The project studies the classification problem, so the dependent variable of the model is classification variable (0 and 1) and the independent and dependent variables of the model are nonlinear. Besides, the project analyses the relationship between the probability that a dependent variable takes a certain value and the independent variable. As a result, logistic regression model is more suitable for this project and the project selected the logistic regression model of the sklearn package as the classifier.

**Logistic regression:**

Logistic regression is well suited to describe the relationship which is expressed as probability between classification results and one or more classifications [5]. It can adapt to multiple classification results. In this project, logistic regression is used to calculate the probability of a binary event occurring under multiple independent features [4].

Model:

denotes the vector of feature variables, and denotes the associated binary output. represents the weight vector. The logistic regression has model:

Logistic loss function:

Average logistic loss:

Problem:

Overfitting problem: in supervised learning when there are many input features, but only a small number of key features determine the classification target. That is, when the number of training set data is insufficient, the classification model may perform well on the training dataset but not well on the test dataset [6]. Thus, when there are many features, overfitting will become a problem of the model unless the training set is ample [3]. In order to solve this problem, L1 and L2 regularization were used.

L1:

Lasso (L1) penalty encourages the sum of the absolute values of the to be small [3]. It uses sparsity to fit model with many features [1]. The sparsity means that L1 penalty will automatically filter some features that have less impact on classification. L1 penalty achieves the filtering by reducing the regression coefficient to 0 and slightly reducing other regression coefficients [2].

L2:

L2 penalty encourages the sum of the squares of the to be small [3]. It will reduce the regression coefficient but will not be zero [2]. Thus, if each feature has an effect on the classification, L2 penalty is more suitable.

**Result**

Noun phrase:

Feature selection: ‘proportion’, ‘ratio’ and ‘average distance’. Among them, ‘proportion’ represents the number of shop whose distance from the central point less than the median distance/ The total number of shops contain the noun phrase. ‘ratio’ means the number of shops whose distance is less than 20000 meters from the center point/ total shop number. ‘average distance’ reflects the average distance of all shops from the central point.

The classification rate means the degree of fit between the logistic regression model and the training set.

|  |  |  |
| --- | --- | --- |
| Penalty | L1 | L2 |
| Classification rate | 89.74%. | 89.74%. |
| Percentage threshold | 50% | 50% |
| Coefficient (‘proportion’, ‘ratio’ and ‘average distance’) | 1.43506897e+00  4.21097118e+00  -1.42088335e-05 | 1.27526880e+00  1.74406725e+00  -1.25081676e-05 |
| The number of regional noun phrases | 65 | 67 |
| Regional noun phrases with probability | See full list of regional noun phrases in Appendix A-A.1 | See full list of regional noun phrases in Appendix A-A.2 |

**Table 1: Comparison of penalty choices for noun phrase**

After comparing the classification results of noun phrase, the project found that there is not much difference between using L1 penalty and L2 penalty. This is because that the project selected just three main features and all these three features have impact on the noun phrase classification results. The reason for choosing these three features instead of the five in the decision tree is that from the results of the decision tree, the impact of ‘city number’ feature on the classification results is negligible. Besides, the project found that if the project adds ‘city number’ and ‘shop number’ features, the model will overfit the training dataset that noun phrases with less than 10 shops will not be excluded. This is because that in the logistic model, ‘shop number’ feature has the smaller weight than other features. Thus, some words that should not participate in classification are given regionality with high probability. In order to solve the above problems, the project ignored ‘city number’ feature and manually excluded words with less than 10 in the test dataset.

Independent word:

If the project chooses the same feature as noun phrase, the logistic regression model result of independent word will be showed as the following table:

|  |  |  |
| --- | --- | --- |
| Penalty | L1 | L2 |
| Classification rate | 84.70% | 55.29% |
| Percentage threshold | 50% | Cannot tell |
| Coefficient (‘proportion’, ‘ratio’ and ‘average distance’) | 4.88476603e+00  3.00632484e+00  -1.99347796e-05 | 3.44111057e-12,  8.63444565e-12,  -1.77054477e-06 |
| The number of regional words | 55 | Cannot tell |

**Table 2: Comparison of penalty choices for independent** **word when selecting ‘proportion’, ‘ratio’ and ‘average distance’ as features**

According to the result of Table 2, using different penalties will cause the difference in classification rates very large. Besides, when using L2 penalty, all words were judged as national words. Thus, the project cannot tell the percentage threshold and the number of regional noun phrases. By comparing coefficient, the project found that, when using L1 penalty, the L1 penalty model reduced the coefficient of average distance feature to close to zero. However, when the project used L2 penalty, coefficients of all features closed to zero and coefficient of average distance is the largest. After comparing the training sets, the project found that the ‘average distance’ feature does not help the classification even has a negative impact on the classification. In the training set, words whose has large average distance appear in national categories and also in regional categories and words with small ‘average distance’ is same. As a consequence, the model is forced to fit the average distance feature when using L2 penalty, which leads to the inability to classify. Thus, the project removed the ‘average distance’ feature and got the following findings:

|  |  |  |
| --- | --- | --- |
| Penalty | L1 | L2 |
| Classification rate | 90.58% | 89.41% |
| Percentage threshold | 50% | 50% |
| Coefficient (‘proportion’, ‘ratio’ and ‘average distance’) | 0.  6.57330811 | 0.48703891  2.65330273 |
| The number of regional words | 49 | 50 |
| Regional words with probability | Meaningless | A-A.3 |

**Table 3: Comparison of penalty choices for independent** **word when selecting ‘proportion’ and ‘ratio’ as features**

According to the results showed in Table. 3, the project found that after removing the ‘average distance’ feature, if the model use L1 penalty, the impact of ‘proportion’ feature will be gone. This means the classification all depends on the ‘ratio’ feature. However,

‘proportion’ is still an important feature for classification. For example, ‘roe’ and ‘pasti’ are regional words, but their ratio is low. Thus, their regional probability is less than 50% and they are classified as non-regional words. When the project used L2 penalty, the model reduced the impact of ‘ratio’ feature and increased the impact of ‘probability’ feature. However, the impact of ‘probability’ feature is still very small that regional words such as ‘roe’, ‘pasti’, ‘kidney’ and ‘bolognese’ were judged as national words.

Word-pair:

According to the result of word pair decision tree, the project found that ‘proportion’ feature, ‘ratio’ feature, ‘city number’ feature and ‘average distance’ feature all have impact on the classification of the word pair. Thus, the project firstly used these four features to test the classification effect and got the following result:

|  |  |  |
| --- | --- | --- |
| Penalty | L1 | L2 |
| Classification rate | 88.77% | 74.48% |
| Percentage threshold | Cannot tell | Cannot tell |
| Coefficient (‘proportion’, ‘ratio’, ‘city number’ and ‘average distance’) | 1.83013399e+00,  6.22791230e+00,  -2.69416502e-01,  -7.79706487e-06 | 1.06066688e-01  1.80860335e-01,  -1.26984839e-01  5.89637008e-06 |
| The number of regional words | Cannot tell | Cannot tell |
| Regional words with probability | meaningless | meaningless |

**Table 4: Comparison of penalty choices for word pair when selecting ‘proportion’, ‘ratio’, ‘city number’ and ‘average distance’ as features**

From the Table. 4, the project found that when using L1 penalty, the L1 algorithm reduced the impact of ‘average distance’ feature to close to zero. Thus, in the result of regional word pair, word pairs such as ‘or garlic’ and ‘tikka meat’ which have ten to twenty shops. Besides, their ratio higher than 60%, but have larger than 200000 meters average distance are all judged as regional word pairs with high probability. However, the threshold of ‘average distance’ feature for regional word pairs which have ten to twenty shops is set as lower than 200000 meters. Thus, the ‘average distance’ feature should have great impact, but the logistic regression model (whether it is L1 penalty or L2 penalty) ignored this feature. Besides, in terms of ‘or garlic’ and ‘tikka meat’, if the project split these word pair into independent words, these independent words also did not appear in the results of the previous regional classification (whether it is decision tree or logistic regression classification). However, it is interesting that in the logistic classification results of noun phrases and independent words, the impact of ‘average distance’ feature of both of them was all closed to zero, but their classification results were expected result. After comparing the classification result of these three kinds of dataset, the project found that in the dataset of noun phrases and independent words, the number of phrases or words such as ‘or garlic’ and ‘tikka meat’ is few. Thus, their classification results are not affected much. However, in the word pair dataset, the number of word pairs such as ‘or garlic’ and ‘tikka meat’ is large, so the results were very different from expectations. As a consequence, in terms of word pair, the logistic regression is not suitable for it. The root cause of this is because of the imperfections of the training set. However, it is difficult to find more representative data which can let the effect of the ‘average distance’ feature over the ratio feature. The solution to solve this problem that the project can think of is to get more data sets.

**References**

[1] Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(3), 273-282.

[2] Goeman, J., Meijer, R., & Chaturvedi, N. (2012). L1 and L2 penalized regression models. *cran. r-project. or*.

[3] Ng, A. Y. (2004, July). Feature selection, L 1 vs. L 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning* (p. 78). ACM.

[4] Walker, S. H., & Duncan, D. B. (1967). Estimation of the probability of an event as a function of several independent variables. *Biometrika*, *54*(1-2), 167-179.

[5] Peng, C. Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *The journal of educational research*, *96*(1), 3-14.

[6] Koh, K., Kim, S. J., & Boyd, S. (2007). An interior-point method for large-scale l1-regularized logistic regression. *Journal of Machine learning research*, *8*(Jul), 1519-1555.

**Appendix A**

Logistic regression classifier results

A.1 Regional Noun phrase classification result (L1 penalty)

|  |  |
| --- | --- |
| **Noun phrases** | **Nation probability, Regional probability** |
| **cheese pattie** | [0.013309049172268272, 0.9866909508277317] |
| **hamburger supper** | [0.0186291426474251, 0.9813708573525749] |
| **pizza crunch** | [0.022596261386979788, 0.9774037386130202] |
| **potato pie** | [0.023568146763607967, 0.976431853236392] |
| **inferno** | [0.03220485137533258, 0.9677951486246674] |
| **naan** | [0.03268870363025178, 0.9673112963697482] |
| **pineapple ring** | [0.034481232740873224, 0.9655187672591268] |
| **pudding supper** | [0.03671211066273117, 0.9632878893372688] |
| **pizza supper** | [0.037933276927266446, 0.9620667230727336] |
| **macaroni cheese** | [0.038477080449522916, 0.9615229195504771] |
| **diet bru** | [0.041033740318725975, 0.958966259681274] |
| **chip shop takeaway order** | [0.049003760243348515, 0.9509962397566515] |
| **diet coke ltr** | [0.05278428446365713, 0.9472157155363429] |
| **hamburger** | [0.05353617178504022, 0.9464638282149598] |
| **pattie** | [0.06093147368861507, 0.9390685263113849] |
| **chip roll** | [0.06283895877899259, 0.9371610412210074] |
| **dandelion** | [0.06420796357829794, 0.9357920364217021] |
| **chicken meat** | [0.06710259810974517, 0.9328974018902548] |
| **suey roll** | [0.07105552105199131, 0.9289444789480087] |
| **king rib** | [0.07476984265165865, 0.9252301573483414] |
| **bit** | [0.07763404564334098, 0.922365954356659] |
| **pollo** | [0.08065191355388068, 0.9193480864461193] |
| **pie supper** | [0.08337077826346695, 0.916629221736533] |
| **shot** | [0.09245513281940321, 0.9075448671805968] |
| **burdock** | [0.11106739501985041, 0.8889326049801496] |
| **cheese tomato** | [0.1334193658886207, 0.8665806341113793] |
| **beef onion pie** | [0.1360885812031546, 0.8639114187968454] |
| **haggis** | [0.14537848581225887, 0.8546215141877411] |
| **funghi** | [0.15603603785303422, 0.8439639621469658] |
| **chicken breast supper** | [0.16533691272400008, 0.8346630872759999] |
| **pasty** | [0.16887846051290278, 0.8311215394870972] |
| **supper** | [0.1828044881275689, 0.8171955118724311] |
| **fish chips** | [0.19210831009980167, 0.8078916899001983] |
| **cheeseburger half pounder** | [0.20097439977528253, 0.7990256002247175] |
| **cheeseburger quarter pounder** | [0.20097439977528253, 0.7990256002247175] |
| **chicken pakora** | [0.2495477396698309, 0.7504522603301691] |
| **cookies** | [0.2592799667107383, 0.7407200332892617] |
| **bull** | [0.26622326820247466, 0.7337767317975253] |
| **bru** | [0.2797324130622031, 0.7202675869377969] |
| **chicken nuggets meal** | [0.297673913638301, 0.702326086361699] |
| **nuggets** | [0.31071290604648216, 0.6892870939535178] |
| **vegetable pakora** | [0.3141644368669968, 0.6858355631330032] |
| **pasties** | [0.32791435320038853, 0.6720856467996115] |
| **spaghetti** | [0.3284135989436908, 0.6715864010563092] |
| **chips salad** | [0.33646537799041176, 0.6635346220095882] |
| **scallops** | [0.3597985894131531, 0.6402014105868469] |
| **pop** | [0.3661374073969268, 0.6338625926030732] |
| **pakora** | [0.38093837612429626, 0.6190616238757037] |
| **site** | [0.3822906709066207, 0.6177093290933793] |
| **pasta** | [0.3924413678971085, 0.6075586321028915] |
| **kebab wrap** | [0.4060773055076441, 0.5939226944923559] |
| **inch margherita** | [0.4082409512546662, 0.5917590487453338] |
| **sausage supper** | [0.4371353690721911, 0.5628646309278089] |
| **piece** | [0.44206894811490416, 0.5579310518850958] |
| **inch bread** | [0.4448106111373529, 0.5551893888626471] |
| **mince** | [0.4451640822512588, 0.5548359177487412] |
| **tray** | [0.45264282577407045, 0.5473571742259296] |
| **pie chips** | [0.45505646034034897, 0.544943539659651] |
| **cod roe** | [0.46053626058973207, 0.5394637394102679] |
| **spring roll** | [0.4624339277638466, 0.5375660722361534] |
| **facebook** | [0.4765961922514401, 0.5234038077485599] |
| **spam fritter** | [0.4841356602295611, 0.5158643397704389] |
| **kidney** | [0.4919224144884715, 0.5080775855115285] |

A.2 Regional Noun phrase classification result (L2 penalty)

|  |  |
| --- | --- |
| **Noun phrases** | **Nation probability, Regional probability** |
| **cheese pattie** | [0.04302663367975612, 0.9569733663202439] |
| **hamburger supper** | [0.05669634713763139, 0.9433036528623686] |
| **pizza crunch** | [0.06500484200373025, 0.9349951579962698] |
| **potato pie** | [0.06658411254621566, 0.9334158874537843] |
| **inferno** | [0.07750836744789169, 0.9224916325521083] |
| **pineapple ring** | [0.08145369004390668, 0.9185463099560933] |
| **naan** | [0.08670872839951937, 0.9132912716004806] |
| **macaroni cheese** | [0.09432914146692861, 0.9056708585330714] |
| **pudding supper** | [0.09525936198881202, 0.904740638011188] |
| **pizza supper** | [0.09820495677690055, 0.9017950432230994] |
| **diet bru** | [0.09986031241828819, 0.9001396875817118] |
| **chip shop takeaway order** | [0.11084386246459865, 0.8891561375354013] |
| **diet coke ltr** | [0.11520124734668147, 0.8847987526533185] |
| **suey roll** | [0.11605131791091994, 0.8839486820890801] |
| **pattie** | [0.12619217081624445, 0.8738078291837555] |
| **hamburger** | [0.1293061165498678, 0.8706938834501322] |
| **chicken meat** | [0.13293857088732897, 0.867061429112671] |
| **dandelion** | [0.13319255416754494, 0.8668074458324551] |
| **chip roll** | [0.13874956035593355, 0.8612504396440664] |
| **bit** | [0.14303300507885575, 0.8569669949211443] |
| **pollo** | [0.15360521749090306, 0.8463947825090969] |
| **king rib** | [0.1536955211457186, 0.8463044788542814] |
| **shot** | [0.1593261813458673, 0.8406738186541327] |
| **burdock** | [0.16458133161698252, 0.8354186683830175] |
| **pie supper** | [0.1776010926627819, 0.8223989073372181] |
| **funghi** | [0.19483683998675805, 0.805163160013242] |
| **cheese tomato** | [0.20707470207423562, 0.7929252979257644] |
| **beef onion pie** | [0.20907048034755127, 0.7909295196524487] |
| **pasty** | [0.22080400831990743, 0.7791959916800926] |
| **haggis** | [0.22384433767294343, 0.7761556623270566] |
| **cheeseburger half pounder** | [0.24157381600573946, 0.7584261839942605] |
| **cheeseburger quarter pounder** | [0.24157381600573946, 0.7584261839942605] |
| **chicken breast supper** | [0.2433056468951732, 0.7566943531048268] |
| **fish chips** | [0.2505011589087093, 0.7494988410912907] |
| **supper** | [0.2820484701165006, 0.7179515298834994] |
| **cookies** | [0.2926068844345514, 0.7073931155654486] |
| **bull** | [0.3199562641957926, 0.6800437358042074] |
| **chicken nuggets meal** | [0.32125571887427606, 0.6787442811257239] |
| **chicken pakora** | [0.32258630530369103, 0.677413694696309] |
| **nuggets** | [0.33817842687597777, 0.6618215731240222] |
| **pasties** | [0.3434913363840276, 0.6565086636159724] |
| **chips salad** | [0.3437544174739232, 0.6562455825260768] |
| **scallops** | [0.37133107972519297, 0.628668920274807] |
| **pop** | [0.37566114070951995, 0.62433885929048] |
| **site** | [0.3828808722957264, 0.6171191277042736] |
| **bru** | [0.3870878662375089, 0.6129121337624911] |
| **inch bread** | [0.3949447590422569, 0.6050552409577431] |
| **inch margherita** | [0.40518983399044395, 0.594810166009556] |
| **kebab wrap** | [0.4092167362183604, 0.5907832637816396] |
| **spaghetti** | [0.4148651966000986, 0.5851348033999014] |
| **vegetable pakora** | [0.4220166362053063, 0.5779833637946937] |
| **pie chips** | [0.42219252756608205, 0.577807472433918] |
| **tray** | [0.42520318676351454, 0.5747968132364855] |
| **sausage supper** | [0.43141384413907924, 0.5685861558609208] |
| **cod roe** | [0.4401889430287208, 0.5598110569712792] |
| **cod fish** | [0.4441861002797133, 0.5558138997202867] |
| **rubicon mango** | [0.4468756085262042, 0.5531243914737958] |
| **spam fritter** | [0.4510542222928562, 0.5489457777071438] |
| **fish bites** | [0.45304200897496705, 0.546957991025033] |
| **spring roll** | [0.454370446988012, 0.545629553011988] |
| **piece** | [0.4562983102715691, 0.5437016897284309] |
| **facebook** | [0.4754315709258996, 0.5245684290741004] |
| **pakora** | [0.47801416041298306, 0.5219858395870169] |
| **mince** | [0.47887974808592726, 0.5211202519140727] |
| **kidney** | [0.47997648893148603, 0.520023511068514] |
| **inch meat feast** | [0.4899211474645704, 0.5100788525354296] |
| **pasta** | [0.49921432925156173, 0.5007856707484383] |

A.2 Independent word phrase classification result (L2 penalty)

|  |  |
| --- | --- |
| **Noun phrases** | **Nation probability, Regional probability** |
| **hamburger** | [0.27518815883964054, 0.7248118411603595] |
| **rump** | [0.29527831457748477, 0.7047216854225152] |
| **cob** | [0.2977341795415793, 0.7022658204584207] |
| **carbonara** | [0.3019618738229689, 0.6980381261770311] |
| **yorkshire** | [0.31070428516008153, 0.6892957148399185] |
| **haagen** | [0.31070428516008153, 0.6892957148399185] |
| **instantly** | [0.3127942768830314, 0.6872057231169686] |
| **naan** | [0.3253809430526645, 0.6746190569473355] |
| **dazs** | [0.3336191488527629, 0.6663808511472371] |
| **rib** | [0.35008689580727004, 0.64991310419273] |
| **suey** | [0.3536508634262119, 0.6463491365737881] |
| **inferno** | [0.3607444550688824, 0.6392555449311176] |
| **scallop** | [0.3675104078076613, 0.6324895921923387] |
| **balti** | [0.369659095446059, 0.630340904553941] |
| **crunch** | [0.37700591256483185, 0.6229940874351682] |
| **securely** | [0.37917683460134133, 0.6208231653986587] |
| **macaroni** | [0.37981010769493173, 0.6201898923050683] |
| **shot** | [0.380443790303286, 0.619556209696714] |
| **quattro** | [0.38504629044553984, 0.6149537095544602] |
| **keema** | [0.38504629044553984, 0.6149537095544602] |
| **pakora** | [0.38659679766675625, 0.6134032023332437] |
| **spaghetti** | [0.38659679766675625, 0.6134032023332437] |
| **rock** | [0.38723473342735193, 0.6127652665726481] |
| **meaty** | [0.3954763196041714, 0.6045236803958286] |
| **parmesan** | [0.40066686023815656, 0.5993331397618434] |
| **haggis** | [0.4015926708516465, 0.5984073291483535] |
| **skate** | [0.4015926708516465, 0.5984073291483535] |
| **splash** | [0.407177446787387, 0.592822553212613] |
| **bru** | [0.407335901975577, 0.592664098024423] |
| **irn** | [0.407335901975577, 0.592664098024423] |
| **cornish** | [0.414003313518011, 0.585996686481989] |
| **burdock** | [0.425990710373368, 0.574009289626632] |
| **dandelion** | [0.425990710373368, 0.574009289626632] |
| **passion** | [0.4262755133396918, 0.5737244866603082] |
| **stagioni** | [0.4262755133396918, 0.5737244866603082] |
| **smokey** | [0.4310467161720981, 0.5689532838279019] |
| **bull** | [0.4321168370536844, 0.5678831629463156] |
| **guava** | [0.43702866884927116, 0.5629713311507288] |
| **pattie** | [0.4406267682710011, 0.5593732317289989] |
| **give** | [0.4466373029798988, 0.5533626970201012] |
| **kiev** | [0.45120536975813474, 0.5487946302418653] |
| **chosen** | [0.4537444662923733, 0.5462555337076267] |
| **pukka** | [0.45615988893423687, 0.5438401110657631] |
| **value** | [0.45682712532342884, 0.5431728746765712] |
| **facebook** | [0.4716521072822266, 0.5283478927177734] |
| **smoked** | [0.48409417370055885, 0.5159058262994412] |
| **samosa** | [0.4870351320656271, 0.5129648679343729] |
| **bit** | [0.4909771542570499, 0.5090228457429501] |
| **cucumber** | [0.4911041698899524, 0.5088958301100476] |
| **lucozade** | [0.49382989642049147, 0.5061701035795085] |