HW1

903434960 Tzu-Chuan Huang

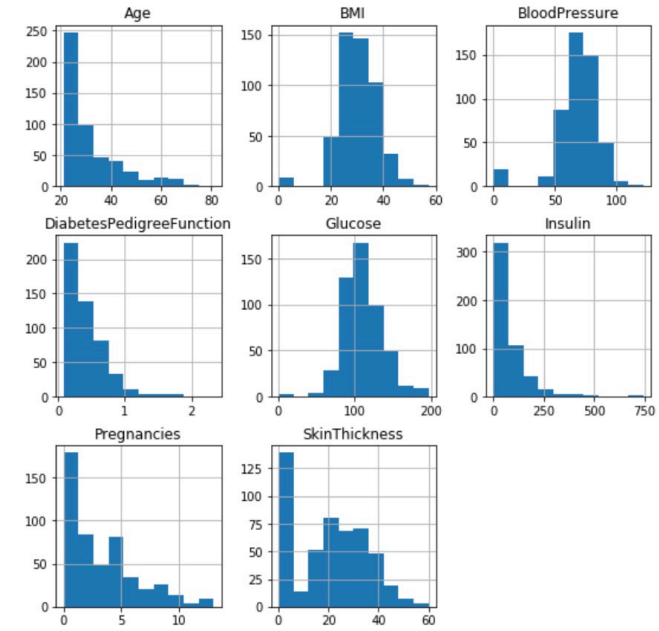
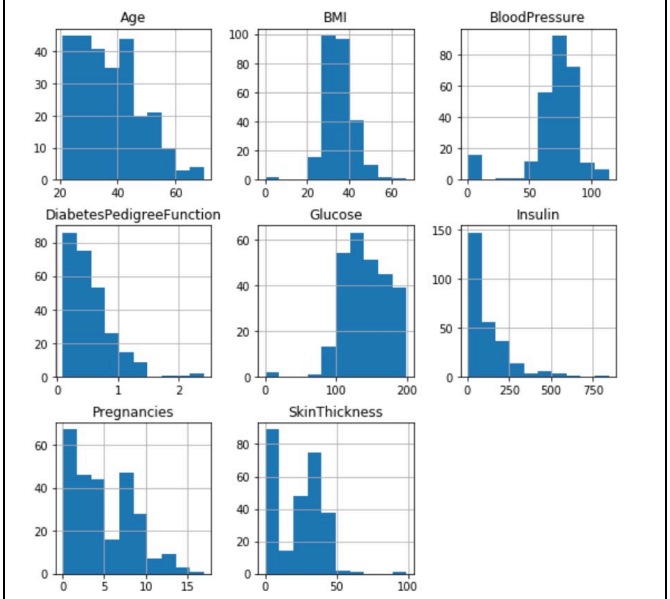
Dataset1:

Introduction:

This is the data from UCI database, and it contains total 9 features, and each patient are diagnosed with or without diabetes. There are total 9 attributes in the data, which is Age, BMI, blood pressure, diabetes Pedigree function, glucose, insulin, pregnancies and skin thickness. Nowadays, people are eager to know whether they have disease or not based on the known features. So this is why I choose the dataset.

Explore the data:

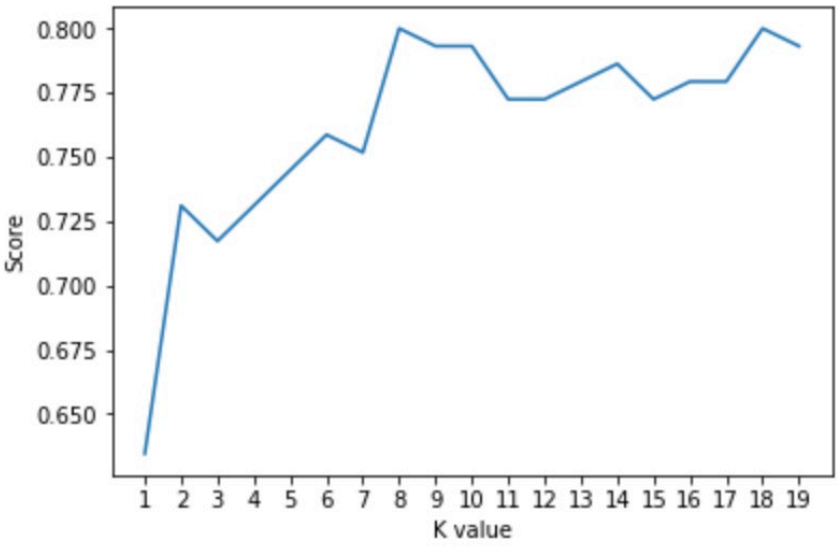
Take a look of the data and visualize the features distribution:

Patients ith diabetes	
	

KNN:

Try to find out the best K value and the plot shows that k equal 8 have the highest accuracy.

So I tried K=4, 8, 9



Before cross validation,
Test size = 0.1:

Decision Tree	Neural network	Decision tree with boosting
Classifier: DT Accuracy: 0.863013698630137 ROC AUC: 0.85515873015873 Precision: 0.8214285714285714 Recall: 0.8214285714285714 F1-score: 0.8214285714285714	Classifier: MLP Accuracy: 0.6027397260273972 ROC AUC: 0.5698412698412699 Precision: 0.48 Recall: 0.42857142857142855 F1-score: 0.4528301886792452	Classifier: RF Accuracy: 0.8493150684931506 ROC AUC: 0.8238095238095239 Precision: 0.8695652173913043 Recall: 0.7142857142857143 F1-score: 0.7843137254901961
KNN(k = 4)	KNN(k = 8)	KNN(k = 9)
Classifier: KNN_4 Accuracy: 0.7123287671232876 ROC AUC: 0.6992063492063492 Precision: 0.6206896551724138 Recall: 0.6428571428571429 F1-score: 0.6315789473684211	Classifier: KNN_8 Accuracy: 0.7123287671232876 ROC AUC: 0.6857142857142857 Precision: 0.64 Recall: 0.5714285714285714 F1-score: 0.6037735849056605	Classifier: KNN_9 Accuracy: 0.7397260273972602 ROC AUC: 0.7146825396825397 Precision: 0.68 Recall: 0.6071428571428571 F1-score: 0.6415094339622641
SVM (linear)	SVM (rbf)	
Classifier: SVC Accuracy: 0.6620689655172414 ROC AUC: 0.5 Precision: 0.0 Recall: 0.0 F1-score: 0.0	Classifier: SVC_RBF Accuracy: 0.6620689655172414 ROC AUC: 0.5 Precision: 0.0 Recall: 0.0 F1-score: 0.0	

Test size = 0.2:

Decision Tree	Neural network	Decision tree with boosting

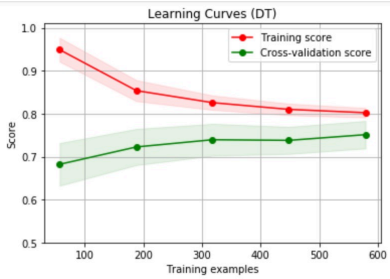
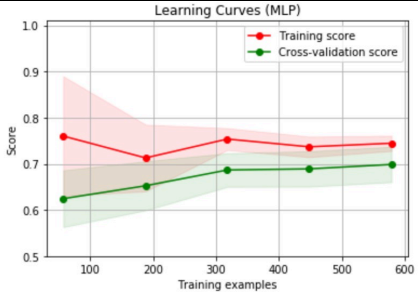
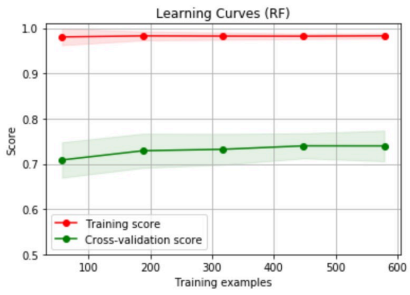
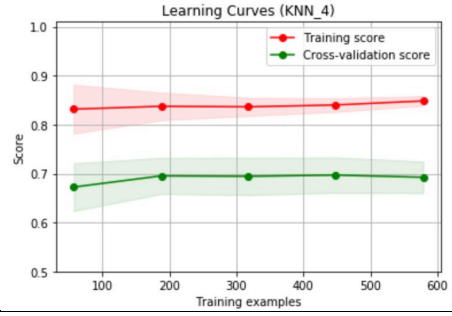
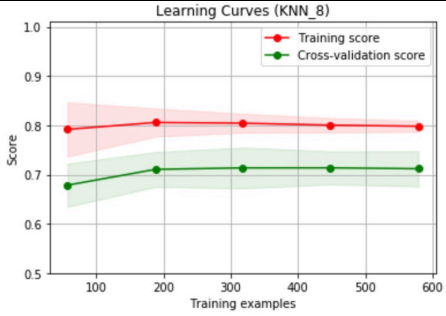
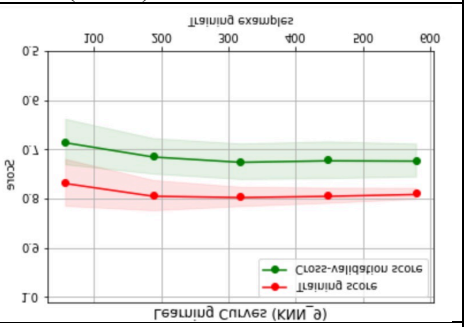
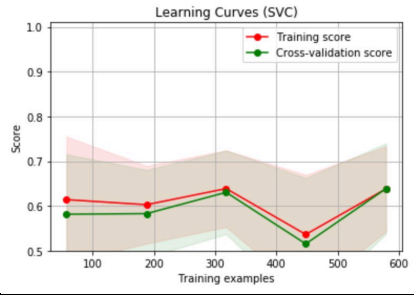
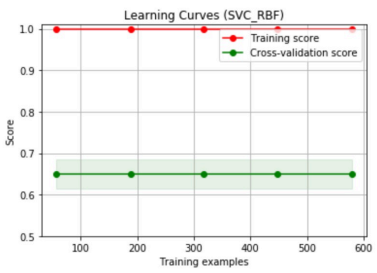
Classifier: DT Accuracy: 0.8068965517241379 ROC AUC: 0.7392644557823129 Precision: 0.8387096774193549 Recall: 0.5306122448979592 F1-score: 0.65	Classifier: MLP Accuracy: 0.7379310344827587 ROC AUC: 0.7171556122448979 Precision: 0.6037735849056604 Recall: 0.6530612244897959 F1-score: 0.6274509803921567	Classifier: RF Accuracy: 0.8482758620689655 ROC AUC: 0.8104804421768708 Precision: 0.8292682926829268 Recall: 0.6938775510204082 F1-score: 0.7555555555555555
KNN(k = 4)	KNN(k = 8)	KNN(k = 9)
Classifier: KNN_4 Accuracy: 0.7172413793103448 ROC AUC: 0.6915391156462586 Precision: 0.5769230769230769 Recall: 0.6122448979591837 F1-score: 0.594059405940594	Classifier: KNN_8 Accuracy: 0.7448275862068966 ROC AUC: 0.7223639455782311 Precision: 0.6153846153846154 Recall: 0.6530612244897959 F1-score: 0.6336633663366337	Classifier: KNN_9 Accuracy: 0.7517241379310344 ROC AUC: 0.7275722789115645 Precision: 0.6274509803921569 Recall: 0.6530612244897959 F1-score: 0.64
SVM (linear)	SVM (rbf)	
Classifier: SVC Accuracy: 0.6620689655172414 ROC AUC: 0.5 Precision: 0.0 Recall: 0.0 F1-score: 0.0	Classifier: SVC_RBF Accuracy: 0.6620689655172414 ROC AUC: 0.5 Precision: 0.0 Recall: 0.0 F1-score: 0.0	

When apply test size = 0.1, the accuracy is better the test size = 0.2 and we could see that Decision tree with boosting have the highest score. Then SVM have the least score.

Following is the score after apply cross-validation. In here, k-fold method are applied. And the number of split is set to 10. Compared the result with the one without cross validation, decision tress and neurual network have a worse performance but others are better. So we could conclude cross validation should applied.

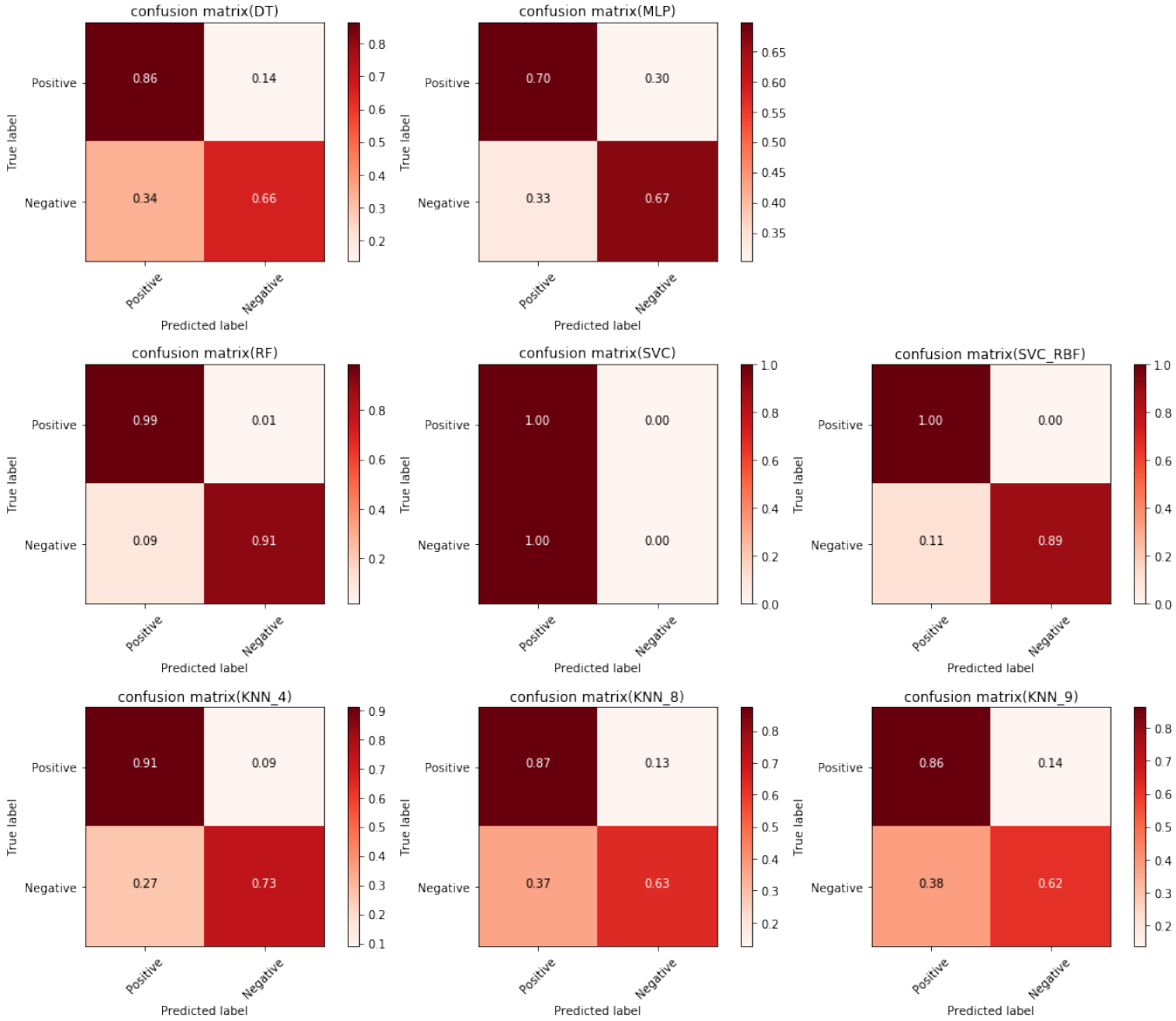
Decision Tree	Neural network	Decision tree with boosting
Classifier: DT Accuracy: 0.7928176795580111 ROC AUC: 0.7608962164447263 Precision: 0.7161572052401747 Recall: 0.6586345381526104 F1-score: 0.6861924686192468	Classifier: MLP Accuracy: 0.6975138121546961 ROC AUC: 0.6404946100190234 Precision: 0.5757575757575758 Recall: 0.4578313253012048 F1-score: 0.5100671140939598	Classifier: RF Accuracy: 0.9696132596685083 ROC AUC: 0.9606002959205242 Precision: 0.9789029535864979 Recall: 0.9317269076305221 F1-score: 0.9547325102880658
KNN(k = 4)	KNN(k = 8)	KNN(k = 9)
Classifier: KNN_4 Accuracy: 0.8494475138121547 ROC AUC: 0.8222067216233354 Precision: 0.8097345132743363 Recall: 0.7349397590361446 F1-score: 0.7705263157894737	Classifier: KNN_8 Accuracy: 0.7914364640883977 ROC AUC: 0.7541111815683788 Precision: 0.7247706422018348 Recall: 0.6345381526104418 F1-score: 0.6766595289079229	Classifier: KNN_9 Accuracy: 0.7790055248618785 ROC AUC: 0.7408158951595858 Precision: 0.7031963470319634 Recall: 0.6184738955823293 F1-score: 0.6581196581196581
SVM (linear)	SVM (rbf)	
Classifier: SVC Accuracy: 0.9613259668508287 ROC AUC: 0.9437751004016064 Precision: 1.0 Recall: 0.8875502008032129 F1-score: 0.9404255319148938	Classifier: SVC_RBF Accuracy: 0.9613259668508287 ROC AUC: 0.9437751004016064 Precision: 1.0 Recall: 0.8875502008032129 F1-score: 0.9404255319148938	

Learning rate under 0.2 test size:

Decision Tree	Neural network	Decision tree with boosting
		
KNN(k = 4)	**KNN(k = 8)**	**KNN(k = 9)**
		
SVM (linear)	**SVM (rbf)**	
		

The learning curve in decision tree algorithm is gradually climbing up as the training score is lower. But as we could see in the other algorithm, the learning rate stays the same so it means that more iteration time would not bring a better solution.

The confusion matrix of all the algorithm after coss-validation:



Dataset 2:

1. Introduction:

The dataset1 is from ACM RecSys challenge 2015, because e-commerce has been more popular now than before, people are tend to purchase items on the Internet than go to physical store. At the time it brings lots of convenience to people, the recommendation system(RS) based on the collected data is much more important to deliver the demand.
As a girl, when I surf some items in the online store, some popped up items definitely catch my eyes and then later being added in my shopping cart. But not every popped up items will go into my cart, it depends on the price, shopping time… and so on. So I am interested in this recommendation systems in e-commerce business to find out is there a pattern or way to predict the recommendation items.

2.Problem statement:

In the challenge, the dataset from YOOCHOOSE is a collection of sequences of click events and click sessions. They are also buying events provided. Here are two classification problem:
(1) Predict whether the user in a session is going to but something or not
(2) If he or she is buying, what items he is going to buy
The information is valuable in the e-business to suggest user some interested items and encourage them to purchase the items.

Following is the description and features of the dataset:

yoochoose-clicks.dat - Click events. Each record/line in the file has the following fields:

1. Session ID – the id of the session. In one session there are one or many clicks.
2. Timestamp – the time when the click occurred.
3. Item ID – the unique identifier of the item.
4. Category – the category of the item.

yoochoose-buys.dat - Buy events. Each record/line in the file has the following fields:

1. Session ID - the id of the session. In one session there are one or many buying events.
2. Timestamp - the time when the buy occurred.
3. Item ID – the unique identifier of item.
4. Price – the price of the item.
5. Quantity – how many of this item were bought.

The Session ID in yoochoose-buys.dat will always exist in the yoochoose-clicks.dat file. The session could be short (few minutes) or very long (few hours), it could have one click or hundreds of clicks.

3. Date Preprocessing and data exploration

Before jump into the machine learning algorithm, there are some tasks needed to be done first. The first thing is to clean and explore the dataset. Because the dataset is too large, I choose to extract the data only in September.
First, take a look on the clicks dataset:

	Session_ID	Time_stamp	Item_ID	Category
26566438	9293604	2014-09-01 18:07:58.937	214839911	S
26566456	9293603	2014-09-01 14:22:06.391	214701787	S
26566457	9293603	2014-09-01 14:23:19.505	214853657	1
26566458	9293603	2014-09-01 14:24:00.404	214701787	S
26566462	9293613	2014-09-01 15:11:33.588	214834871	S

We could find out that several item IDs appears in the same session ID, it means some buyers are interested in lots of items.

Also take a look at buys dataset:

	Session_ID	Time_stamp	Item_ID	Price	Quantity
900729	9641594	2014-09-01 09:09:25.575	214853342	2093	2
900730	9641594	2014-09-01 09:09:25.596	214853340	837	2
900731	9641594	2014-09-01 09:09:25.614	214853420	1046	2

Same session ID will but more than one items at a time.

Then, merge the two dataset into one dataset where contains the following features:

Session_ID Time_stamp_x Item_ID Category Time_stamp_y Price Quantity

Where Time_stamp_x represent the click time and Time_stamp_y represent buys time.
The dataset now looks like:

	Session_ID	Time_stamp_x	Item_ID	Category	Time_stamp_y	Price	Quantity
258481	9641594	2014-09-01 08:47:30.541	214853342	S	2014-09-01 09:09:25.575	2093.0	2.0
258482	9641594	2014-09-01 08:48:00.179	214853340	S	2014-09-01 09:09:25.596	837.0	2.0
258483	9641594	2014-09-01 08:48:30.753	214853420	S	2014-09-01 09:09:25.614	1046.0	2.0
258484	9641594	2014-09-01 08:50:21.575	214850947	S	NaT	NaN	NaN

After, convert the click and buy time stamp into a more readable one, and it would be much easier to visualize the data, which is a set of
(Click_Weekday, Click_HourofDay, Buy_Weekday, Buy_HourofDay, Time_Gap)
Also, add the attributes 'Buyer' and 'Sales' (Sale = Price x Quantity)
Now the datasets looks like

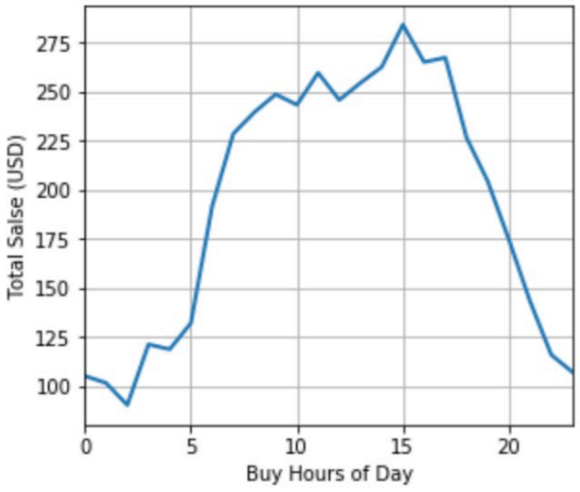
	Session_ID	Clicked_Time	Item_ID	Category	Buy_Time	Price	Quantity	Buyer	Clicked_Weekday	Clicked_HourOfDay	Clicked_Day	Time_Gap	Sales
0	9293604	2014-09-01 18:07:58.937	214839911	S	2014-09-01 18:07:58.937	0.0	0.0	False	0	18	09/01	0.0	0.0
1	9293603	2014-09-01 14:22:06.391	214701787	S	2014-09-01 14:22:06.391	0.0	0.0	False	0	14	09/01	0.0	0.0
2	9293603	2014-09-01 14:23:19.505	214853657	1	2014-09-01 14:23:19.505	0.0	0.0	False	0	14	09/01	0.0	0.0
3	9293603	2014-09-01 14:24:00.404	214701787	S	2014-09-01 14:24:00.404	0.0	0.0	False	0	14	09/01	0.0	0.0
4	9293613	2014-09-01 15:11:33.588	214834871	S	2014-09-01 15:11:33.588	0.0	0.0	False	0	15	09/01	0.0	0.0

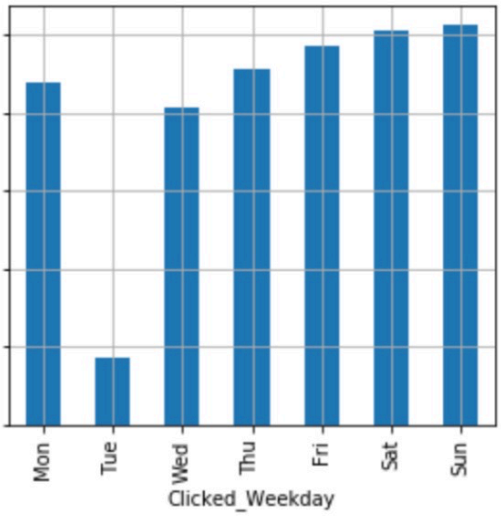
Now I could visualize the data with different features:
(1) Click day vs Sales amount :
It is obvious that the total sales is an up-and-down patterns and reached to the peak at the 20th of September.

(2) Click hours of a day v.s. total sales and Click_Weekday v.s. total sales

Total sales reached to it's peak at 3.p.m. in everyday routine. It's interesting maybe it's the break time during work.

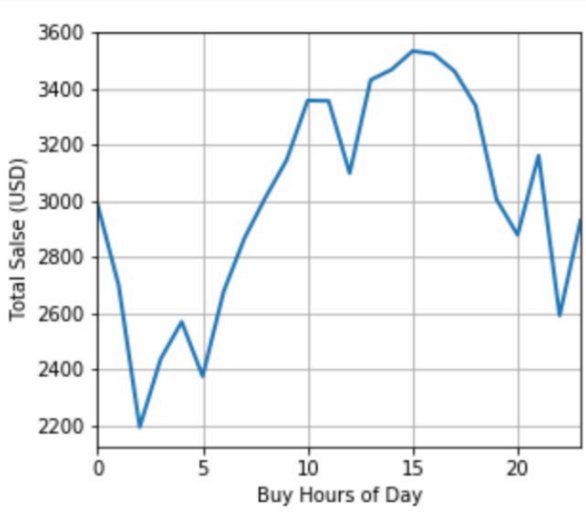
The clicked time is much lower on Tuesday but will climb up gradually when it comes to weedend.

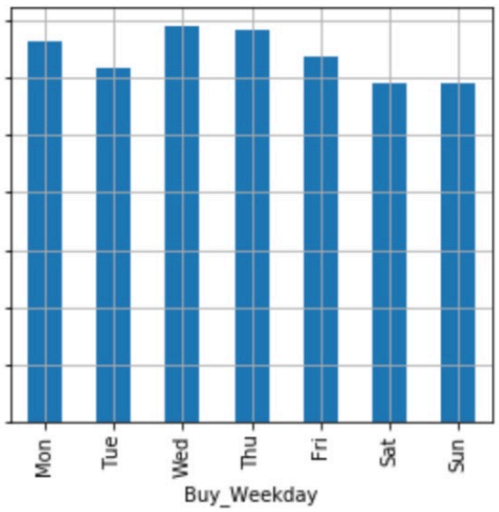




(3) Buy_hours of a day v.s. total sales and Buy_Weekday v.s. total sales

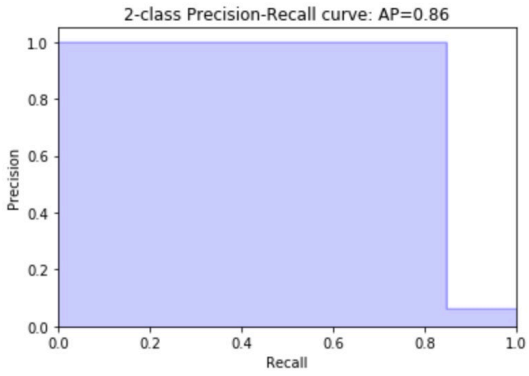
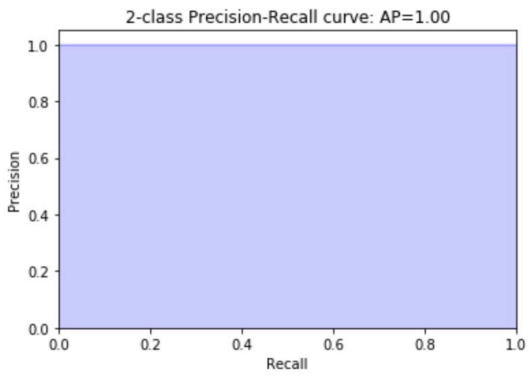
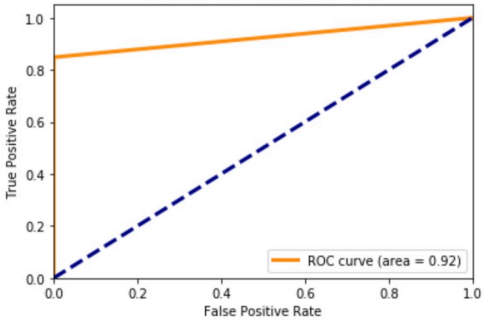
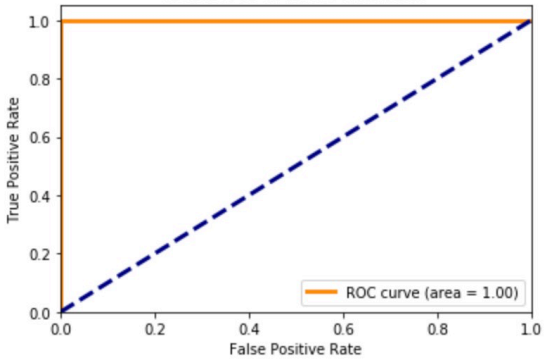
The buy hours reached to the top at 3 p.m. and people have a higher tendency to purchase items on Wednesday and Thursday.

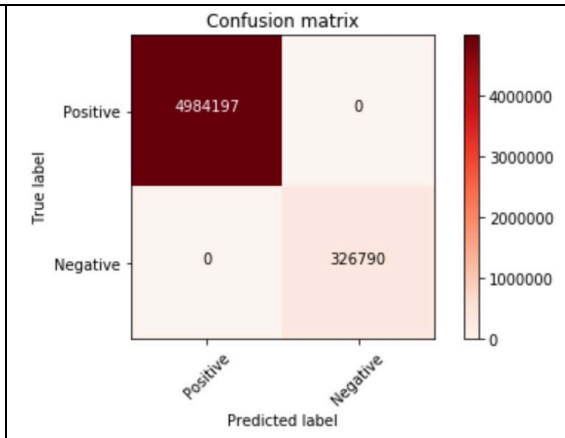
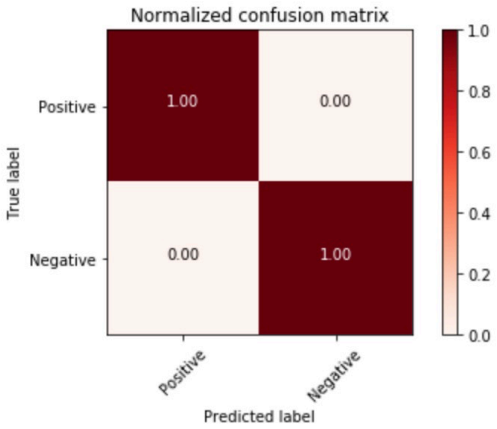




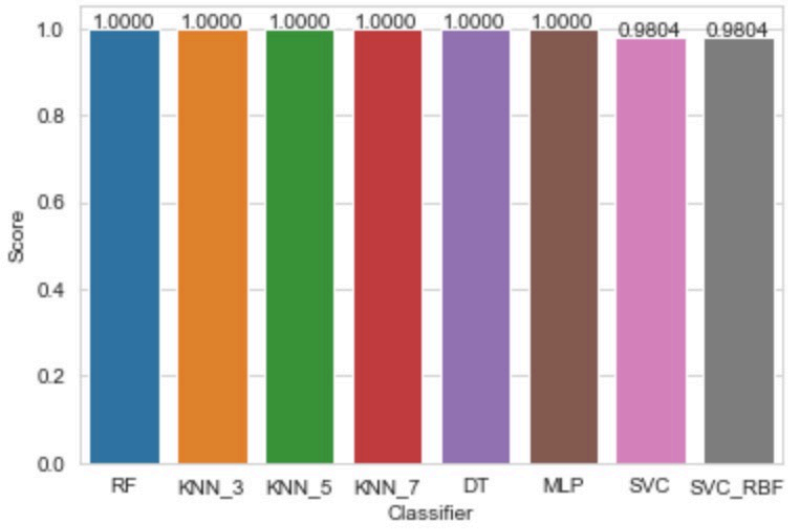
3. Machine learning algorithm

Logistic regression:

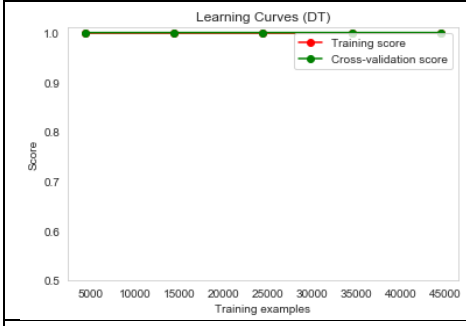
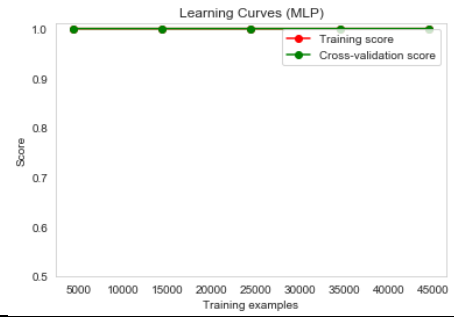
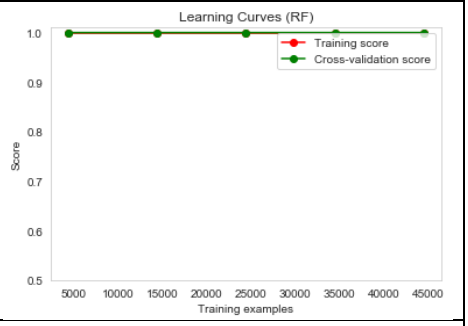
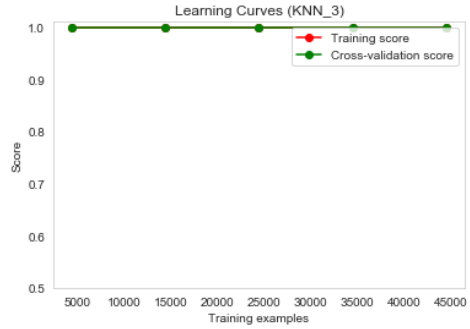
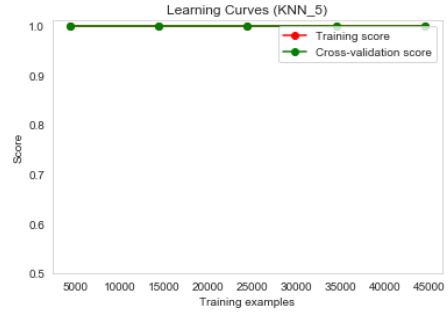
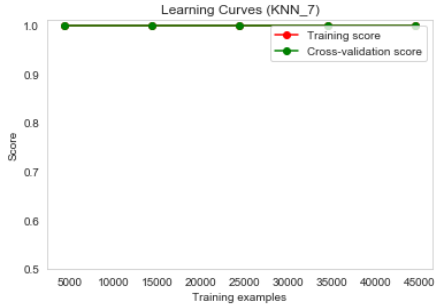
	Before normalization	After normalization
Summary	Classifier: Logistic Regression Accuracy: 0.9906574804268962 ROC AUC: 0.9240827442700206 Precision: 1.0 Recall: 0.848165488540041 F1-score: 0.9178458247373179	Classifier: Logistic Regression Accuracy: 1.0 ROC AUC: 1.0 Precision: 1.0 Recall: 1.0 F1-score: 1.0
2 class precision-recall curve		
ROC curve		

Confusion matrix		

	Name	Score
2	RF	1.000000
5	KNN_3	1.000000
6	KNN_5	1.000000
7	KNN_7	1.000000
0	DT	0.999982
1	MLP	0.999964
3	SVC	0.980382
4	SVC_RBF	0.980382



And because in problem 1, we know that cross-validation will increase the accuracy. So in here, we directly apply cross-validation here.
Here are the learning curve according to different algorighm. The line of training score and the cross-validation score overlap each other. It means the algorithm applied on the dataset could precisely predict the outcome.

Decision Tree	Neural network	Decision tree with boosting
		
KNN(k = 3)	KNN(k = 5)	KNN(k = 7)
		
SVM (linear)	SVM (rbf)