FORDHAM UNIVERSITY

THE JESUIT UNIVERSITY OF NEW YORK

DATA MINING FINAL REPORT

Tong Wang

Ming Chen

Yun Song

Wenjia Zheng

Shuyan Liu

content

1. Prepossessing 2

1.1 Data description 2

1.2 Missing data 9

1.3 Attributes 10

1.4 Feature Filter 20

2.The choose of criterion--roc_auc 22

2.1 Random Forest 22

2.2 SVM 26

2.3 KNN 29

2.4 Logistic Regression 31

3.The Ensemble 33

4. Conclusion 34

1. Prepossessing

1.1 Data description

1.1.1 General introduction of dataset

The dataset is from the UCI machine learn- ing lab, which represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. The data contains 50 attributes such as race, gender, age, HbA1c test result, diabetic medications, etc. And the dataset includes 101766 instances.

1.1.2 Data source

The data are submitted on behalf of the Center for Clinical and Translational Research, Virginia Commonwealth University, a recipient of NIH CTSA grant UL1 TR00058 and a recipient of the CERNER data. John Clore (jclore '@' vcu.edu), Krzysztof J. Cios (kcios '@' vcu.edu), Jon DeShazo (jpdeshazo '@' vcu.edu), and Beata Strack (strackb '@' vcu.edu). This data is a de-identified abstract of the Health Facts database (Cerner Corporation, Kansas City, MO).

1.1.3 Data type

There are two types of data: numeric and nominal

Table 1.1 list of numeric attributes

attribute name	description	Missing value

encounter_id	Unique code of each encounter	0%
patient_nbr	Unique code of each patient	0%
weight	Each patient’s weight (in pounds)	97%
time_in_hospital	Integer number of days between admission and discharge	0%
num_lab_procedures	Integer number of lab procedures	0%
num_procedures	Integer number of procedures	0%
num_medications	Integer number of distinct generic names administered during the encounter	0%
number_outpatient	Integer number of outpatient visits of the patient in the year	0%
number_emergency	Intger number of emergency visits of the patient in the year	0%
number_diagnoses	Integer Number of diagnoses entered to the system	0%

Table 1.2 list of categorical attributes

attribute name	Description	Missing value
race	Category: Caucasian, Asian, African America, Hispanic, other	2%
gender	Category: male, female, unknown/invaild	0%

age	Category: [0-10], [10-20], [20-30], [30-40], [40-50], [50-60], [60-70], [70-80], [80-90], [90-100]	0%
admission_type_id	Category: integer 8 distinct values mean different types.	0%
discharge_disposition_id	Category: integer 26 distinct values mean different types.	0%
admission_source_id	Category: integer 17 distinct values mean different types.	0%
payer_code	Category: integer 23 distinct values mean different types.	52%
medical_specialty	Category: integer 84 distinct values mean different types.	53%
diag_1	Category: distinct ranges of values mean different types.	0.02%
diag_2	Category: distinct ranges of values mean different types.	0.35%
diag_3	Category: distinct ranges of values mean different types.	1%
max_glu_serum	Indicates the range of the Glucose serum test result or if the test was not taken. Values: “>200,” “>300,” “normal,” and “none” if not measured	0%
A1Cresult	Indicates the range of the A1c test result or	0%

	if the test was not taken. Values: “>8”: >8%, “>7”: >7%&<8%, “normal” : <7%, “none” :not measured.	
Change of medications	Category: distinct values mean whether diabetic medication were prescribed. Values: yes & no	0%
Metformin	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
Repaglinide	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
nateglinide	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
chlorpropamide	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
glimepiride	Category: distinct values mean the change of the drug dosage or whether the drug was	0%

	prescired Values: down, steady, up, no	
acetohexamide	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
glipizide	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
glyburide	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
tolbutamide	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
pioglitazone	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
rosiglitazone	Category: distinct values mean the change of the drug dosage or whether the drug was prescired	0%

	Values: down, steady, up, no	
acarbose	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
miglitol	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
troglitazone	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
tolazamide	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
examide	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
citoglipton	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%

insulin	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
glyburide-metformin	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
glipizide-metformin	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
glimepiride-pioglitazone	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
metformin-rosiglitazone	Category: distinct values mean the change of the drug dosage or whether the drug was prescired Values: down, steady, up, no	0%
metformin-pioglitazone	Category: distinct values mean the change of the drug dosage or whether the drug was prescribed Values: down, steady, up, no	0%
Change	Category: distinct values mean whether	0%

	there was a change in diabetic medications Values: change, no change	
diabetesMed	Category: distinct values mean whether diabetic medications were prescribed Values: no yes	0%
Readmitted	Category: distinct values mean days of inpatient readmission or whether impatient were readmitted. Values: No, >30, <30	0%

1.2 Missing data

There are 7 attributes which have missing data. We decide to deal with these missing data according to the proportion of missing data.

1.2.1 Proportion of missing data greater than 50%

Attribute name	Proportion of missing data
Weight	97%
payer_code	52%
medical_specialty	53%

Because the proportion of missing data is greater than 50%, if we replace them, there are too many data which are predicted. In other word, the attribute data are full of subjective prediction and lack of confidence, that will have a strong negative influence on our model prediction. So we drop these three attributes.

1.2.2 Proportion of missing data less than 50%

Attribute name	Proportion of missing value
race	2%
diag_1	0.02%
diag_2	0.35%
diag_3	1%

For these missing data, due to the number of dataset is over than 100000, these instances with missing values just make a little difference on model prediction. Therefore, we drop the instance with these missing data.

1.3 Attributes

1.3.1 Numeric Attributes

“num_lab_procedures”, “num_procedures”, “num_medications”, “number_outpatient”, “number_emergency”, “number_inpatient” and “number_diagnoses” are numeric attributes. So we have no need to deal with these attributes.

1.3.2 Categorical Attributes

Normal attributes processing

“race” is an unordered categorical attribute. According to data, this attribute already have 5 type classes. After drop instances that include missing data we could directly use this classification.

“gender” is an unordered categorical attribute. According to data, this attribute already have 5 type classes. We could directly use this classification.

“age” is expressed by 10 age ranges. So it is not a continuous numerical attribute but a ordered categorical attribute. We could replace them using ordered integers.

“max_glu_serum” is an unordered categorical attribute. According to data, this attribute already have 4 type classes. We could directly use this existing classification without change.

“A1Cresult” is an unordered categorical attribute. According to data, this attribute already have 4 type classes. We could directly use this existing classification without change.

“change” is a categorical attribute. According to data, this attribute already have 2 type classes. We could directly use this existing classification without change.

“diabetesMed” a categorical attribute. According to data, this attribute already have 2 type classes. We could directly use this existing classification without change.

Attribute that need to compress original classes

“admission_type_id” is an unordered categorical attribute and has 8 type classes. Count number of encounters of each class is: “emergency” 53990, “urgent” 18480, “elective” 18869, “newborn” 10, “not available” 4785, “null” 5291, “trauma center”

21, “not mapped” 320. According to the mapping file we could compress to 4 type classes. “emergency”, “urgent” and “elective” have large proportion. We compress the rest of classes that each of them has small proportion into one class “other”.

“admission_source_id” is an unordered categorical attribute and has 21 type classes. Count number of encounters of each class is: “physician referral” 29565, “emergency room” 57494, “transfer from a hospital” 3187. Due to the meaning of each class is separate and independent. we decide to extract two classes “physician referral” and “emergency room” that have large proportion as two independent classes. And then we compress the rest of classes that each of them has small proportion into one class “other”.

“diag_1”&”diag_2”&”diag_3”

This attribute refer to the primary diagnosis (coded as first three digits of ICD9), according to wekipedia International Statistical Classification of Diseases and Related Health Problems as follows:

List of ICD-9 codes 001–139: infectious and parasitic diseases

List of ICD-9 codes 140–239: neoplasms

List of ICD-9 codes 240–279: endocrine, nutritional and metabolic diseases, and immunity disorders

List of ICD-9 codes 280–289: diseases of the blood and blood-forming organs

List of ICD-9 codes 290–319: mental disorders

List of ICD-9 codes 320–389: diseases of the nervous system and sense organs

List of ICD-9 codes 390–459: diseases of the circulatory system

List of ICD-9 codes 460–519: diseases of the respiratory system

List of ICD-9 codes 520–579: diseases of the digestive system

List of ICD-9 codes 580–629: diseases of the genitourinary system

List of ICD-9 codes 630–679: complications of pregnancy, childbirth, and the puerperium

List of ICD-9 codes 680–709: diseases of the skin and subcutaneous tissue

List of ICD-9 codes 710–739: diseases of the musculoskeletal system and connective tissue

List of ICD-9 codes 740–759: congenital anomalies

List of ICD-9 codes 760–779: certain conditions originating in the perinatal period

List of ICD-9 codes 780–799: symptoms, signs, and ill-defined conditions

List of ICD-9 codes 800–999: injury and poisoning

List of ICD-9 codes E and V codes: external causes of injury and supplemental classification

According to the above classification range, we could transfer these encounters into 19 categories in advance.

And then we count the number of each of these 19 categories and sort them in descending order. The result shows in the Table.

In these condition, we could find that some number of categories are less than other obviously. For example, encounters that suffer diseases of the sense organs(216) are less than diseases of the respiratory system(9490).

So we decide to reserve 8 categories that have large number of encounters. And then combine the rest of categories into one category. And name it “other”.

There are two important points. Firstly, we could notice in category “other symptoms, signs, and ill-defined conditions” include some categories need to extract add into one of the 8 categories. For example, the category “785 Symptoms involving cardiovascular

system" is belong to "other", but we decide extract it and add into category of "circulatory", because we think they have the same meaning. More details show in Table.

Secondly, due to we are dealing with the problem of diabetes, so we extract the category of "250 diabetes" as an independent category to optimize preprocessing.

In conclusion, the final classification is as follows:

Group name	ICD 9 code	Number of encounters
Circulatory	390-459,785	21,411
Respiratory	460-519,786	9,490
Digestive	520-579,787	6,485
Diabetes	250	5,747
Injury	800-999	4,697
Musculoskeletal	710-739	4,076
Genitourinary	580-629,788	3,435
Neoplasms	140-239	2,536
Other	780, 781, 784, 790–799	2,136
	240-279, without 250	1,851
	680-709,782	1,846
	011-139	1,683
	E-V	1,544
	280-289	918
	320-359	652

	630-679	634
	360-389	586
	740-759	41

Special meaningful attribute processing

“discharge_disposition_id” is an unordered categorical attribute and has 29 type classes. Count number of encounters of each class is: “discharged to home” 69234, “expired” 1642, “discharged/transferred to another short term hospital” 2128, etc. According to the mapping file we could compress to 2 type classes. Although there are 29 type classes, on the basis of each meaning we could find out there are only 3 type classes in a rough angle that are “home”, “other” and “expired”. Considering “expired” instances are not make sense to our result. After drop instances that include “expired” data we could get 2 type classes ultimately.

The distinct values of discharge_disposition_id mean different discharge condition of patient.

discharge_disposition_id	description
1	Discharged to home
2	Discharged/transferred to another short term hospital
3	Discharged/transferred to SNF
4	Discharged/transferred to ICF
5	Discharged/transferred to another type of inpatient care institution
6	Discharged/transferred to home with home health service

7	Left AMA
8	Discharged/transferred to home under care of Home IV provider
9	Admitted as an inpatient to this hospital
10	Neonate discharged to another hospital for neonatal aftercare
11	Expired
12	Still patient or expected to return for outpatient services
13	Hospice / home
14	Hospice / medical facility
15	Discharged/transferred within this institution to Medicare approved swing bed
16	Discharged/transferred/referred another institution for outpatient services
17	Discharged/transferred/referred to this institution for outpatient services
18	NULL
19	Expired at home. Medicaid only, hospice.
20	Expired in a medical facility. Medicaid only, hospice.
21	Expired, place unknown. Medicaid only, hospice.
22	Discharged/transferred to another rehab fac including rehab units of a hospital .
23	Discharged/transferred to a long term care hospital.
24	Discharged/transferred to a nursing facility certified under Medicaid but not certified under Medicare.
25	Not Mapped
26	Unknown/Invalid

30	Discharged/transferred to another Type of Health Care Institution not Defined Elsewhere
27	Discharged/transferred to a federal health care facility.
28	Discharged/transferred/referred to a psychiatric hospital of psychiatric distinct part unit of a hospital
29	Discharged/transferred to a Critical Access Hospital (CAH).

From the descriptions mentioned above, we could find that some values of 'discharge_disposition_id' mean the patient was died.

11	Expired
19	Expired at home. Medicaid only, hospice.
20	Expired in a medical facility. Medicaid only, hospice.
21	Expired, place unknown. Medicaid only, hospice.

And then we found that the all these instances corresponding 'readmitted' value are 'NO', in other word, these patients didn't go back to the hospital not because of encounters, but because of their expiration, that produces a bad effort on our accuracy of prediction. So we drop the instances which 'discharge_disposition_id' equal to '11', '19', '20' and '21'.

Meaningless attributes

After analyzing the meaning of attributes, we found that some of attributes make no effect on our prediction.

attribute name	description
encounter_id	Unique code of each encounter

patient_nbr	Unique code of each patient

"encounter_id" is used to differ diverse encounters and "patient_nbr" is used to distinguish different patients. They attribute nothing to our prediction. So we drop these features.

Attributes which all of their values are closed to one specific value

"23 features for medications"(metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, citoglipton, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone) are all unordered categorical attributes. According to data, this attribute already have 4 type classes. "up" if the dosage was increased during the encounter, "down" if the dosage was decreased, "steady" if the dosage did not change, and "no" if the drug was not prescribed. We could directly use this existing classification without change. Count number of encounters of each class shows in the below Table.

After analyzing the meaning of attributes, we also found that most of values of attributes are closed to one specific value. So there is little relationship between this attributes and 'readmitted' label, we decide to drop them.

	Metformin	Repaglinide	nateglinide	chlorpropamide	glimepiride	acetohexamide
Down	575	45	11	1	194	0
No	81778	10027	101063	101680	96575	101765
Steady	18346	1384	668	79	4670	1
Up	1067	110	24	6	327	0

	glipizide	glyburide	tolbutamide	pioglitazone	rosiglitazone	acarbose

Down	560	564	0	118	87	3
No	89080	91116	101743	94438	95401	101458
Steady	11356	9274	23	6976	6100	295
Up	770	812	0	234	178	10

	miglitol	troglitazone	tolazamide	examide	citoglipton	insulin
Down	5	0	0	0	0	12218
No	101728	101763	101727	101766	101766	47383
Steady	31	3	38	0	0	30849
Up	2	0	1	0	0	11316

	glyburide-metformin	glipizide-metformin	glimepiride-pioglitazone	metformin-rosiglitazone	metformin-pioglitazone
Down	6	0	0	0	0
No	101060	101753	101765	101764	101765
Steady	692	13	1	2	1
Up	8	0	0	0	0

1.3.3 Summary

In conclusion, The processing methods of categorical attributes are summarized as follows.

Feature	Process result	Classification
Encounter_id	Drop	/
Patient_nbr	Drop	/

Race	Original 5 classes	race_AfricanAmerica, race_Asian, race_Caucasian, race_Hispanic race_Other
Gender	Original 3 classes	gender_Female gender_Male gender_Unknown/Invalid
age	Original 10 classes	[0-10], [10-20], [20-30], [30-40], [40-50], [50-60], [60-70], [70-80], [80-90], [90-100]
admission_type_id	Compress original 8 classes into 4 classes	admission_type_Emergency admission_type_Urgent admission_type_Elective admission_type_others
discharge_disposition_id	Compress original 26 classes into 2 classes	Home others
admission_source	Compress original 17 classes into 3 classes	admission_source_emergencyroom admission_source_physician_referral admission_source_others

diag_1 & diag_2 & diag_3	Compress original 19 classes into 9 classes	Circulatory Respiratory Digestive Injury Musculoskeletal Genitourinary Neoplasms Diabetes others
max_glu_serum	Original 4 classes	max_glu_serum_>200 max_glu_serum_>300 max_glu_serum_None max_glu_serum_Norm
A1Cresult	Original 4 classes	A1Cresult_>7 A1Cresult_>8 A1Cresult_None A1Cresult_Norm
Metformin, repaglinide, nateglinide, glimepiride, glipizide, glyburide, pioglitazone, rosiglitazone,	Original 4 classes	Down, No, Steady, Up

insulin		
change	Original 2 classes	change_Ch change_No
diabetesMed	Original 2 classes	diabetesMed_No diabetesMed_Yes
readmitted	Compress original 3 classes into 2 classes	Yes no

1.3.4 Reassigning categorical variables——One Hot Encode

We binarize the categorical input so that they can be thought of as a vector from the Euclidean space. We call this as embedding the vector in the Euclidean space. Because many algorithms for classification/regression/clustering etc. requires computing distances between features or similarities between features. And many definitions of distances and similarities are defined over features in Euclidean space. So, we would like our features to lie in the Euclidean space as well.

Dummy coding is a classic way to transform nominal into numerical values and a system to code categorical predictors in a regression analysis. A system to code categorical predictors in a regression analysis in the context of the general linear model. We can't put categorical predictors such as character variable, or a string variable into a regression analysis function. We need to make it a numeric variable in some way. That's where dummy coding comes in. It allows to look at categorical predictors in the same model as continuous predictors and put them together in moderation analyses. Featurizing via a one-hot-encoding representation lead to a very large feature vector. To reduce the dimensionality of the feature space, feature hashing is generally used.

Let us take an example of the dataset with just one feature (max_glu_serum) and it takes four values >200, >300, None, Norm. Now, let us take four feature vectors >200 = (1), >300 = (2), None = (3), Norm = (4). The euclidean distance between them are

d(>200, >300) = 1, d(>300, None) = 1, d(>200, None) = 2. This shows that distance between >200 and >300 is smaller than >200 and None. So we need to make it fair to assume that all categorical features are equally far away from each other. Then we binary the same feature vectors: >200 = (1, 0, 0, 0), >300 = (0, 1, 0, 0), None = (0, 0, 1, 0), Norm = (0, 0, 0, 1). When we binarize the input, we implicitly state that all values of the categorical features are equally away from each other.

1.4 Feature Filter

First, We use filter method to select features, and choose PCC as the feature ranking criteria. Here's our PCC output.

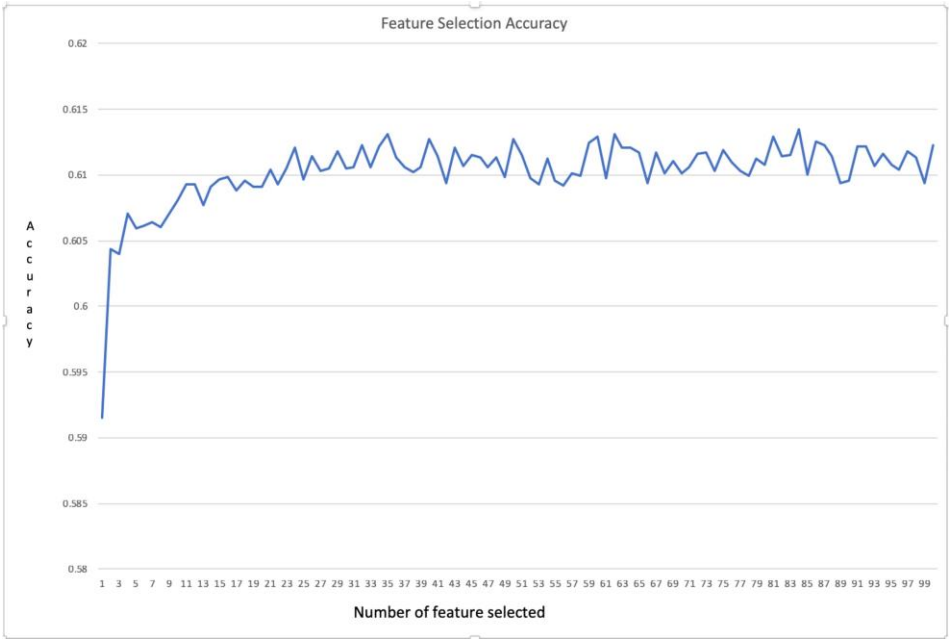
feature	|r| values	header
23	0.166258	number_inpatient
13	0.074919	discharge_disposition_id
22	0.060590	number_emergency
51	0.049212	number_diagnoses
17	0.044735	time_in_hospital
20	0.040039	num_medications
93	0.034867	insulin_No
92	0.032427	insulin_Down
98	0.025826	diabetesMed_No
99	0.025826	diabetesMed_Yes
61	0.025552	metformin_No
62	0.024047	metformin_Steady
18	0.022992	num_lab_procedures
95	0.021640	insulin_Up
39	0.020831	diag_2_Neoplasms
29	0.020343	diag_3_Genitourinary
49	0.019959	diag_1_Diabetes
14	0.019720	admission_source_emergencyroom
21	0.018591	number_outpatient
58	0.018512	A1Cresult_None
96	0.018252	change_Ch
97	0.018252	change_No
43	0.017857	diag_1_Respiratory
15	0.017675	admission_source_physician_referral
8	0.016773	age
30	0.014845	diag_3_Neoplasms
11	0.013216	admission_type_Elective
53	0.013099	max_glu_serum_>300
9	0.013047	admission_type_Emergency
46	0.012554	diag_1_Musculoskeletal
...	...	...

70	0.001404	nateglinide_Steady
78	0.001375	glipizide_Steady
7	0.001159	gender_Unknown/Invalid
91	0.001083	rosiglitazone_Up
10	0.000959	admission_type_Urgent
35	0.000949	diag_2_Digestive
0	0.000847	race_AfricanAmerican
68	0.000799	nateglinide_Down
69	0.000631	nateglinide_No
75	0.000459	glimepiride_Up
55	0.000428	max_glu_serum_Norm
41	0.000137	diag_2_others
94	0.000097	insulin_Steady

The sequence of index of features:

[23, 13, 22, 51, 17, 20, 93, 92, 98, 99, 61, 62, 18, 95, 39, 29, 49, 14, 21, 58, 96, 97, 43, 15, 8, 30, 11, 53, 9, 46, 59, 31, 19, 63, 57, 54, 76, 25, 74, 45, 24, 56, 73, 86, 65, 67, 89, 85, 81, 90, 4, 66, 52, 42, 32, 82, 34, 12, 44, 37, 40, 88, 16, 79, 38, 48, 80, 26, 3, 84, 50, 71, 64, 2, 47, 36, 28, 33, 1, 5, 6, 72, 77, 27, 83, 87, 60, 70, 78, 7, 91, 10, 35, 0, 68, 69, 75, 55, 41, 94]

And then we use random forest as our algorithm to select feature subset which realizes the highest accuracy.



From output, we find that the highest accuracy is 0.61349541337841, and the number of feature selected is 84. So, finally, we use the first 84 features which sorted by |r| values as our dataset.

2.The choose of criterion--roc_auc

For our project we cannot only pursue the high of accuracy of the model, since if we we get a model only performance well on class 0, although the accuracy may over 70 percent, the model is not good. So we need to find a better criterion. At last we choose roc_auc as our criterion, since it will take the accuracy both for class 0 and class 1 both into consideration.

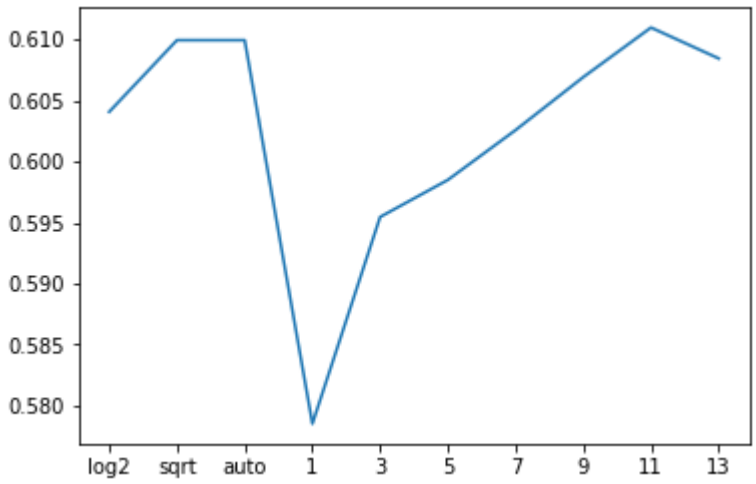
2.1 Random Forest

For random forest by checking the internet, we find that the most important are n_estimators ,criterion, max_features, max_depth.As for criterion, after do the cross validation, we found that there are not much difference between ‘gini and entropy, so we choose gini as our critition.

For the rest parameters, we do the loop to find the best parameters one by one. And here is the result for each parameter.

2.1.1 For tuning parameters

a.max_features-----The number of features to consider when looking for the best split:



The average roc_auc is 0.6040675535265508

The average roc_auc is 0.609933589834305

The average roc_auc is 0.609933589834305

The average roc_auc is 0.5785222959791587

The average roc_auc is 0.5954541010162115

The average roc_auc is 0.5984895059512594

The average roc_auc is 0.6025702404962981

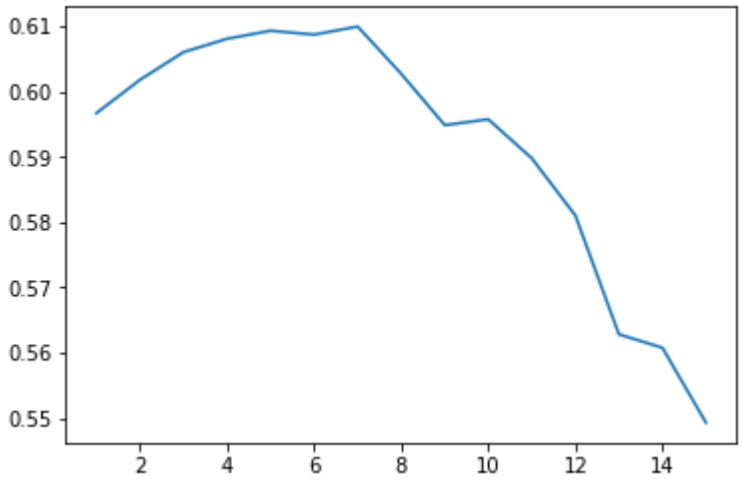
The average roc_auc is 0.6069094577697429

The average roc_auc is 0.610966455072594

The average roc_auc is 0.6084354213259868

We can see that the best one is max_features is a int type, and the optimal number is 11.

b.max_depth----The maximum depth of the tree.



The average roc_auc is 0.59669239629248

The average roc_auc is 0.6017908234969538

The average roc_auc is 0.6060224911355867

The average roc_auc is 0.6080613677099098

The average roc_auc is 0.6092967037423748

The average roc_auc is 0.6087033170194739

The average roc_auc is 0.609933589834305

The average roc_auc is 0.6027027644790185

The average roc_auc is 0.5948484281755485

The average roc_auc is 0.5957421830633024

The average roc_auc is 0.5897823087457171

The average roc_auc is 0.5810254565806139

The average roc_auc is 0.5628513979430204

The average roc_auc is 0.5607849325666833

The average roc_auc is 0.5493288864524846

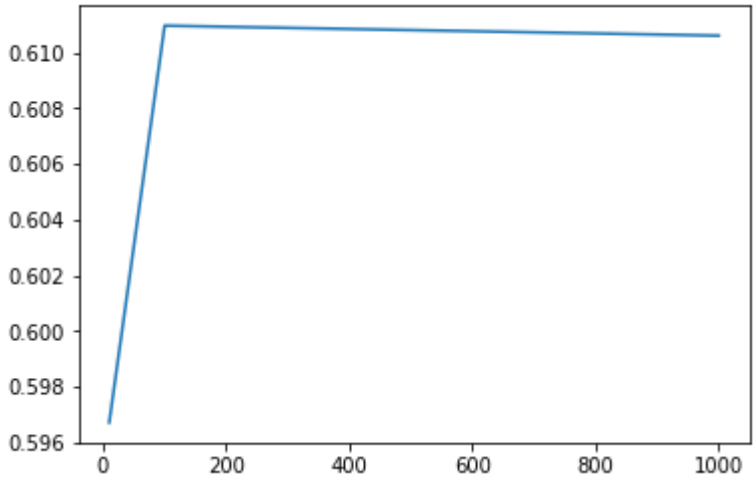
The average roc_auc is 0.502648289259674

We can see that the best maximum depth i is 7.

b.n_estimators ---The number of trees in the forest.

For this parameters, the reasonable interval is too large, so we first find on n=10,100,1000.

The result:

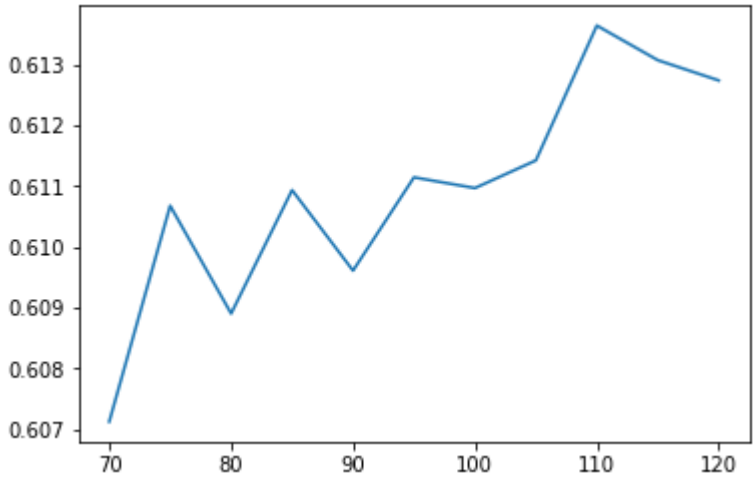


The average roc_auc is 0.5967162487804154

The average roc_auc is 0.610966455072594

The average roc_auc is 0.6106076906218995

We can infer that the best n is around 100, so we do another loop in which we find n around 100.



The average roc_auc is 0.607125321325873

The average roc_auc is 0.6106760361682881

The average roc_auc is 0.6089031808295426

The average roc_auc is 0.6109329687938049

The average roc_auc is 0.6096090571469073

The average roc_auc is 0.6111426746358761

The average roc_auc is 0.610966455072594

The average roc_auc is 0.611421304265527

The average roc_auc is 0.6136382165274898

The average roc_auc is 0.6130692429434768

The average roc_auc is 0.6127361072547869

So we can see the best estimator number is 110

As shown, after use train data to do the cross validation(cv=5), we can get the best parameters

for random forest is(rf_max_feature=11,max_depth=7,n_estimators=110)

2.1.2 Fix the model and get the test performance

After find the best parameters, we train the model and do the prediction.And the result of random forest is:

The test roc_auc is 0.6172538817761726

The confusion matrix:

	0	1
0	18298	9868
1	1519	2140

By calculating, we can get:

	precision	recall	f1-score	support
0	0.92	0.65	0.76	28166
1	0.18	0.58	0.27	3659

As we can see the performance on class 0 is better, however the class 1 is more important, so we change the threshold to improve the performance of the class 1.

So we set the minimum recall of class 1, and output the result of corresponding threshold:

minimum_sensitive(recall)	Thresholds:	True positive rate	True negative rate
Sens >= 0.7	0.47484	0.7001913091008473	0.52123
Sens >= 0.75	0.46248	0.75020	0.45494
Sens >= 0.8	0.45080	0.80021	0.38883

Since we can not make the true negative rate under 0.5, so we can put the threshold as 0.47484.

After we change the threshold, the result is:

The test roc_auc is 0.6156603152547162

The confusion matrix:

	0	1

0	15260	12906
1	1136	2523

By calculating, we can get:

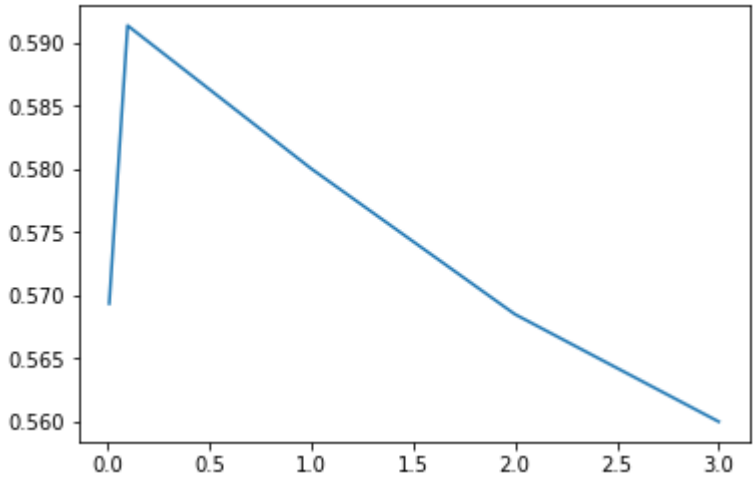
	precision	recall	f1-score	support
0	0.93	0.54	0.68	28166
1	0.16	0.70	0.26	3659

2.2 SVM

For Support vector machines, we chose C-Support Vector Classification.And we mainly do the cross validation to find parameter C and the kind of kernel.

2.2.1 For finding the best parameters:

a.C-----Penalty parameter C of the error term.:



The average roc_auc is 0.5693393207015005

The average roc_auc is 0.5913327638031567

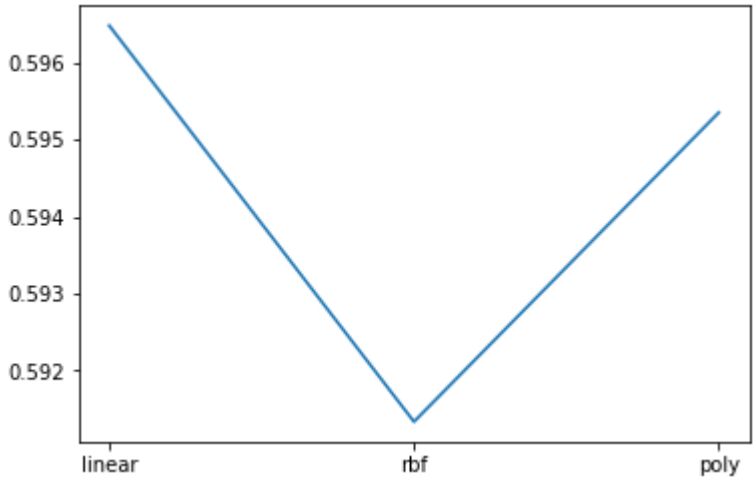
The average roc_auc is 0.5800261687527953

The average roc_auc is 0.5685016922324255

The average roc_auc is 0.5600033083812508

We can see that the best C is 0.1

b.kernel



The average roc_auc is 0.5964820010661028

The average roc_auc is 0.5913327638031567

The average roc_auc is 0.5953510779127877

So,the best kernel is linear kernel.

As shown, after use train data to do the cross validation(cv=5), we can get the best parameters

SVC(C=0.1, kernel='linear',probability=True).

2.2.2 Fix the model and get the test performance.

After find the best parameters, we train the model and do the prediction.

And the result of svc is:

The test roc_auc is 0.6003765702328893

The confusion matrix:

	0	1
0	18933	9233
1	1725	1934

By calculating, we can get:

	precision	recall	f1-score	support
0	0.92	0.67	0.78	28166
1	0.17	0.53	0.26	3659

As we can see the performance on class 0 is better, however the class 1 is more important, so we change the threshold to improve the performance of the class 1.

So we set the minimum recall of class 1, and output the result of corresponding threshold:

minimum_sensitive(recall)	Thresholds:	True positive rate	True negative rate
Sens >= 0.7	0.4199155	0.700191	0.522793
Sens >= 0.75	0.402392	0.7502049	0.454874
Sens >= 0.8	0.3874681	0.80021	0.379109

Since we can not make the true negative rate under 0.5, so we can put the threshold as 0.47484.

After we change the threshold, the result is:

The test roc_auc is 0.611391228440563

The confusion matrix:

	0	1
0	14727	13439
1	1098	2561

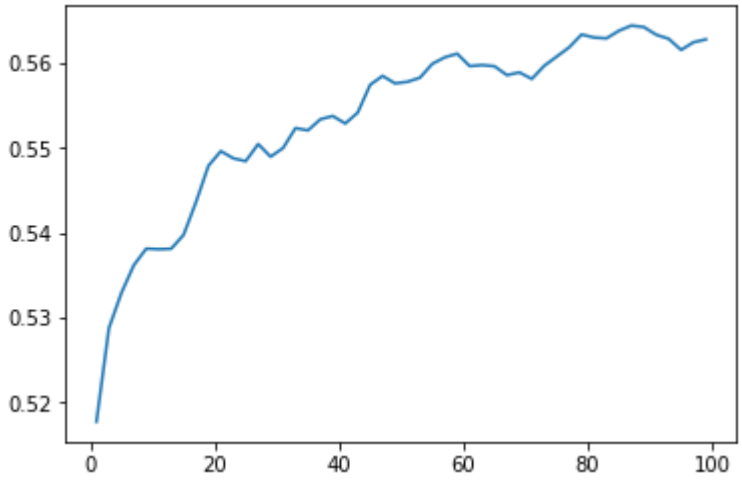
By calculating, we can get:

	precision	recall	f1-score	support
0	0.93	0.52	0.67	28166
1	0.16	0.70	0.26	3659

2.3 KNN

For knn we only need to find the best parameter k. To find the best k we randomly split the dataset and use 5-fold cross validation in the train data to get the roc_auc. The following picture is the roc_auc number when we choose different k. We did the knn from k=1 to k=101.

2.3.1 For finding the best parameters:



When k=87 we got the biggest roc-auc. And the final roc-auc $\approx$ 0.563

2.3.2 Fix the model and get the test performance

After find the best parameters, we train the model and do the prediction.And the result of knn:

The test roc_auc is 0.5591533072666816

The confusion matrix:

	0	1
0	16988	11178
1	1423	2236

By calculating, we can get:

	precision	recall	f1-score	support
0	0.91	0.60	0.72	28166
1	0.14	0.52	0.23	3659

As we can see the performance on class 0 is better, however the class 1 is more important, so we change the threshold to improve the performance of the class 1.

So we set the minimum recall of class 1, and output the result of corresponding threshold:

minimum_sensitive(recall)	Thresholds:	True positive rate	True negative rate
Sens >= 0.7	0.4252	0.7157	0.3922

Sens >= 0.75	0.4023	0.7627	0.3359
Sens >= 0.8	0.3793	0.8136	0.2830

Since we can not make the true negative rate under 0.5, so we can put the threshold as 0.47484.

After we change the threshold, the result is:

The test roc_auc is 0.5606782628665563

The confusion matrix:

	0	1
0	14372	13794
1	1423	2236

By calculating, we can get:

	precision	recall	f1-score	support
0	0.91	0.51	0.65	28166
1	0.14	0.61	0.23	3659

2.4 Logistic Regression

For logistic regression we finally decided to adjust the parameter C and penalty. We tested the different C from 0.01 to 100. And for penalty we separately used l1 and l2 . But the results differs a little.

The parameter C stands for Inverse of regularization strength

The parameter penalty is used to specify the norm used in the penalization

2.4.1 For finding the best parameters:

L1 penalty

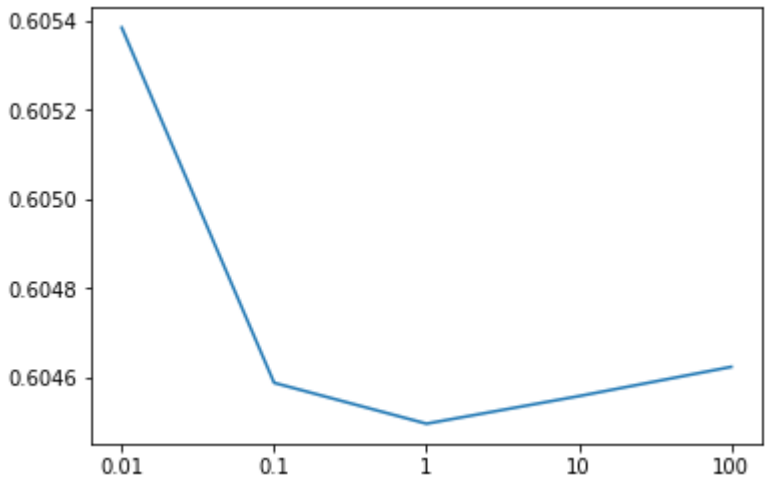
When C= 0.01 The average roc_auc is 0.6053854654178745

When C= 0.1 The average roc_auc is 0.6045871813632608

When C= 1 The average roc_auc is 0.6044953869947962

When C= 10 The average roc_auc is 0.6045573047158859

When C= 100 The average roc_auc is 0.6046230494056055



L2 penalty

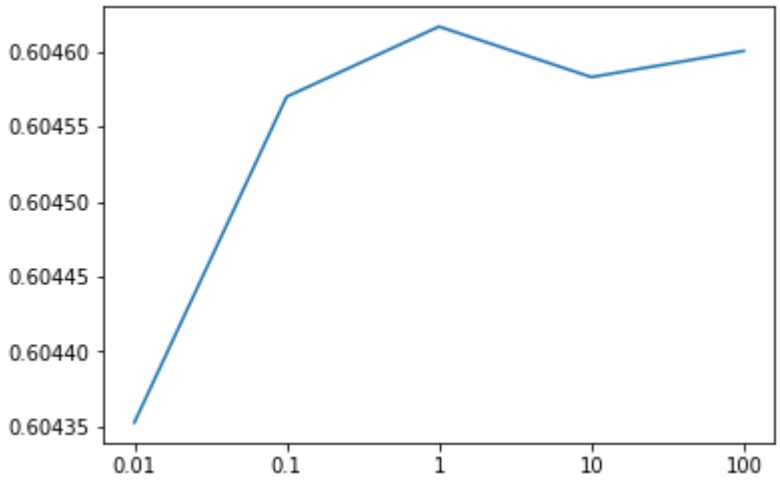
When C= 0.01 The average roc_auc is 0.6043525959554963

When C= 0.1 The average roc_auc is 0.6045697210599152

When C= 1 The average roc_auc is 0.6046164911699975

When C= 10 The average roc_auc is 0.604582830232667

When C= 100 The average roc_auc is 0.6046002837744066



Finally we choose L1 penalty and C=0.01 as our final parameter to fit in the whole model.

2.4.2 Fix the model and get the test performance.

After find the best parameters, we train the model and do the prediction.And the result of logistic regression:

The test roc_auc is 0.6124983424606592

The confusion matrix:

	0	1
0	18746	9420
1	1612	2047

By calculating, we can get:

	precision	recall	f1-score	support
0	0.92	0.67	0.77	28166
1	0.18	0.56	0.27	3659

As we can see the performance on class 0 is better, however the class 1 is more important, so we change the threshold to improve the performance of the class 1.

So we set the minimum recall of class 1, and output the result of corresponding

threshold:

minimum_sensitive(recall)	Thresholds:	True positive rate	True negative rate
Sens >= 0.7	0.45324	0.7	0.5246
Sens >= 0.75	0.4327	0.7502	0.4509
Sens >= 0.8	0.4153	0.8	0.3856

Since we can not make the true negative rate under 0.5, so we can put the threshold as 0.47484.

After we change the threshold, the result is:

The test roc_auc is 0.6124154727709733

The confusion matrix:

	0	1
0	15260	12906
1	1136	2523

By calculating, we can get:

	precision	recall	f1-score	support
0	0.93	0.52	0.67	28166
1	0.16	0.70	0.26	3659

3.The Ensemble

After we do the three models, we can see algorithm svc, random forest, logistic regression has a better performance. So we do the ensemble on these three algorithm, and we do it two times both on the algorithm with and without changing threshold. And we use majority voting to do that.

(1)without change the threshold.

he test roc_auc is 0.6181428157825186

The confusion matrix:

	0	1
0	17586	10580
1	1420	2239

By calculating, we can get:

	precision	recall	f1-score	support

0	0.93	0.62	0.75	28166
1	0.17	0.61	0.27	3659

(2） change the threshold.

The test roc_auc is 0.6170309666288161

The confusion matrix:

	0	1
0	15037	13129
1	1097	2562

By calculating, we can get:

	precision	recall	f1-score	support
0	0.93	0.53	0.68	28166
1	0.16	0.70	0.26	3659

4. Conclusion

After we conclude the the results of four models and two kinds of ensembles, we have the result :

With every medel maintain the threshold of 0.5

	RF	SVM	KNN	LR	ENSEMBLE
0	0.65	0.67	0.60	0.67	0.62
1	0.58	0.53	0.52	0.56	0.61
roc_auc	0.61725	0.60037	0.559	0.612	0.61703

With every medel change the threshold and do the ensemble

	RF	SVM	KNN	LR	ENSEMBLE
0	0.54	0.52	0.51	0.52	0.53
1	0.70	0.70	0.61	0.70	0.70
roc_auc	0.61566	0.611	0.56	0.612	0.61814