters (K) that you want to create.
2. Initialize the centroids of the K clusters randomly or based on some other criterion.
3. Assign each data point to the cluster whose centroid is closest to it. The distance metric used can be Euclidean, Manhattan, or any other distance metric.
4. Recalculate the centroids of the K clusters by taking the mean of all the data points assigned to each cluster.
5. Repeat steps 3 and 4 until the assignments no longer change or a maximum number of iterations is reached.
6. Once the algorithm converges, we have K clusters, each containing similar data points.

In each iteration, we calculate the distance between each data point and each centroid, and assign the data point to the cluster with the closest centroid. This process is repeated until the assignments no longer change or a maximum number of iterations is reached.

### 3. What are the differences between KNN and K-means algorithms (in terms of operation steps) ?

KNN is a supervised learning algorithm used for classification and regression tasks, while K-means is an unsupervised learning algorithm used for clustering tasks. KNN finds the K-nearest neighbors of a given data point to predict its class or value, while K-means divides a dataset into K clusters based on similarities between the data points.

Here are the main differences in the steps of KNN and K-Means clustering:

1. Purpose: KNN is a classification algorithm that is used for supervised learning, while K-Means is an unsupervised clustering algorithm used for unsupervised learning.
2. Data Preparation: In KNN, the data is split into training and testing sets, where the training set is used to fit the model and the testing set is used to evaluate its performance. In K-Means, the data is not split into training and testing sets, but is preprocessed by scaling and normalizing the features to ensure equal weight is given to all features.
3. Feature Selection: In KNN, feature selection is an important step to choose the most relevant features for classification. In K-Means, all features are considered for clustering.
4. Distance Metric: In KNN, distance metrics such as Euclidean distance or Manhattan distance are used to calculate the distance between observations. In K-Means, the distance metric used is typically the Euclidean distance between observations and cluster centroids.
5. Number of Clusters: In KNN, the number of classes is determined by the number of unique labels in the training data. In K-Means, the number of clusters is determined by the user, and an appropriate number of clusters is usually determined through trial and error.
6. Cluster Assignment: In KNN, the class label of a test observation is determined by finding the K closest training observations based on the distance metric, and then selecting the most frequent class label among those K training observations. In K-Means, observations are assigned to clusters based on their proximity to the cluster centroids, which are calculated by taking the mean of all observations assigned to that cluster.

#### **4. What are the limitations of the K-means algorithm?**

- 1) Need for proper scaling: K-means is sensitive to scaling, which can affect the distances between data points and the centroids.
- 2) Difficult to determine the initial centroids and number of clusters: Choosing different initial centroids and number of clusters will affect the quality of the clusters, leading to poor performance in clustering.

#### **5. How to choose the number of clusters for the K-means algorithm?**

Choosing the number of clusters (K) in K-means is an important step in the clustering process, and there are several methods that can be used to determine the optimal number of clusters:

- 1) Elbow method: Calculate the deviance, which is the sum of squared distances between the data points and the assigned centroids, and plot the result. The optimal k will be the point of elbow where the reduction of deviance begins to level off.



- 2) IC metrics: Calculate the AIC and BIC for each K and select the K which has the lowest IC values. BIC tends to be more reliable than AIC, but it is better to decide K based on the descriptive intuition.

## 6. How do I examine how good my K-means model is?

1. Within-cluster sum of squares (WSS): WSS measures the sum of squared distances between the data points and their assigned cluster centers. A lower WSS indicates that the clusters are more tightly packed, which is a sign of a better K-means model.
2. Silhouette score: Silhouette score is a measure of how similar a data point is to its assigned cluster compared to other clusters. A higher Silhouette score indicates that the data points are appropriately assigned to their clusters, and hence the K-means model is better.
3. % change of data points added and % change of deviance increase: If the percentage change of deviance increased is greater than the percentage change of data points added, meaning that the model performed poorly.

## PCA

### 1. What is principal component analysis (PCA)? How does it work step by step ?

Principal Component Analysis (PCA) is a widely used technique in data analysis to reduce the dimensionality of high-dimensional datasets. It is a statistical method that transforms a large number of potentially correlated variables into a smaller number of uncorrelated variables, called principal components (PCs).

The idea behind PCA is to identify the directions in the data that explain the most variation, and then project the data onto these directions. The first principal component is the direction that captures the most variation in the data. The second principal component is the direction that captures the most variation that is orthogonal to the first principal component, and so on. Each principal component is a linear combination of the original variables, and they are sorted in order of the amount of variation they explain.

The transformation of the data into the principal components allows for a reduction in the number of variables needed to describe the dataset, while still retaining as much of the original variation as possible. This can be useful for data visualization, clustering, and classification.

To compute the principal components, PCA uses a mathematical technique called singular value decomposition (SVD), which decomposes the data matrix into orthogonal matrices of left and right singular vectors, and a diagonal matrix of singular values. The singular values represent the amount of variation captured by each principal component.

**Jorn**可能會比較喜歡的版本:

PCA is a technique used to transform data into unknown factors, during which process we will not lose any information. Geometrically speaking, PCA is rotating the input vectors so that for the new set of vectors, the first component will align with the direction that contains highest variances and the second component aligns with the second highest variances and at the same time is perpendicular to the first component, and so on.

## **2. Please highlight limitations of PCA.**

Although PCA is a useful method to reduce the number of variables, there are several limitations to using this approach. Here are some below:

- 1) Normality assumption: PCA assumes that the relationships between variables are linear and normally distributed. PCA may not be effective for data sets which are not linear or normally distributed. In addition, these two assumptions are difficult to satisfy in practice.
- 2) Outliers matter: PCA is sensitive to outliers and differences in scale between variables. Outliers and different scales between variables will affect or even distort the result of PCA. Data preprocessing is the most important step before using PCA.
- 3) Hard to interpret: It is difficult to understand and interpret the resulting principal components as we do not know the underlying structure of the data. In addition, we may lose some information as we reduce the number of variables. The resulting principal components may not capture all of the important information in the original data set.

## **3. How to decide the number of PCs to use in PCA?**

Deciding on the number of principal components (PCs) to use in PCA involves finding the right balance between the amount of variance retained in the reduced dataset and the computational complexity of the analysis.

- 1) Proportion of variance: This method involves selecting the minimum number of PCs that account for a certain proportion of the total variance. For example, selecting the number of PCs that explain 90% of the total variance.
- 2) Cross-validation: In some cases, the number of PCs can be determined using cross-validation. A model is trained on a subset of the data and validated on the remaining data using different numbers of PCs. The optimal number of PCs is the one that produces the best performance on the validation data.

## **4. What are differences between factor models (which PCA is used for) and variable selection in reducing dimensionality of data ? What are the pros and cons of both ? When to use which one ?**

Factor models and variable selection are two common techniques used to reduce the dimensionality of data. While they share some similarities, they differ in their approach and purpose.

Factor models, such as principal component analysis (PCA), aim to identify the underlying factors or sources of variation that explain the correlations between the observed variables. These factors are typically unobservable, and are represented as linear combinations of the observed variables. PCA generates a new set of orthogonal variables, called principal components, that capture the maximum amount of variation in the data. Factor models assume that the observed variables are all related to the same underlying factors, and they aim to identify those factors that are most important for explaining the data.

Variable selection, on the other hand, aims to identify a subset of the observed variables that are most relevant for explaining the response variable or outcome of interest. This approach is often used in machine learning and statistical modeling, where the goal is to build a predictive model with a smaller set of variables. Variable selection methods include stepwise regression, Lasso, and Ridge regression, among others. These methods can be used to select a subset of variables that are most predictive of the outcome, and to reduce the risk of overfitting by eliminating irrelevant or redundant variables.

The choice between factor models and variable selection depends on the specific goals of the analysis. If the goal is to identify the underlying factors or sources of variation that explain the correlations between the observed variables, then factor models such as PCA may be more appropriate. If the goal is to build a predictive model with a smaller set of variables, then variable selection methods may be more appropriate. In some cases, both approaches may be used in combination to achieve a better understanding of the data.

The pros of factor models are that they can identify underlying factors that are not immediately visible from the observed variables, and can provide insight into the structure of the data. The cons of factor models are that they can be difficult to interpret, and the resulting principal components may not be directly meaningful.

The pros of variable selection are that they can reduce the dimensionality of the data and improve the interpretability of the model, and can lead to better predictive performance. The cons of variable selection are that it can be sensitive to noise and collinearity in the data, and can lead to overfitting if the wrong variables are selected.

## **5. What is loading/rotation in the factor model ? How to interpret the size of loading or rotation ?**

A factor model is a statistical method used to identify and analyze underlying latent factors that can explain the patterns of covariance among a set of observed variables. In other words, a factor model seeks to identify the common factors that influence the observed variables, and **to** quantify how much each variable is influenced by each factor.

In a factor model, the loading of a variable on a factor represents the extent to which the variable is related to that factor. Loadings can be interpreted as weights that indicate the degree of association between the variable and the factor. A high loading means that the variable is strongly related to the factor, while a low loading means that the variable is weakly related to the factor.

## **6. What is PCR and PLS ? What's the difference ?**

PCR (Principal Component Regression) and PLS (Partial Least Squares) are two closely related regression methods that are commonly used for modeling high-dimensional data.

PCR is a regression method that combines the principles of PCA and linear regression. In PCR, the data is first transformed using PCA to reduce the dimensionality of the data and extract the most important principal components. Then, linear regression is applied to the transformed data to model the relationship between the predictors and the response variable. PCR can be a useful method when there are a large number of predictors and the number of observations is relatively small, as it can help to reduce the risk of overfitting.

Here are steps of PCR :

1. Perform PCA: Apply PCA to the predictor variables to extract the most important principal components that explain the majority of the variation in the data. Typically, only the first few principal components are retained.
2. Calculate the scores: Calculate the scores for each observation on the retained principal components. These scores represent the new variables that will be used in the regression analysis.
3. Perform linear regression: Use the scores on the retained principal components as the predictors in a linear regression model, with the response variable as the outcome. Estimate the regression coefficients using ordinary least squares (OLS) or another appropriate method.

PLS is a regression method that is similar to PCR, but instead of using PCA to transform the data, it uses a different approach called "partial least squares" to extract the relevant information from the predictors. PLS works by identifying a set of latent variables (also called "latent factors" or "latent variables") that capture the most important information in the predictor variables, and then using these latent variables to model the relationship between the predictors and the response variable.

1. Estimate the weights for the predictor variables: Estimate the weights for each predictor variable by computing the correlation between each predictor variable and the response variable.
2. Calculate the first set of latent variables: Use the weights to calculate the first set of latent variables, which are linear combinations of the predictor variables that are most strongly associated with the response variable.
3. Repeat steps 1 and 2 to calculate additional sets of latent variables: Calculate additional sets of latent variables by repeating steps 3 and 4, but with the predictor variables adjusted to account for the previously calculated latent variables.
4. Use the latent variables to predict the response variable: Use the calculated latent variables as the predictors in a linear regression model, with the response variable as the outcome.

The main difference between PCR and PLS is that PCR uses PCA to transform the predictors, while PLS uses partial least squares to extract the relevant information from the predictors. In general, PLS tends to be more robust than PCR when there are a large number of predictors and the correlations among the predictors are high. However, the choice between PCR and PLS ultimately depends on the specific characteristics of the data and the research question at hand.

**7. Under PCA, when we exhaust all the variances with transformed vectors, is the number of PCs going to be identical to the number of original variables?**

Yes, the number of PCs is for most of the time going to be the same as the number of variables. Reason is that, when doing PCA, PCs have to account for the entire vector space that the original variables can span, which means that, if there is **no perfect collinearity** between the input variables, we will need the exact same number of PCs to account for all the dimensions.

# Decision Tree

## 1. What is a decision tree? How does it work step by step ?

A decision tree is a supervised machine learning algorithm used for classification and regression analysis. It is a tree-like model where each node represents a feature or attribute, each branch represents a decision rule, and each leaf node represents a decision or outcome.

The algorithm works by recursively partitioning the data based on the values of the input features until a decision can be made. At each node, the algorithm selects the feature that best separates the data based on a certain criterion, such as information gain. The feature with the highest information gain is chosen as the splitting criterion.

Once a feature is selected, the data is partitioned into subsets based on the values of that feature. The process is then repeated for each subset, recursively creating more branches and nodes until a stopping criterion is met. The stopping criterion could be a maximum depth of the tree, a minimum number of samples per leaf node, or a minimum decrease in impurity.

After the tree is built, new data can be classified by following the decision rules in the tree from the root node to a leaf node. Each decision rule corresponds to a path through the tree, where the value of the selected feature determines which branch to follow. The leaf node reached by following the decision rules corresponds to the predicted class or outcome for the new data.

## 2. How is the classification tree/ regression tree grown? What is the difference between these two tree algorithms?

Classification tree: output 是 class/label

- 1) The tree starts with a single node that contains all the observations in the data set.
- 2) The algorithm selects a feature and a split point that divides the data into two or more subsets that are similar as possible with respect to the target variable (lowest entropy). We can use misclassification error or Gini index to determine the split points which maximize the separation between classes.
- 3) Once a split point has been selected, the data is divided into two or more subsets and the process will continue recursively until some stopping criterion is met. For example, when a subset contains a minimum number of observations in a leaf node or a maximum depth of the tree.
- 4) As the tree grows, each internal node represents a decision based on a feature and each leaf node represents a class label.

Regression tree: output 是 numerical value

- 1) The tree starts with a single node that contains all the observations in the data set.
- 2) The algorithm selects a feature and a split point that divides the data into two or more subsets that are similar as possible with respect to the target variable. We can use mean squared error or mean absolute error to determine the split points which minimize the reduction in the variance of the target variable.
- 3) Once a split point has been selected, the data is divided into two or more subsets and the process will continue recursively until some stopping criterion is met. For example, when a

subset contains a minimum number of observations in a leaf node or a maximum depth of the tree.

- 4) As the tree grows, each internal node represents a decision based on a feature and each leaf node represents a class label.

In sum, the main differences between Classification Tree and Regression Tree are the objective, output, and splitting criteria. For the Classification Tree, it is used to predict categorical variables, produce class labels as output, and split the data by maximizing the separation between classes. For the Regression Tree, it is used to predict numerical variables, produce numerical values as output, and split the data by minimizing the variance of the predicted values.

### 3. How to prune the tree?

Pruning is a technique used in decision tree algorithms to prevent overfitting and improve the accuracy of the model. Pruning involves removing branches of the decision tree that do not contribute much to the accuracy of the model. There are two main types of pruning methods:

- 1) Pre-pruning: This involves setting a stopping criterion before the tree is fully grown. The tree is pruned if the stopping criterion is met. Examples of stopping criteria include limiting the maximum depth of the tree, requiring a minimum number of samples to be present in a leaf node, or setting a minimum value for the improvement in accuracy from splitting a node.
- 2) Post-pruning: This involves growing the tree to its maximum size and then removing branches that do not improve accuracy or contribute least to the deviance reduction. The most common method for post-pruning is called Reduced Error Pruning. It involves starting at the leaves of the tree and working upwards, removing branches that result in the smallest decrease in accuracy on a validation set. This process continues until no further improvement in accuracy can be achieved.

In both pre-pruning and post-pruning, the optimal parameters for pruning need to be determined through a validation set or cross-validation as pruning will yield lots of candidate trees.

### 4. What are the advantages of tree models compared with other regression models and limitations of the decision tree algorithm?

**Advantages :** (1/2/6属于考试范围内容)

- 1) **Non-linearity**: Tree-based models can capture non-linear relationships between variables, which can be difficult to model with linear regression or other linear models.
- 2) **Interpretability**: Decision trees can be easily visualized and interpreted, making it easy to understand the decision-making process of the model. This can be especially useful in situations where the model's predictions need to be explained to stakeholders.
- 3) **Robustness**: Tree-based models are less sensitive to outliers and can handle missing values well compared to other regression models.

- 4) Feature selection: Random forests and gradient boosting can automatically perform feature selection, identifying the most important variables in the data for making predictions. This can help simplify the model and improve its accuracy.
- 5) Ensemble learning: Random forests and gradient boosting are based on ensemble learning, which combines the predictions of multiple individual models to improve the overall performance of the model. This can result in more accurate predictions than a single model.
- 6) Interaction Detection : Nonconstant variance will not be the problem anymore as trees are able to automatically detect interactions between variables.

#### Limitations :

- 1) Overfitting: Decision trees are prone to overfitting, meaning they can become too complex and capture noise in the data rather than the underlying patterns. This can lead to poor generalization to new data and reduced accuracy.

(Solutions :

Split to lower deviance until leaves hit minimum size

-> create a set of candidates trees by pruning back from this

-> Choose the best among those trees by cross validation)

- 2) Instability: Tree-based models can be sensitive to small variations in the data, leading to instability in the model. This can result in different trees being generated for different subsets of the data or for different runs of the algorithm, which can make the model difficult to interpret and reproduce.
- 3) Biased outcomes: Decision trees can be biased if the training data is imbalanced or if certain features are more important than others in determining the outcome.

## Random Forests

### 1. What is the random forest algorithm? How does it work step by step?

Random forest is a learning algorithm used in machine learning for both classification and regression problems. The algorithm creates multiple decision trees and combines their predictions to make more accurate and robust predictions. Here is how the random forest algorithm works:

- 1) Data Sampling: Random Forest randomly selects a subset of the data from the original dataset. This is done by sampling with replacement, which is also called bootstrapping. The size of the subset is typically around two-thirds of the original dataset.
- 2) Feature Selection: At each node of the decision tree, Random Forest selects a random subset of features to consider for splitting. This helps to prevent overfitting and improve the generalization of the model.
- 3) Building Decision Trees: Random Forest builds a set of decision trees based on the bootstrapped samples and feature subsets. Each tree is trained on a different subset of the data and a different set of features.



- 4) **Splitting Nodes:** At each node of the decision tree, Random Forest selects the best split based on a criterion such as the Gini impurity or the information gain. The split is made to maximize the homogeneity of the samples in each resulting node.
- 5) **Predicting Output:** Once all the decision trees have been built, Random Forest combines their predictions by taking the majority vote (for classification problems) or the average (for regression problems). This ensemble method helps to reduce the variance of the model and improve its accuracy.
- 6) **Tuning Hyperparameters:** Finally, Random Forest can be fine-tuned by adjusting hyperparameters such as the number of trees, the depth of the trees, and the size of the feature subsets. This can be done using cross-validation or grid search to find the optimal set of hyperparameters that minimize the error of the model.

## 2. What are the key techniques the random forest uses to avoid overfitting?

Random forest uses two key techniques to reduce overfitting and increase accuracy:

- 1) **Bootstrapping:** Random forest creates multiple decision trees using bootstrapping, a statistical technique that involves resampling data with replacement. By building decision trees on different subsets of data, random forest avoids repeatedly including outliers into trees, therefore, reduces the risk of overfitting and improves the model's ability to generalize to new data.

Repeatedly re-sample the data with replacement to get a jittered dataset of observations

-> fit a CART tree for each resample

-> take the average prediction from the forest of trees

- 2) **Feature Randomization:** Random forest also randomly selects a subset of features for each decision tree. This limits the amount of information that a single tree can learn, therefore, trees grown in this fashion are less deep, with larger bias, in particular for complex dataset. Trees built using a subsample tend to be less correlated to each other, as they are often built on completely different subsets of features. This technique reduces the correlation between decision trees and ensures that each tree makes predictions based on different subsets of data, further reducing overfitting and improving accuracy.

## 3. Please explain why a Random Forest usually outperforms regular regression methods (such as linear regression, logistic regression, and lasso regression).

- 1) A Random Forest can capture nonlinear relationships between features and the target variable, which regular regression methods often cannot. It can capture complex interactions between features, which will help improve the accuracy of the predictions.
- 2) A Random Forest is less affected by outliers because they use multiple decision trees to make predictions, while regular regression methods are often sensitive to outliers. The outliers may only affect some of the trees but are less likely to affect all the trees in the forest, which reduces the impact of outliers.

- 3) A Random forest is less likely to overfit because it uses multiple decision trees, which can reduce the impact of the random variations in the data. Also, it can be regularized by controlling the number of trees, the depth of the trees, and the number of observations in each leaf node. However, the regular regression methods may overfit the data if the model is too complex or contains too many features.

#### **4. What is the difference between the decision tree and random forests algorithms?**

The key differences between decision trees and random forests are:

- 1) Overfitting: Decision trees are prone to overfitting, meaning they can capture noise in the data and fail to generalize well to new data. Random forest, on the other hand, reduces overfitting by using multiple trees, each trained on a different subset of the data and features.
- 2) Accuracy: Random forests generally have higher accuracy than decision trees, especially for complex datasets with many features.
- 3) Interpretability: Decision trees are more interpretable than random forests because they create a single tree-like structure that can be easily visualized and understood. Random forests, however, are more complex because they use multiple trees, and their predictions are based on an aggregate of many trees.

In summary, while decision trees are simpler and more interpretable, random forests are more accurate and less prone to overfitting. The choice between the two algorithms depends on the specific problem at hand, including the complexity of the data, the number of features, and the desired level of interpretability.

#### **5. What is boosting ? How to boost the tree algorithm? What are the pros and cons of boosting ?**

Boosting is a machine learning technique that combines a learning algorithm in series to achieve a strong learner from many sequentially connected weak learners.

It can be applied to many algorithms. Take the decision tree as an example of a weak learner, each tree attempts to minimize the errors of the previous tree. Adding many trees in series and each focusing on the errors from the previous one makes boosting highly efficient and accurate. Every time a new tree is added, it fits on the residuals of the previous tree. Unlike bagging, boosting does not involve bootstrapping.

Here are the steps to boost a tree-based algorithm using gradient boosting:

- 1). Train a base tree model on the data. This is typically a decision tree, but other tree-based models can be used as well. The base model is a weak learner, meaning that it has low accuracy and is prone to overfitting.
- 2). Compute Residuals: After training the base model, the residuals are computed by subtracting the predicted values from the actual values. The residuals represent the errors of the base model.
- 3). Train a New Tree Model : A new tree model is trained on the residuals, with the goal of correcting the errors of the base model. The new tree model is added to the ensemble.

4). Combine Models : The predicted values of all the tree models in the ensemble are combined to make the final prediction repeated multiple times, with each iteration producing a new tree model that corrects the errors of the previous models. The number of iterations is a hyperparameter that can be tuned to optimize the performance of the model.

Steps 2) - 4) are repeated multiple times, with each iteration producing a new tree model that corrects the errors of the previous models. The number of iterations is a hyperparameter that can be tuned to optimize the performance of the model.

One key difference between random forest and gradient boosting decision trees is the number of trees used in the model. Increasing the number of trees in gradient boosting decision trees will cause overfitting, so it's important to stop adding trees at some point. Increasing the number of trees in random forests does not cause overfitting.

#### Pros of Boosting:

- Boosting can lead to significant improvements in accuracy compared to using a single model, especially for complex datasets with a high degree of noise.
- **Boosting can handle both classification and regression tasks.**
- **Slightly more accurate predictions compared to random forests**
- **Can handle mixed type of features and no preprocessing is needed**
- Boosting is a flexible technique that can be applied to a wide range of models, including decision trees, neural networks, and support vector machines.

#### Cons of Boosting:

- Boosting can be computationally intensive and may take longer to train than other machine learning models.
- Boosting can be prone to overfitting, especially if the number of iterations or trees is too high. It is important to monitor the performance of the model during training and use techniques such as early stopping to prevent overfitting.
- Boosting may not perform well on small datasets, as it requires a sufficient amount of data to learn the underlying patterns.
- **Sensitive to outliers**
- **Hard to interpret (random forest is more straightforward)**
- **Require careful tuning of parameters**

#### **6. What is the importance of variables in a random forest? How to estimate the importance of variables?**

Variables that have a high importance are those that have a strong relationship with the target variable, while variables with low importance are those that have little or no relationship with the target variable. Estimating the importance of variables in a random forest can be done using one of several methods.

The main idea of estimation of importance is we randomly make one variable noise to eliminate the effect of it on the response variable and we calculate the accuracy lost/% decrease in

deviance under this situation. The larger the accuracy lost/% decrease in deviance the higher importance that variable has.

Here are three common approaches:

1. Mean decrease impurity: This method calculates the importance of each variable by measuring the reduction in the impurity of the split caused by that variable across all trees in the forest. Variables that consistently lead to a large reduction in impurity are considered more important.
2. Mean decrease accuracy: This method calculates the importance of each variable by measuring the reduction in the accuracy of the model when that variable is removed from the dataset across all trees in the forest. Variables that lead to a large reduction in accuracy are considered more important.
3. Permutation importance: This method calculates the importance of each variable by randomly permuting the values of that variable in the dataset and measuring the resulting reduction in accuracy of the model. Variables that lead to a large reduction in accuracy when permuted are considered more important.

## Neural Network

### \*Prep work for NN model training:

Normalize the data before training! Why?

- 1) Reducing the impact of scale: When different input features have vastly different scales, the larger features can dominate the training process, leading to suboptimal weights and bias values. By normalizing the input data to have a similar scale, the impact of any one feature is reduced, and the network can learn more efficiently and effectively.
- 2) Improving convergence speed: Normalizing the input data can also help to speed up the convergence of the training process, particularly when using gradient-based optimization algorithms. **Not normalizing may lead to useless results or to a very difficult process (most of the time the algorithm will not converge before the number of maximum iterations allowed)**. Normalizing the data can reduce the overall range of the activation functions used in the network, which can help to prevent the gradients from becoming too small or too large, leading to more stable and efficient updates of the weights and biases.
- 3) Making the model more robust: Normalizing the input data can also help to improve the robustness of the network to changes in the input distribution, such as shifts or scaling. This is because normalizing the data can help to remove any biases or trends in the data, making the network more adaptable to new or unseen data.

1. **What is a neural network? What elements are included in one neural network ? Describe the training process of a neural network.**

Neural network is a type of machine learning model that is inspired by the structure and function of the human brain. Neural network works by processing input data through multiple layers of

interconnected neurons, with each neuron applying an activation function to the input it receives.

The input data is fed into the network through the input layer, which consists of neurons that represent the input features. The hidden layers of the neural network, which are located between the input and output layers, are where the computation takes place. Each hidden layer consists of a set of neurons that are connected to the neurons in the previous and next layers. Each neuron in the hidden layers of the neural network applies an activation function to the input it receives. The activation function helps to introduce nonlinearity into the model, which enables it to learn complex patterns and relationships in the data. Common activation functions include the sigmoid function and the rectified linear unit (ReLU) function. The output layer is the final layer of the neural network, which produces the output of the model. The weights and biases of the neural network are the parameters that are learned during the training process.

The weights determine the strength and direction of the connections between the neurons, and the biases help to shift the output of the activation function in a particular direction. During the training process, the weights and biases are updated based on the error between the predicted output and the actual output, using techniques such as backpropagation and gradient descent.

## **2. What is activation function ? Describe the difference between sigmoid function and ReLU function (第二小題感覺太難了應該不會考)**

Activation functions are mathematical functions that are used to introduce nonlinearity into a neural network, allowing it to learn complex patterns and relationships in the data. The output of each neuron in a neural network is transformed by the activation function before being passed on to the next layer.

- Sigmoid function: The sigmoid function is a smooth, S-shaped curve that maps any input to a value between 0 and 1. It is commonly used in the output layer of a neural network for binary classification problems, where the output represents the probability of the positive class. However, the sigmoid function can suffer from the problem of vanishing gradients, where the gradient approaches zero as the input becomes very large or very small.
- ReLU (Rectified Linear Unit): ReLU is a simple and popular activation function that returns the input if it is positive, and 0 otherwise. It is computationally efficient and can help accelerate training by avoiding the problem of vanishing gradients. ReLU is commonly used in the hidden layers of a neural network.

## **3. What is backpropagation ? How does it work ?**

Back propagation is a supervised learning algorithm used in neural networks to train the model by adjusting the weights and biases of the network to minimize the difference between the predicted output and the actual output. The algorithm works by propagating the error backwards

through the network, from the output layer to the input layer, and updating the weights and biases accordingly.

The backpropagation algorithm consists of the following steps:

1. Forward pass: The input data is fed into the network through the input layer, and the output of each neuron is calculated based on the weights and biases of the network. The output of the network is compared to the actual output, and the error is calculated.
2. Backward pass: The error is propagated backwards through the network, from the output layer to the input layer. The error is used to calculate the gradient of the loss function with respect to the weights and biases of the network.
3. Weight update: The weights and biases of the network are updated using the gradient descent algorithm, which adjusts the weights and biases in the direction that minimizes the loss function.
4. Repeat: Steps 1-3 are repeated for each training example, and for multiple epochs until the loss function converges or reaches a desired level of accuracy.

During the backward pass, the error is propagated backwards through the network using the chain rule of calculus. The error at each neuron is split among the neurons that contributed to its output in the forward pass, according to their relative contributions. This process continues all the way back to the input layer, where the error is used to update the weights and biases of the network.

#### **4. What is Gradient Descent Algorithm? Describe the types of Gradient Descent ? How does each work ?**

Gradient Descent is an optimization algorithm used in machine learning to minimize the loss function of a model by iteratively adjusting the parameters or weights of the model. The basic idea of gradient descent is to calculate the gradient of the loss function with respect to each weight, and update the weights in the opposite direction of the gradient, in order to move towards the minimum of the loss function.

There are three main types of gradient descent:

1. Batch Gradient Descent: In batch gradient descent, the entire training dataset is used to compute the gradient of the cost function with respect to the weights. The weights are then updated after each epoch, based on the average of the gradients computed over the entire dataset. Batch gradient descent is computationally efficient for small datasets, but can be slow for large datasets, since it requires a complete pass over the entire dataset for each weight update.
2. Stochastic Gradient Descent: In stochastic gradient descent, the weights are updated after each training example. The gradient is computed on a single training example at a time, and the weights are updated based on the gradient of the cost function with respect to that example. Stochastic gradient descent can be much faster than batch gradient descent for large datasets, but can be noisy and can converge to a suboptimal solution due to the high variance of the gradient estimates.

3. Mini-batch Gradient Descent: Mini-batch gradient descent is a compromise between batch gradient descent and stochastic gradient descent. In mini-batch gradient descent, the dataset is divided into small batches, and the gradient is computed on each batch. The weights are then updated based on the average of the gradients computed over the entire batch. Mini-batch gradient descent can be more stable than stochastic gradient descent, since the gradient estimates are less noisy, and can be more efficient than batch gradient descent for large datasets, since it can take advantage of vectorized operations on GPUs.

#### Difference between batch sizes:

- In general, training with larger batch sizes can lead to faster convergence and improved generalization performance, but it can also require more computational resources and longer training times per epoch. This is because larger batch sizes require more memory to store the intermediate activations and gradients during backpropagation, and may require parallel processing to efficiently perform the matrix operations required for forward and backward passes.
- On the other hand, smaller batch sizes may converge more slowly but can be trained more quickly per epoch, since they require less memory and can be more easily parallelized. Additionally, smaller batch sizes can sometimes lead to better generalization performance by introducing more stochasticity and preventing the network from overfitting to the training set.

The choice of gradient descent algorithm depends on the size of the dataset, the complexity of the model, and the available computational resources. Batch gradient descent can be used when the dataset is small, and stochastic gradient descent or mini-batch gradient descent can be used for larger datasets. Stochastic gradient descent can be used to escape from local minima, while mini-batch gradient descent can be used to achieve a balance between computational efficiency and stability.

### **5. How to identify whether a neural network is overfit or not ? What are some regularization methods for neural networks ?**

#### How to identify ?

When the weights in a neural network are very large, the neural network may overfit.

#### How to avoid overfitting ?

- Most used:
  - Weight Regularization (Weight Decay): Penalize the model during training based on the magnitude of the weights.
  - Dropout: Probabilistically remove inputs during training.

**The basic idea is to calculate loss function each time leaving out one neuron (normally x% of neurons), to make the activation function of that neuron/x% of neurons to be zero. This avoids any neuron/neurons being trained on some outliers.**

During each training iteration, each neuron has a probability of being dropped out, which is typically set to a value between 0.2 and 0.5. The effect of dropout is to make the network more



robust and less sensitive to the specific weights of individual neurons. By randomly dropping out neurons during training, the network is forced to learn redundant representations of the data, which helps to prevent overfitting.

During the testing phase, when the network is making predictions on new, unseen data, dropout is turned off and all neurons are used. This means that the network is making predictions using the full set of neurons, which should result in better performance than during training.

When we are implementing dropout, we set the output of some stochastically selected neurons to 0, so the remaining neurons have to take up the responsibility of explaining the deviance, which will make the model more generalized, because no neurons in this sense will be responsible for explaining some specific trait.

- Less used:
  - Activity Regularization: Penalize the model during training based on the magnitude of the activations.
  - Weight Constraint: Constrain the magnitude of weights to be within a range or below a limit.
  - Noise: Add statistical noise to inputs during training.

The model will be more resilient to errors because we demand the model to make predictions on the same set of outputs even if our data is being modified (contains additional noises).

- Early Stopping: Monitor model performance on a validation set and stop training when performance degrades.

## **6. What are types of neural networks and please describe concepts of each.**

- Single-layer perceptron
  - No hidden layer
- Multi-layer perceptron (MLP)
- Convolutional Neural Network (CNN)
  - What is the role **of convolutional layers in convolutional neural networks, and how do they work?**

Convolutional layers are a key component of convolutional neural networks (CNNs) that are used for image and video processing tasks such as object recognition, segmentation, and detection.

The role of convolutional layers is to extract features from the input image using filters, also known as kernels or weights, that slide across the image to perform a mathematical operation called convolution. The filters are learned during the training process, and the values in the filters determine the patterns that the CNN is able to detect.

The convolution operation involves taking the dot product of the filter and a small portion of the input image, called the receptive field or kernel size, at each position of the image. This produces a single value in the output feature map, which corresponds to a specific feature detected in the input image. By sliding the filter across the entire image, the CNN is able to detect features at various positions and scales.

Convolutional layers typically include several filters that produce multiple feature maps, allowing the CNN to learn and detect a variety of features simultaneously. The

output of each convolutional layer is then passed through a non-linear activation function, such as ReLU, to introduce non-linearity into the model.

- Recurrent Neural Networks (RNN)
  - Connections between units may form directed cycles
    - I.e. They propagate data forward, but also back forwards, from later processing stages to earlier stages.

## **7. What is the role of the learning rate in neural network optimization, and how can it be tuned?**

The learning rate determines how much the weights are adjusted in response to the gradient of the loss function during optimization. If the learning rate is too high, the optimization process can become unstable and may not converge to a good solution. If the learning rate is too low, the optimization process may be very slow. The learning rate can be tuned using techniques like grid search or adaptive learning rate methods like Adam.

## **8. What is the vanishing gradient problem in neural networks, and how can it be addressed?**

The vanishing gradient problem occurs when the gradient of the loss function with respect to the weights becomes very small as it propagates through multiple layers of the neural network. This can make it difficult for the network to learn long-term dependencies. One way to address this is to use activation functions like ReLU that do not saturate for large inputs, and to use normalization techniques like batch normalization to ensure that the inputs to each layer are in a suitable range.

# **NLP**

## **1. What are bag-of-words ? How does it work ? Pros and cons of BOW. (一定會考)**

### Definition :

Bag-of-words (BOW) is a technique to extract features from text for use in modeling. It is a representation of text that describes the occurrence of words within a document. It involves a vocabulary of known words and a measure of the presence of known words. Any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document.

### Assumption (why BOW works):

Documents are similar if they have similar content.

The bag-of-words approach involves the following steps:

- 1) Collect data (text document)

- 2) Preprocessing : Unify case, handle mis-spellings, strip stop words (short function words), negation, stemmings
- 3) Determine vocabulary : The text is split into individual words, which are called tokens. All the unique tokens in the text are collected to create a vocabulary.
- 4) Create document vectors : create a fixed-length vector representation of vocabulary, with one position representing each word.
  - a) The simplest scoring method is to mark the presence of words(0 for present, 1 for present)
  - b) Alternative : count/ frequency of words presence

#### Pros of BOW:

- 1) BOW is simple and easy to implement.
- 2) It is computationally efficient and can be used with large datasets.
- 3) BOW can handle variable-length documents and can be used with different text types, such as emails, tweets, and news articles.

#### Cons of BOW: *does not capture language structure and features*

- 1) BOW does not capture the meaning of the text, as it only considers the frequency of individual words.
- 2) It does not consider the context in which the words appear or the relationships between the words.
- 3) BOW can result in high-dimensional feature vectors, which can be difficult to interpret.

Overall, bag-of-words is a widely used technique in NLP, particularly for tasks such as text classification and sentiment analysis, where the focus is on the presence or absence of specific words in the text. However, it is important to be aware of its limitations and consider other approaches when a more nuanced analysis of the text is required.

## **2. What is Word2Vec ? How does it work ? Pros and cons of Word2Vec.**

#### Definition :

Word2Vec is a shallow, two-layer neural network. Input is a large corpus of words and output is vector space, typically of several hundred dimensions. Each unique word in the corpus is assigned a corresponding vector in the space.

How it works: if two different words have very similar contexts, then model tend to output very similar results for these two words (intuitive)

In a Word2Vec, to predict the next word in a sentence given a specific word in the middle of the sentence, the model looks at the words nearby, within a certain window size. This window size is typically 5 words behind and 5 words ahead of the input word. From this set of nearby words, the model randomly selects one and predicts the probability of every word in the vocabulary being the chosen word. The output probabilities indicate how likely it is to find each vocabulary word nearby the input word, based on the model's learned patterns and associations between

words in the training data. These probabilities can be utilized for a range of natural language processing applications, such as text classification, language translation, and text generation.

Notes:

- Contrary to BOW, it has been found that little to no preprocessing performs best in Word2Vec

Contrary to the Bag-of-Words (BOW) approach, it has been observed that little to no preprocessing works best for Word2Vec because the model is able to learn the context and relationships between words from the raw text data.

In BOW, the focus is on the frequency of words in the text, which requires preprocessing steps like tokenization, stop-word removal, and stemming/lemmatization. However, in Word2Vec, the focus is on the distributional semantics of words, which involves capturing the context in which words appear in the text.

Preprocessing steps like stemming or lemmatization can change the form of words, which can potentially remove some of the variation in the corpus and obscure meaningful relationships between words. Additionally, stop-word removal can discard potentially informative words like "not" or "very," which can affect the meaning of the surrounding words.

By avoiding preprocessing and using raw text data, Word2Vec can better capture the nuances and complexity of natural language, including idiomatic expressions, metaphors, and other figurative language. This can lead to more accurate and nuanced word embeddings, which can be beneficial for a range of natural language processing tasks, such as text classification, language translation, and text generation.

- There is no activation function on the hidden layer neurons, but the output neurons use softmax
  - Probability that if you randomly pick a word nearby "A", that it is "B"

There are two main approaches to training Word2Vec models: continuous bag-of-words (CBOW) and skip-gram. In the CBOW approach, the model predicts the target word based on its neighboring words. **In the skip-gram approach, the model predicts the neighboring words given a target word(考试范围).**

Pros of Word2Vec:

- Word2Vec captures the semantic and syntactic relationships between words, which makes it useful for a wide range of NLP tasks.
- It can handle out-of-vocabulary words, as the word embeddings are generated based on the context in which the words appear, rather than being predefined.
- Word2Vec can generate high-quality word embeddings even with a small amount of training data.

Cons of Word2Vec:

- The quality of the word embeddings is highly dependent on the quality and size of the training data.
- The training process can be computationally expensive, particularly for large datasets.
- Word2Vec does not capture the relationships between words at the sentence or document level.