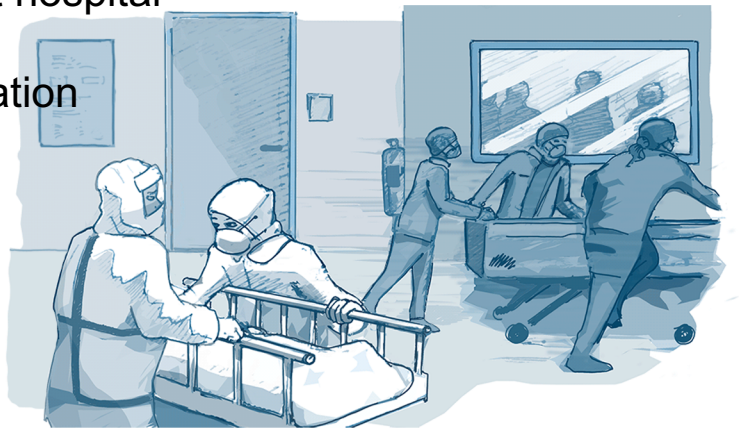
Prediction of Coronavirus Patients' Length of Stay in Hospital

Jack Lin, Jianing Pei, Qixuan Chen

Dataset: Coronavirus Healthcare

- X: 15 features including: severity of illness, type of admission, etc.
- Y: Patients Length of Stay at Hospital (Divided into 11 Categories:

  0-10, 11-20,…,91-100, above 100 days)
- Classification Task: predict patients' length of stay at hospital
- Motivation: Promote efficient medical resource allocation



Dataset: Coronavirus Healthcare

Type of Admission	Severity of Illness	Visitors with Patient	Age	Admission_Deposit	Stay
Emergency	Extreme	2	51-60	4911	0-10
Trauma	Extreme	2	51-60	5954	41-50
Trauma	Extreme	2	51-60	4745	31-40
Trauma	Extreme	2	51-60	7272	41-50
Trauma	Extreme	2	51-60	5558	41-50
Trauma	Extreme	2	51-60	4449	11-20
Emergency	Extreme	2	51-60	6167	0-10
Trauma	Extreme	2	51-60	5571	41-50
Trauma	Extreme	2	51-60	7223	51-60
Trauma	Extreme	2	51-60	6056	31-40
Urgent	Extreme	2	51-60	5797	21-30
Urgent	Extreme	2	51-60	5993	11-20
Emergency	Extreme	2	51-60	5141	0-10
Urgent	Extreme	2	51-60	8477	21-30

Method: Classification Pipeline

Label Encoding

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

One Hot Encoding

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

- Preprocessing:
  - Feature Encoding (**one-hot encoding** or **simple label encoding**)
  - Principal Component Analysis (>= 85 exaplained variance)
- Classification Models: KNN, SVM, Random Forest
  - Hyperparamter Tunning (80% Training, 10% Validation, 10% Test)
- Modle Comparison

K-nearest Neighbors

(implemented with python sklearn)

Preprocessing

Normalization (**Min-max normalization**)

$$y_i = \frac{x_i - \min_{1 \le j \le n}\{x_j\}}{\max_{1 \le j \le n}\{x_j\} - \min_{1 \le j \le n}\{x_j\}}$$

make the data fall between 0 and 1

Preprocessing

One-hot encoding

- categorical features→0 and 1
- 11 features→23 features

Hospital_region_Z	Hospital_region_X	Hospital_region_Y
1	0	0
1	0	0
0	1	0
0	0	1
0	0	1

Simple encoding

- categorical features→1, 2, 3……
- 11 features→ 11 features

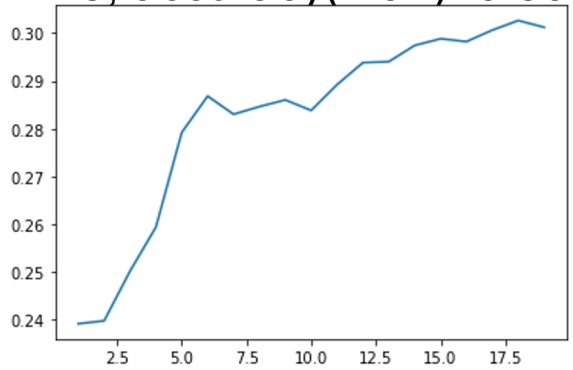
Hospital_region_code
1
1
0
0.5
0.5

the result of the different ways of encoding

when K is between (1,20)

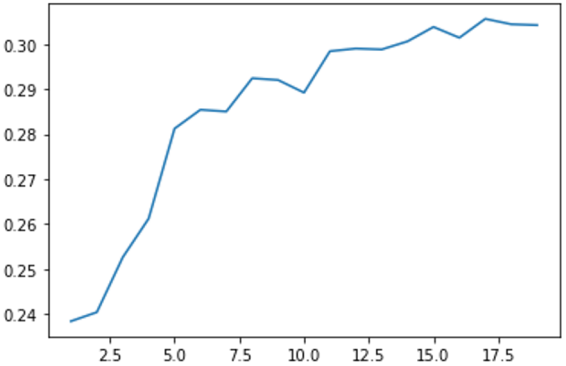
one-hot ecodoing

K=18, accuracy(max)=**0.3026**



simple encoding

K=17, accuracy(max)=**0.3056**



x-axis: K y-axis: accuracy

Hyperparameters

1. The distance measure

- Manhattan distance
- Euclidean Metric

1. Different ways of processing

- change features according to correlation

1. Number of neighbors: **K**

Preprocessing

Correlation

the correlation of each features with stay

(from the lowest to the highest)

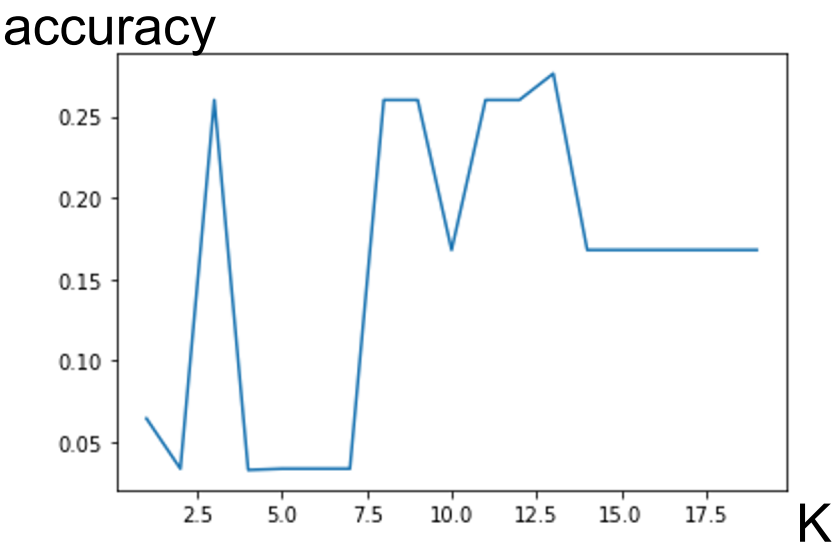
Features	Stay
Patientid	-0.001906
City_Code_Patient	-0.001906
City_Code_Hospital	-0.004931
Ward_Facility_Code	-0.007634
Department	0.018589
Bed Grade	0.026279
Hospital_region_code	0.027808
Ward_Type	0.032883
Hospital_code	0.039903
Age	0.043340
Hospital_type_code	0.059653
Admission_Deposit	-0.064331
Type of Admission	0.072283
Severity of Illness	0.105599
Available Extra Rooms in Hospital	-0.119464
Visitors with Patient	0.554307
Stay	1.000000

Distance measure

Manhattan distance

$$|x_1 - x_2| + |y_1 - y_2|$$

- dataset: 50,000
- K(1,20)



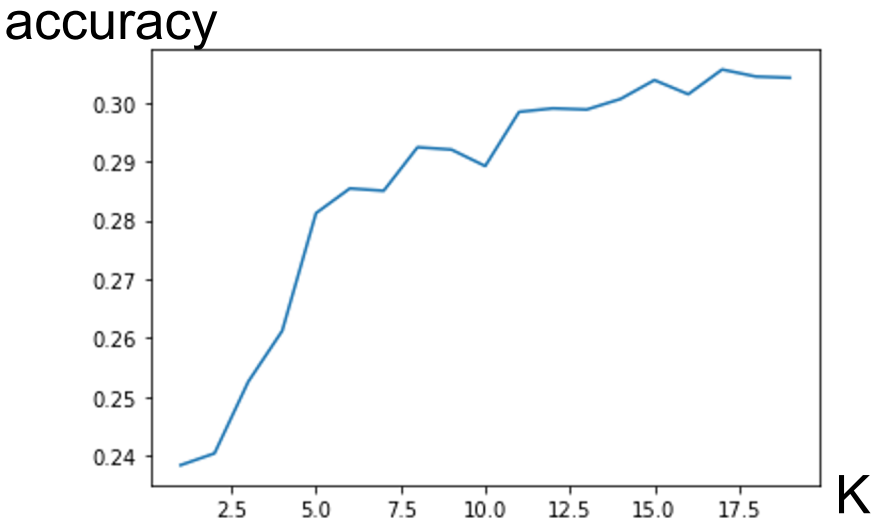
K=13 accuracy(max)=**0.2762**

Distance measure

Euclidean Metric

$$\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

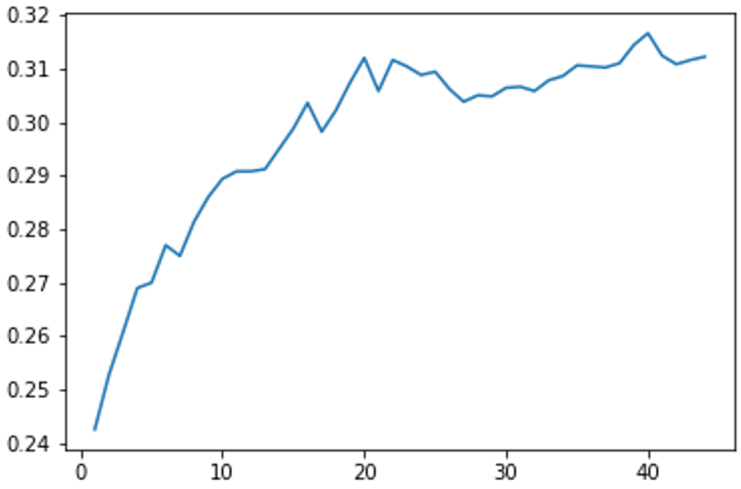
- dataset: 50,000
- K(1,20)



K=17 accuracy(max)=**0.3056**

Change of Features——17 features

accuracy



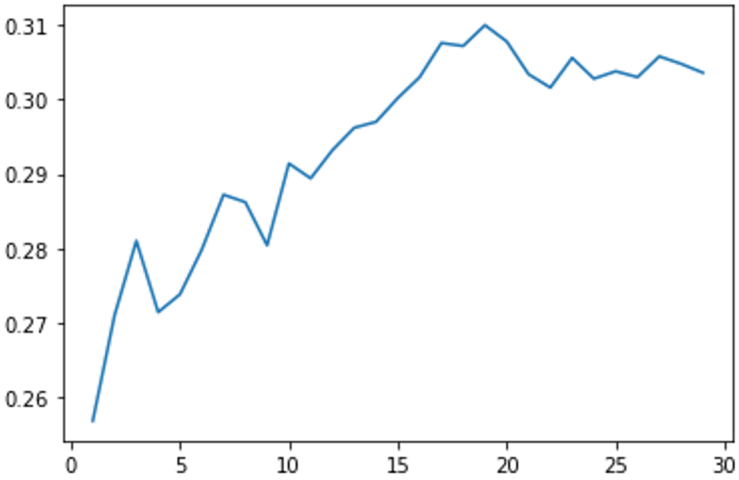
K

K=40 accuracy(max)=**0.3166**

Patientid
City_Code_Patient
City_Code_Hospital
Ward_Facility_Code
Department
Bed Grade
Hospital_region_code
Ward_Type
Hospital_code
Age
Hospital_type_code
Admission_Deposit
Type of Admission
Severity of Illness
Available Extra Rooms in Hospital
Visitors with Patient

Change of Features——12 features(corr>0.01)

accuracy



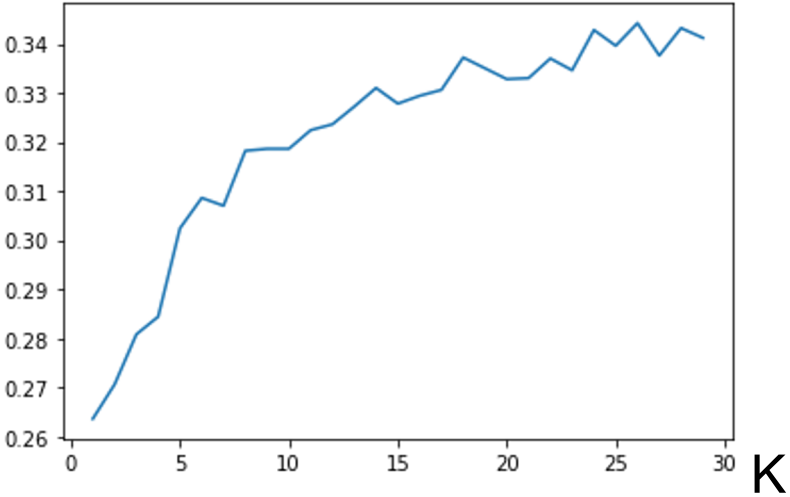
K

K=19 accuracy(max)=**0.31**

Department	0.018589
Bed Grade	0.026279
Hospital_region_code	0.027808
Ward_Type	0.032883
Hospital_code	0.039903
Age	0.043340
Hospital_type_code	0.059653
Admission_Deposit	-0.064331
Type of Admission	0.072283
Severity of Illness	0.105599
Available Extra Rooms in Hospital	-0.119464
Visitors with Patient	0.554307

Change of Features——6 features(corr>0.05)

accuracy

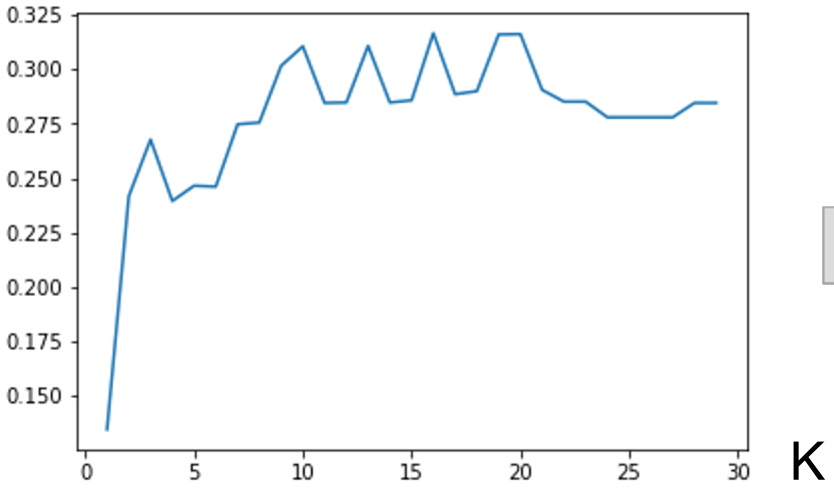


Hospital_type_code	0.059653
Admission_Deposit	-0.064331
Type of Admission	0.072283
Severity of Illness	0.105599
Available Extra Rooms in Hospital	-0.119464
Visitors with Patient	0.554307

K=26 accuracy(max)=**0.3442**

Change of Features——1 features(corr>0.5)

accuracy



Visitors with Patient	0.554307

K=16 accuracy(max)=**0.3166**

Final Model

Preprocessing

- simple encoding
- Min-max normalization
- Correlation

Hyperparameters

- Distance measure: Euclidean Metric
- 6 features(correlation>0.05)
- Number of neighbors: K=26

Accuracy:

- 0.3442

Discussion

1. only use part of the data(50,000 of 400,000)
2. the accuracy has a trend of increase with the increase of K
3. do not test the accuracy of all of the different combinations of features
4. the accuracy does not always increase with higher correlation features
5. make some assumptions(distance measure/ecoding measure)

Final Model

Preprocessing

- simple encoding
- Min-max normalization
- Correlation

Hyperparameters

- Distance measure: Euclidean Metric
- 6 features(correlation>0.05)
- Number of neighbors: K=26

Accuracy:

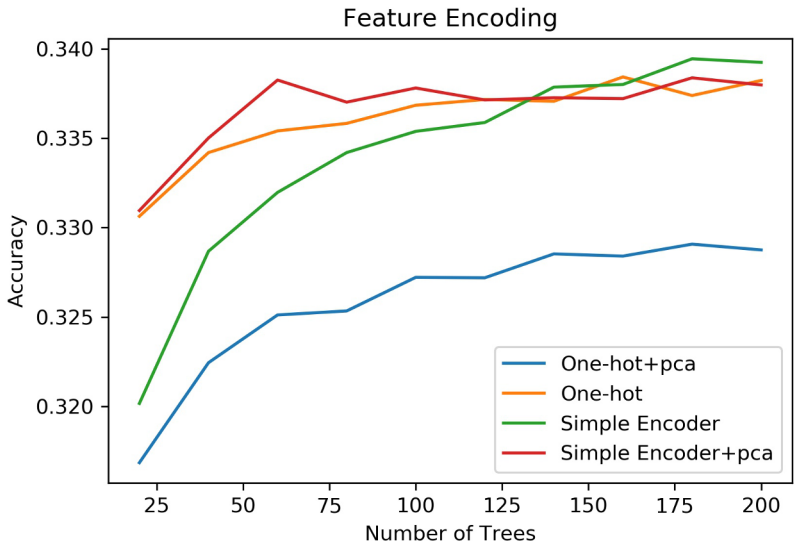
- 0.3442

Random Forest Classifer

(implemented with python sklearn)

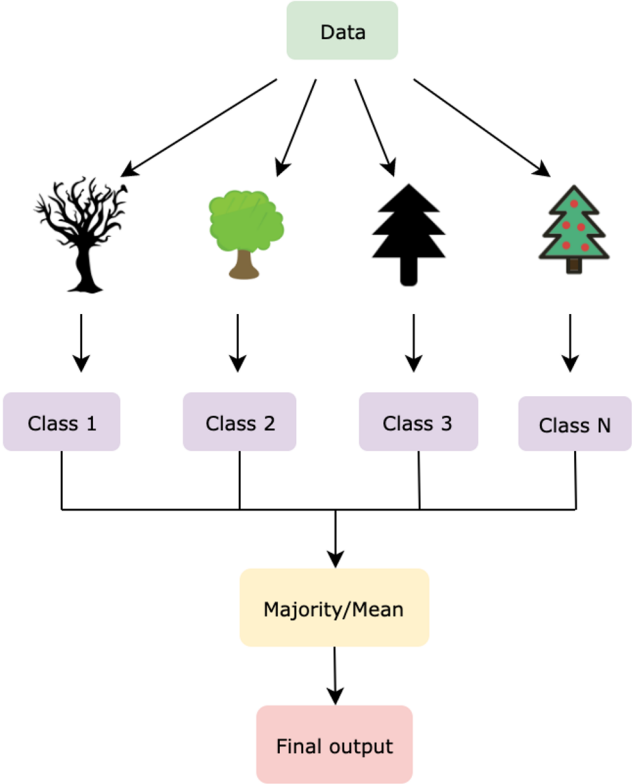
Feature Encoding for RF: Simple Encoder + PCA

- One-hot encoding
- One-hot encoding + PCA (70/127 features)
- Simple encoding
- Simple encoding + PCA (10/15 features)



Hyperparameters for Random Forest Classifier

- **n_estimators:** Number of Trees
- **max_depth:** Maximum number of levels in tree
- **min_samples_split:** Minimum number of samples required to split a node
- **min_samples_leaf:** Minimum number of samples required at each leaf node

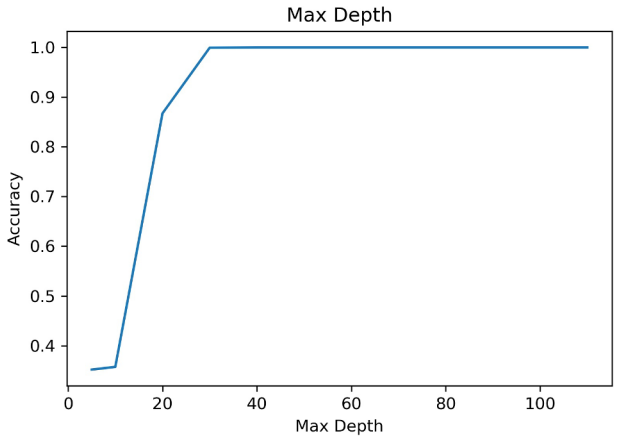


Hyperparameter: number of trees = 100

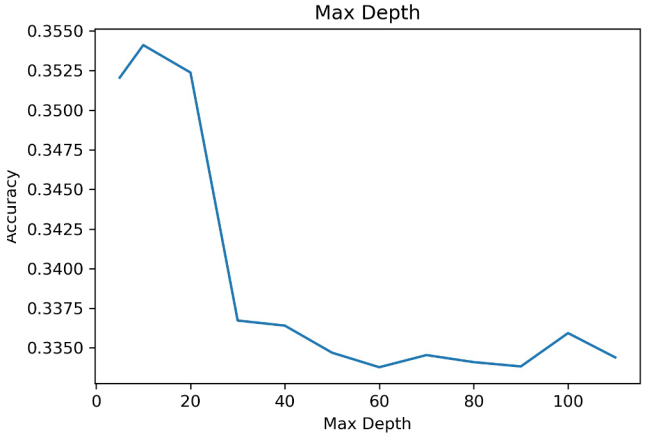
- **n_estimators:** Number of Trees
- range: [20, 40, 60, 80, 100, 120, 140, 160, 180, 200]

Hyperparameter: max depth = 10

- **max_depth:** Maximum number of levels in tree
- range: [5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110]



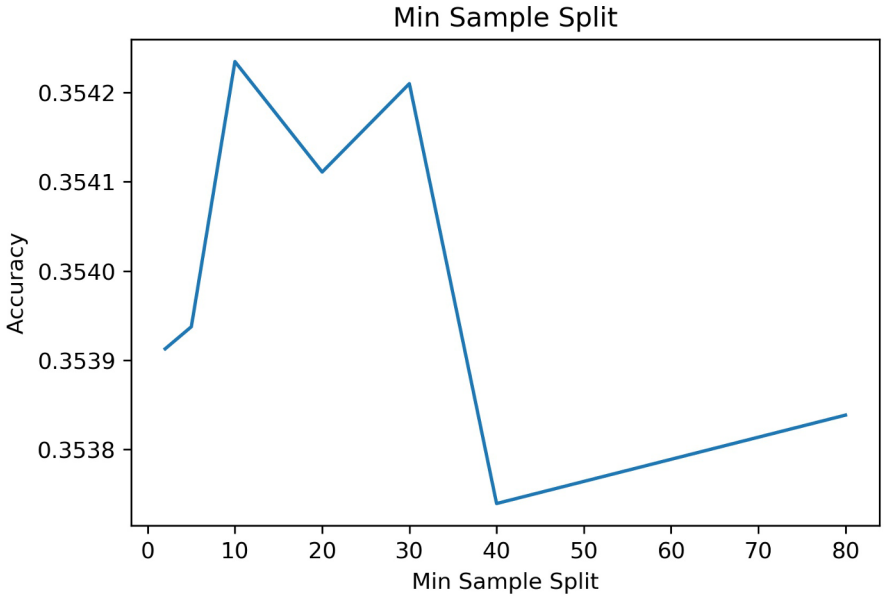
Trainning Set



Testing Set

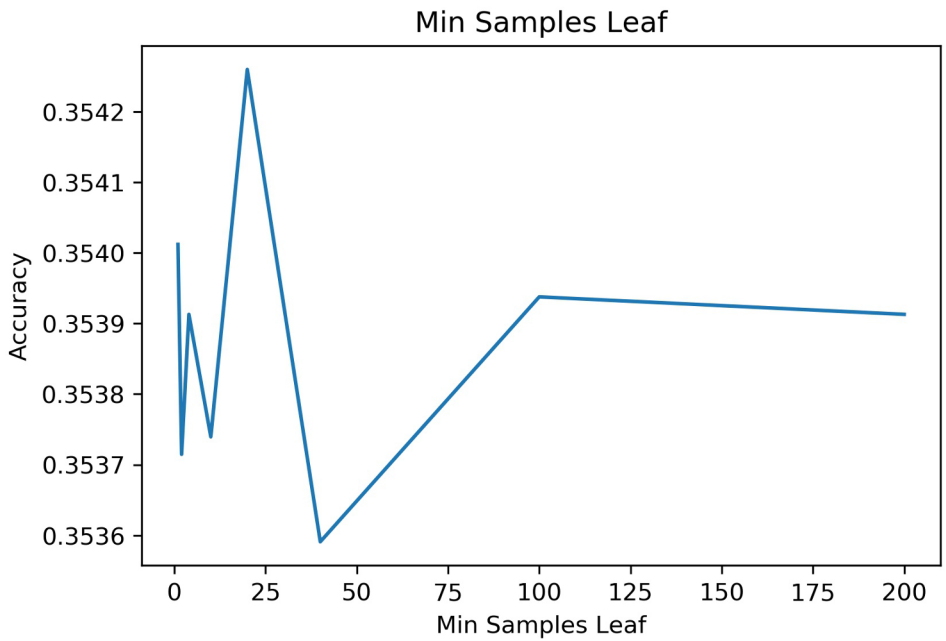
Hyperparameter: min sample split = 20

- **min_samples_split:** Minimum number of samples required to split a node
- range: [2, 5, 10, 20, 30, 40, 80]



Hyperparameter: minimum samples leaf = 20

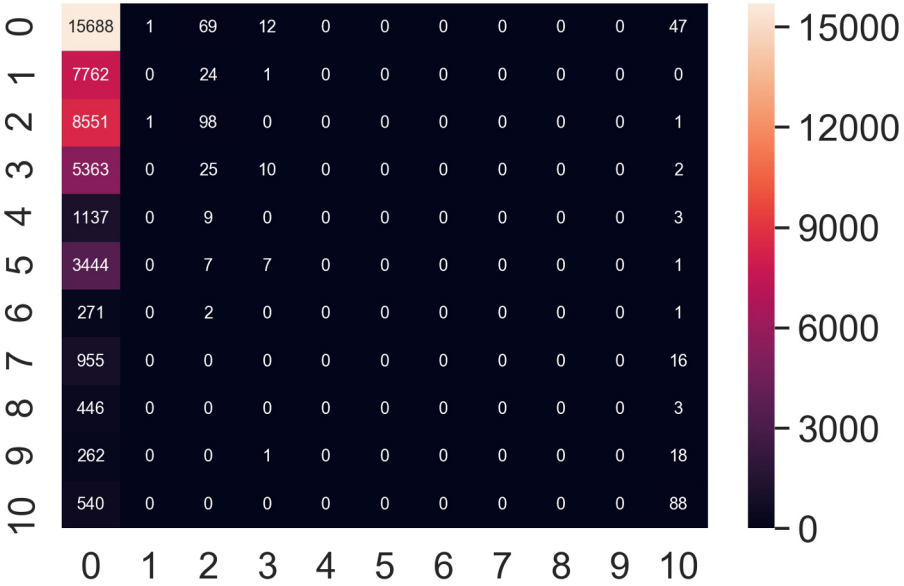
- **min_samples_leaf:** Minimum number of samples required at each leaf node
- range: [1, 2, 4, 10, 20, 40, 100, 200]



Results: Final Random Forest Model

- Preprocessing: simple encoding + PCA (10 components)
- Hyperparameters:
  - n_estimator = 100,
  - max depth = 10,
  - min sample split =20,
  - Minimum Samples Leaf =20
- Accuracy
  - Test Accuracy = 0.354
  - Train Accuracy = 0.356

Results: Confusion Matrix



	precision	recall	f1-score	support
class 0	0.35	0.99	0.52	15817
class 1	0.00	0.00	0.00	7787
class 2	0.42	0.01	0.02	8651
class 3	0.32	0.00	0.00	5400
class 4	0.00	0.00	0.00	1149
class 5	0.00	0.00	0.00	3459
class 6	0.00	0.00	0.00	274
class 7	0.00	0.00	0.00	971
class 8	0.00	0.00	0.00	449
class 9	0.00	0.00	0.00	281
class 10	0.49	0.14	0.22	628

Support Vector Machine

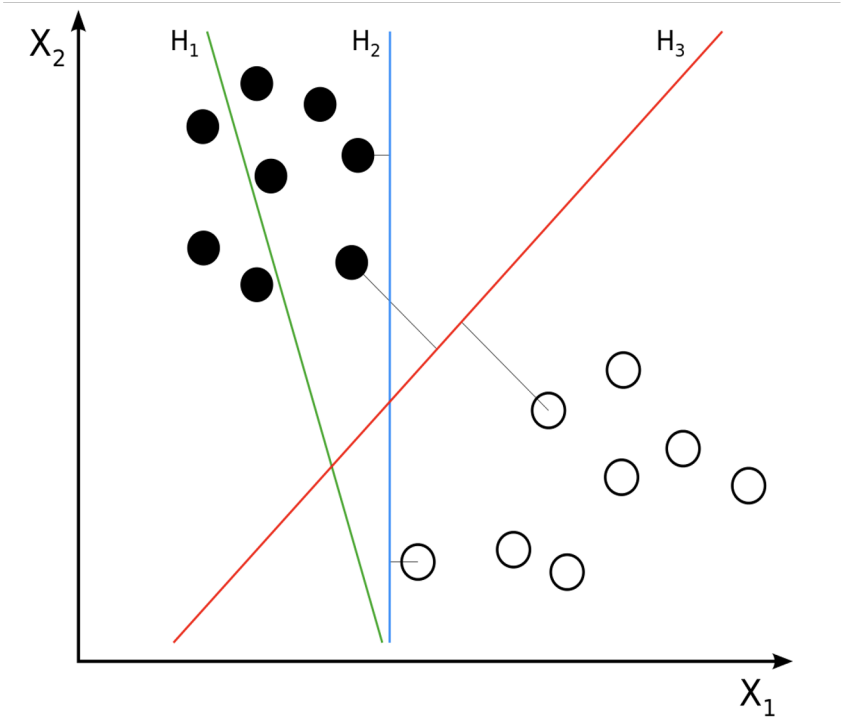
(implemented with python sklearn)

Basic Idea for SVM

Non-probabilistic binary linear clssifier

Separate such point with hyperplan

We need to choose the best hyperplan, which maximize the distance.



Nonlinear Multiclass SVM

Nonliner: mapping the input into high-dimension space.

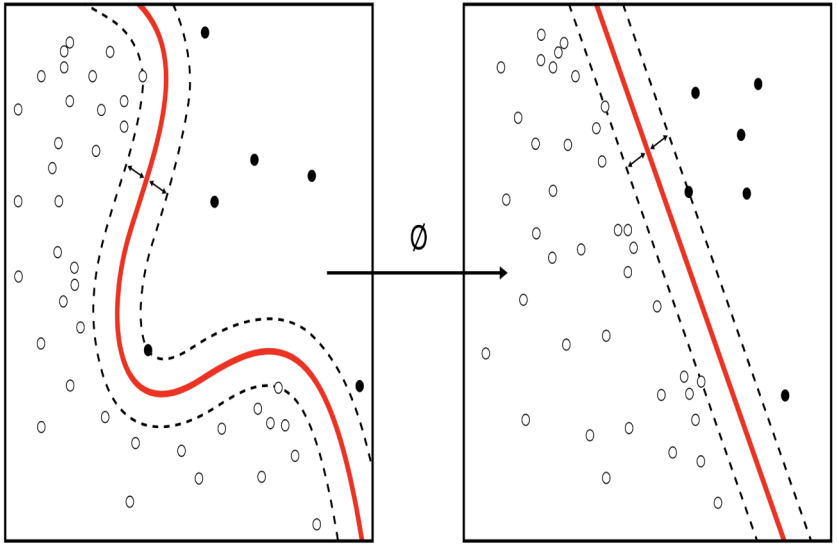
kernel function

1 linear: <x,x'>

2 polynomial: (γ<x,x'>+r)^d

3 rbf: exp(-γ||x-x'||^2)

4 sigmoid: tanh(γ<x,x'>+r)



Nonlinear Multiclass SVM

Multiclass svm:

single multiclass problem $\longrightarrow$ multiple binary classification problem

Two method:

1: one-versus-all

2: one-versus-one

Preprocessing

1 Drop the missing value

Train dataset:1.4% test dataset: 1.5%

2 Feature Encoding

one-hot encoding: 116 features

simple encoding: 15 features

3 PCA (>=85 explained variance)

one-hot encoding: 68 features

ordinary encoding: 11 features

One-hot Encoding + Linear SVM

Train sample size is around 350000 (>100000) and test sample size is around 80000

	TRAIN	TEST
SIZE	358906	89752
# of features	68	68

Model: Linear SVM + One-Versus-One

Accuracy:0.35003119707638825

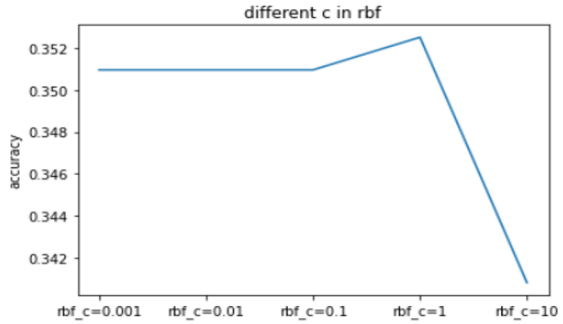
New dataset

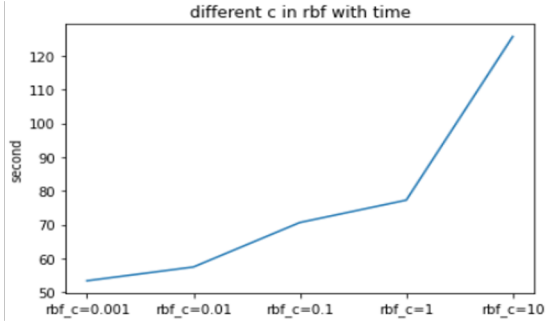
Ordinary Encoding + PCA (11 features)

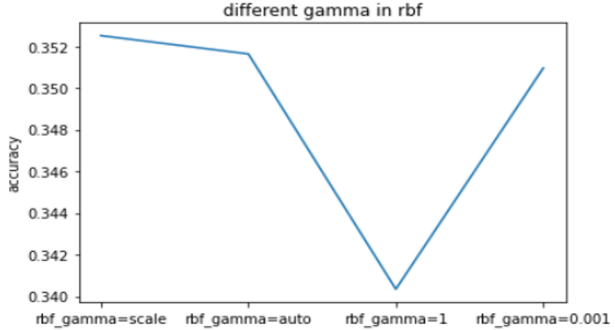
Train sample size + test sample size = 50000

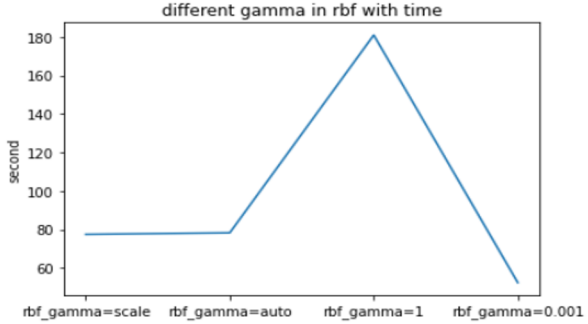
	Train	Test
Sample Size	40000	10000
# of Features	11	11

Gaussian radial basis function (rbf)



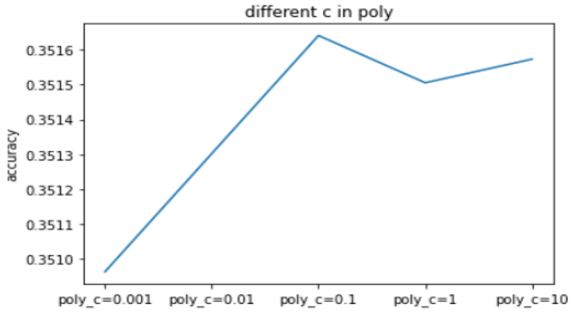


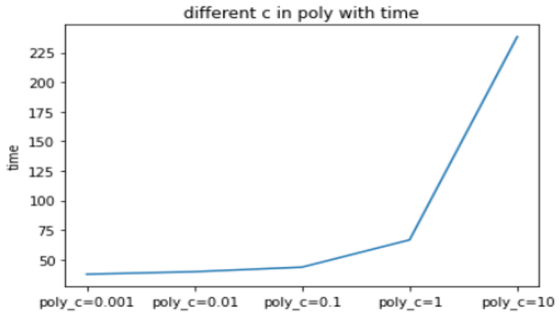


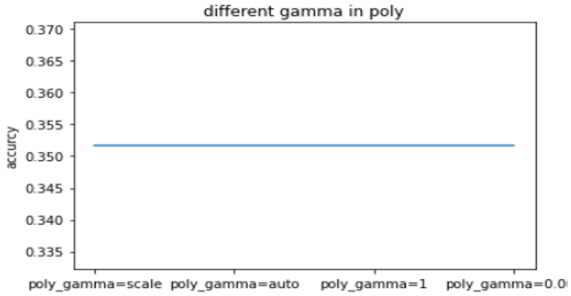


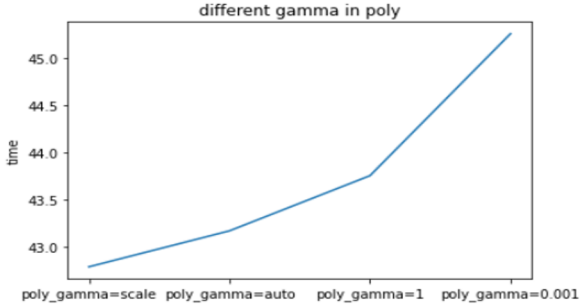
C	Gamma	Accuracy
1	Scale	0.3525

Polynomial



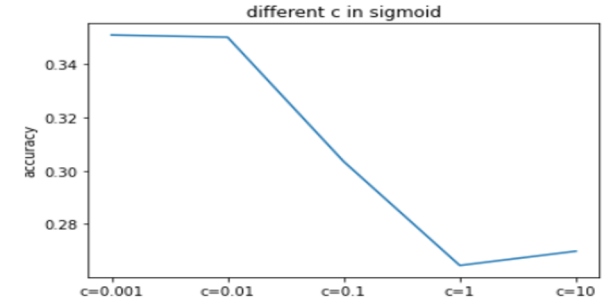


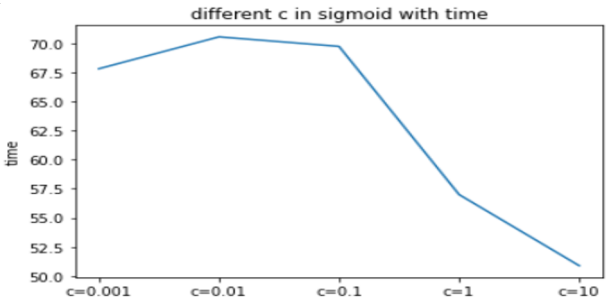


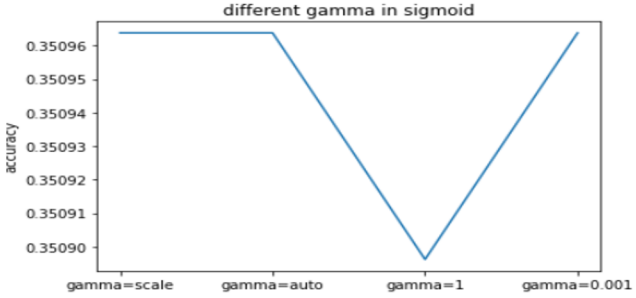


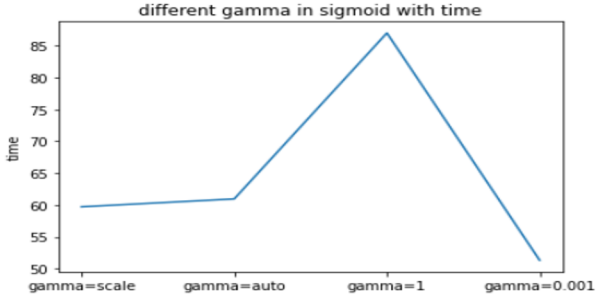
C	Gamma	Accuracy
0.1	Scale	0.3516

Sigmoid



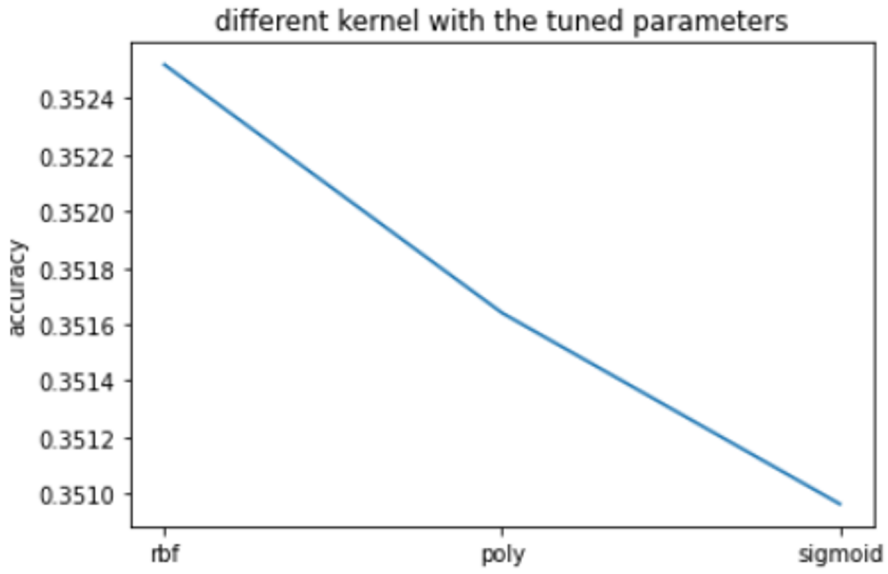


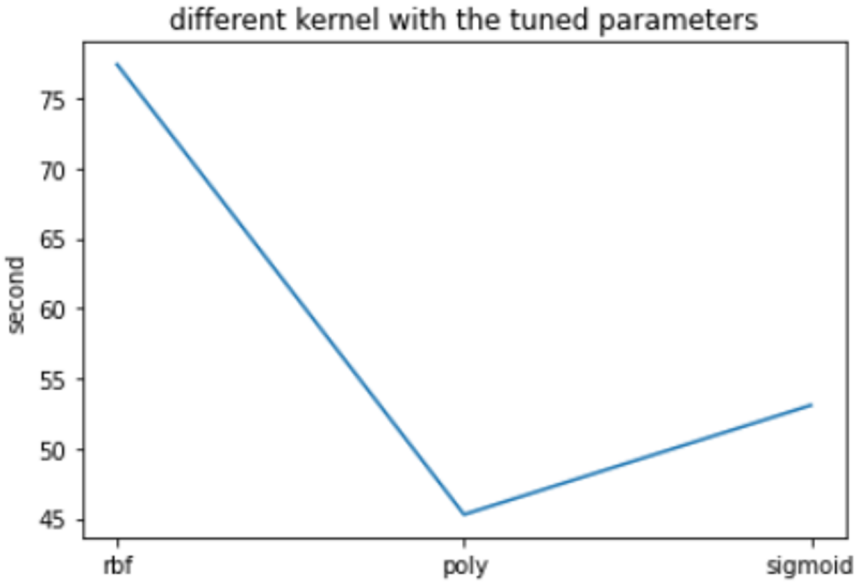




C	Gamma	Accuracy
0.001	0.001	0.3509

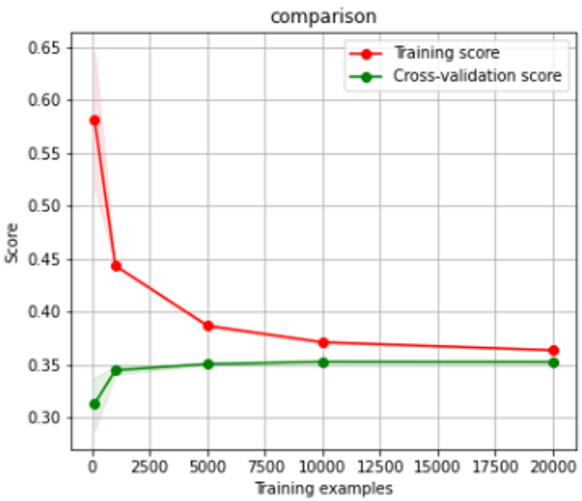
Kernel Comparison

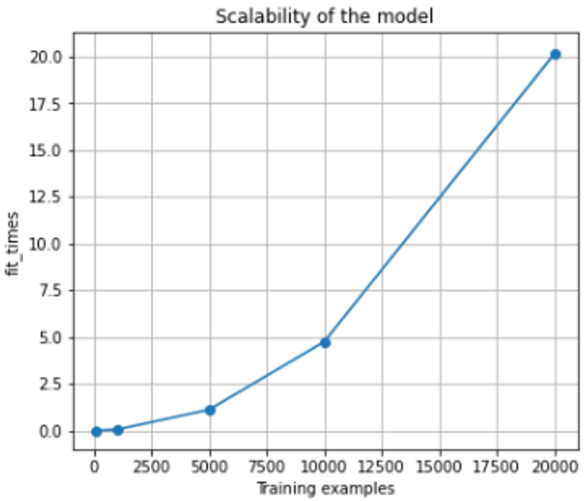


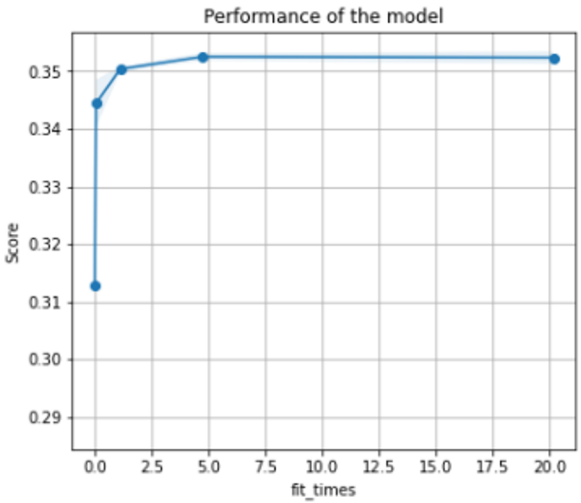


Summary: RBF is the best choice

Overfitting







Summary: The model does not overfit because as the sample size of data increase, the performance of the model doesn't change.

Final Model

Preprocessing: Simple encoding + PCA (11 features)

Kernel function: RBF

Hyperparameter:

c = 1

gamma = scale

Accuracy: 0.3525

Conclusion

Model Comparison

Similarity:

- Simple encoding (15 features)
- Standardized features by removing the mean and scaling to unit variance

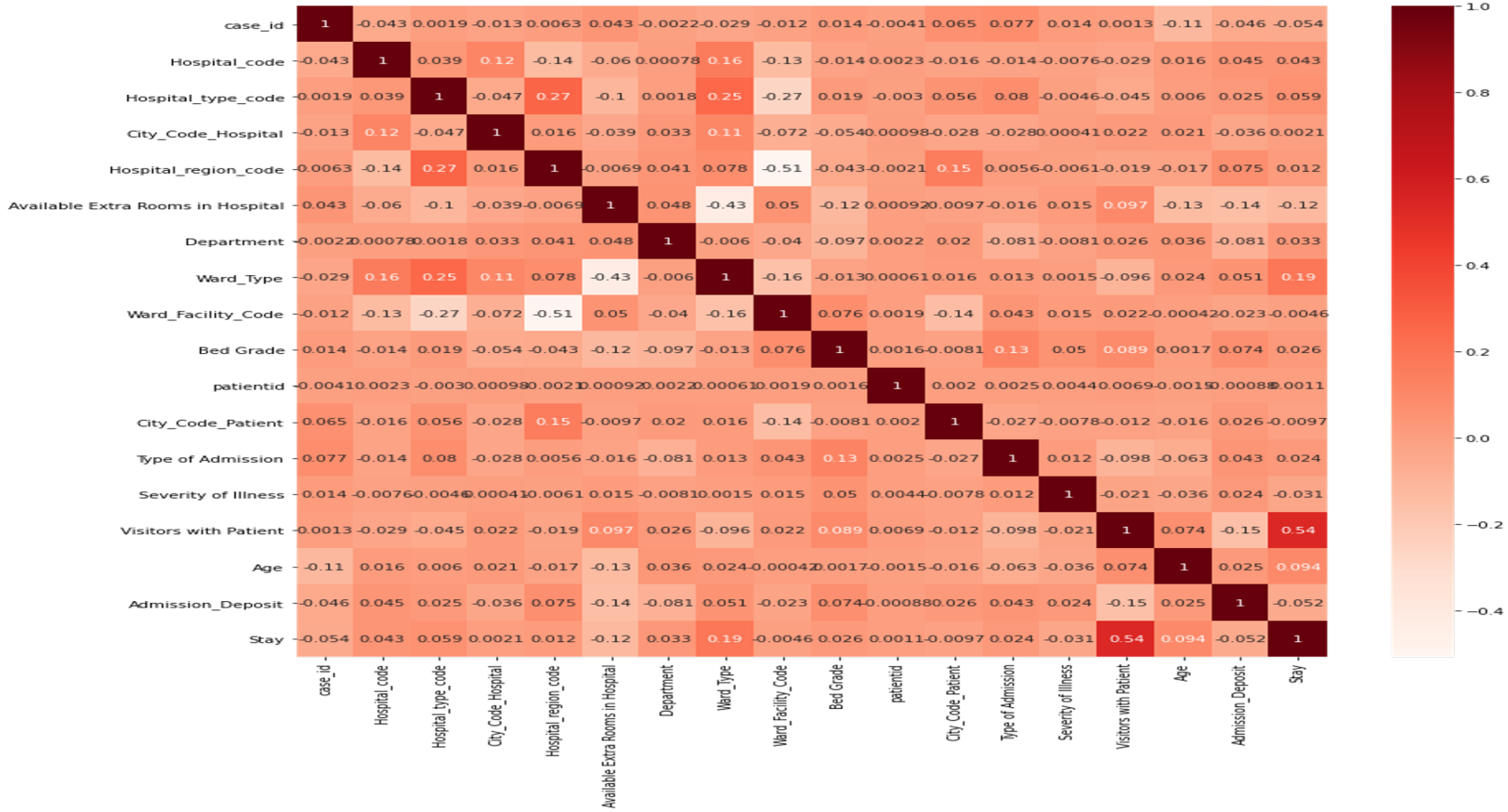
Difference:

- Feature selection: PCA (11 features) or Correlation (6 features)
- Hyperparameters

Accuracy:

- KNN: 0.3442
- Random Forest: 0.3541
- SVM: 0.3525

Correlation



Class-imbalance

Class0	Class1	Class2	Class3	Class4	Class5
160661	78139	87491	55159	11743	35018

Class6	Class7	Class8	Class9	Class10
2744	10254	4838	2765	6683

Conclusion

This data is not suitable for predicting the length of stay, because the correlation between the features we considered and the length of stay is too small and the number of sample for each class are different.

Improved method:

1. Use other features
2. Sample size for each class are same

Thank You !