12 Appendix

12.1 PYTHON Code

```
import xlrd
import xlwt
ExcelFile=xlrd.open_workbook(r'C:\Users\tianzhy\Desktop\sumai.xlsx')
sheet=ExcelFile.sheet_by_name('速卖')
workbook = xlwt.Workbook(encoding = 'ascii')
worksheet = workbook.add_sheet('My Worksheet')
row=0
for i in range (1,2291):
    count=0
    for j in range (0,19):
        temp = str(sheet.cell(i,j).value)
        if float(temp)==0.0:
            count=count+1
        #print(j)
    if count<=3:
        row=row+1
        for j in range (0,56):
            temp = str(sheet.cell(i,j).value)
            worksheet.write(row, j, label = str(temp))
        worksheet.write(row, 56, label = str(i+1))
    #print(i)
workbook.save('Excel_Workbook.xls')

import xlrd
import xlwt
ExcelFile=xlrd.open_workbook(r'C:\Users\tianzhy\Desktop\detail_numerical.xls')
sheet=ExcelFile.sheet_by_name('My Worksheet')
workbook = xlwt.Workbook(encoding = 'ascii')
worksheet = workbook.add_sheet('My Worksheet')
for i in range (1,2291):
    temp = str(sheet.cell(i,7).value)
    if ('Gravity Response' in temp) :
        worksheet.write(i, 0, label = str(1))
    else:
        worksheet.write(i, 0, label = str(0))
    if ('GPRS' in temp):
        worksheet.write(i, 1, label = str(1))
    else:
        worksheet.write(i, 1, label = str(0))
workbook.save('Excel_Workbook.xls')
```

```
import xlrd
import xlwt
ExcelFile=xlrd.open_workbook(r'C:\Users\tianzhy\Desktop\工作簿 1.xlsx')
sheet=ExcelFile.sheet_by_name('Sheet1')
workbook = xlwt.Workbook(encoding = 'ascii')
worksheet = workbook.add_sheet('My Worksheet')
for i in range (1,2291):
    temp = str(sheet.cell(i,0).value)
    if 'x' in temp:
        temp = str.split(temp,'x')
        worksheet.write(i, 0, label = str(temp[0]))
        worksheet.write(i, 1, label = str(temp[1]))
        worksheet.write(i, 2, label = str(temp[2]))
    elif '*' in temp:
        temp = str.split(temp,'*')
        worksheet.write(i, 0, label = str(temp[0]))
        worksheet.write(i, 1, label = str(temp[1]))
        worksheet.write(i, 2, label = str(temp[2]))
    elif 'X' in temp:
        temp = str.split(temp,'X')
        worksheet.write(i, 0, label = str(temp[0]))
        worksheet.write(i, 1, label = str(temp[1]))
        worksheet.write(i, 2, label = str(temp[2]))
    else:
        worksheet.write(i, 0, label = str(temp[0]))
        worksheet.write(i, 1, label = str(temp[1]))
        worksheet.write(i, 2, label = str(temp[2]))
        temp=[0,0,0]
    temp[0]=float(temp[0])
    temp[1]=float(temp[1])
    temp[2]=float(temp[2])
    judge=temp[0]*temp[1]*temp[2]
    if judge<36.8633431902425:
        temp[0]=temp[0]*25.4
        temp[1]=temp[1]*25.4
        temp[2]=temp[2]*25.4
    elif judge<4712.4514674042:
        temp[0]=temp[0]*10
        temp[1]=temp[1]*10
        temp[2]=temp[2]*10
    worksheet.write(i, 3, label = str(temp[0]))
    worksheet.write(i, 4, label = str(temp[1]))
    worksheet.write(i, 5, label = str(temp[2]))
```

```
        #worksheet.write(i, 0, label = str(count))
workbook.save('Excel_Workbook.xls')

import xlrd
import xlwt
ExcelFile=xlrd.open_workbook(r'C:\Users\tianzhy\Desktop\sumai.xlsx')
sheet=ExcelFile.sheet_by_name('速卖')
workbook = xlwt.Workbook(encoding = 'ascii')
worksheet = workbook.add_sheet('My Worksheet')
properties1=['Unlock Phones','Google Play','Battery Type','Battery Capacity','Display Resolution','Operation System','Feature','SIM Card Quantity','Recording Definition','Touch Screen Type','RAM','ROM','color']

properties2=['Size','Display Size']
properties3=['Camera：','Camera Type','Front Camera：']
properties4=['CPU：Octa Core','CPU：Quad Core','CPU：Dual Core']
for i in range (1,2291):
    temp = sheet.cell(i,6).value
    temp = str.split(temp,'<br>')
    length=len(temp)
    for j in range (0,13):
        vari=0
        for k in range (0,length):
            if properties1[j] in temp[k]:
                vari=temp[k]
        worksheet.write(i, j, label = str(vari))
    vari=0
    for k in range (0,length):
        if properties4[0] in temp[k]:
                vari=properties4[0]
        if properties4[1] in temp[k]:
                vari=properties4[1]
        if properties4[2] in temp[k]:
                vari=properties4[2]
    worksheet.write(i, 13, label = str(vari))
    vari=0
    for k in range (0,length):
        if 'Size' in temp[k]:
            vari=vari+1
            if vari < 3:
                worksheet.write(i, (13+vari), label = str(temp[k]))
    vari=0
    for k in range (0,length):
        if 'Camera：' in temp[k]:
```

```
                vari=vari+1
                worksheet.write(i, (15+vari), label = str(temp[k]))

workbook.save('Excel_Workbook.xls')

import xlrd
import xlwt
ExcelFile=xlrd.open_workbook(r'C:\Users\tianzhy\Desktop\detail_numerical.xls')
sheet=ExcelFile.sheet_by_name('My Worksheet')
workbook = xlwt.Workbook(encoding = 'ascii')
worksheet = workbook.add_sheet('My Worksheet')
for i in range (1,2291):
    temp = str(sheet.cell(i,13).value)
    if ('Black' in temp) or ('black' in temp):
        worksheet.write(i, 0, label = str(1))
    else:
        worksheet.write(i, 0, label = str(0))
    if ('White' in temp) or ('white' in temp):
        worksheet.write(i, 1, label = str(1))
    else:
        worksheet.write(i, 1, label = str(0))
    if ('Blue' in temp) or ('blue' in temp):
        worksheet.write(i, 2, label = str(1))
    else:
        worksheet.write(i, 2, label = str(0))
    if ('Rose' in temp) or ('rose' in temp):
        worksheet.write(i, 3, label = str(1))
    else:
        worksheet.write(i, 3, label = str(0))
    if ('Gold' in temp) or ('gold' in temp) or ('champange' in temp) or ('Champange' in temp):
        worksheet.write(i, 4, label = str(1))
    else:
        worksheet.write(i, 4, label = str(0))
    if ('Silver' in temp) or ('silver' in temp):
        worksheet.write(i, 5, label = str(1))
    else:
        worksheet.write(i, 5, label = str(0))
    if ('Grey' in temp) or ('grey' in temp) or ('titanium' in temp) or ('Titanium' in temp):
        worksheet.write(i, 6, label = str(1))
    else:
        worksheet.write(i, 6, label = str(0))
    if ('Pink' in temp) or ('pink' in temp):
        worksheet.write(i, 7, label = str(1))
    else:
```

```
            worksheet.write(i, 7, label = str(0))
        if ('Brown' in temp) or ('brown' in temp):
            worksheet.write(i, 8, label = str(1))
        else:
            worksheet.write(i, 8, label = str(0))
        if ('Orange' in temp) or ('orange' in temp):
            worksheet.write(i, 9, label = str(1))
        else:
            worksheet.write(i, 9, label = str(0))
        if ('Yellow' in temp) or ('yellow' in temp):
            worksheet.write(i, 10, label = str(1))
        else:
            worksheet.write(i, 10, label = str(0))
        if ('Red' in temp) or ('red' in temp):
            worksheet.write(i, 11, label = str(1))
        else:
            worksheet.write(i, 11, label = str(0))

workbook.save('Excel_Workbook.xls')

import xlrd
import xlwt
ExcelFile=xlrd.open_workbook(r'C:\Users\tianzhy\Desktop\sumai.xlsx')
sheet=ExcelFile.sheet_by_name('速卖')
workbook = xlwt.Workbook(encoding = 'ascii')
worksheet = workbook.add_sheet('My Worksheet')
for i in range (1,2291):
    vari=0
    temp = sheet.cell(i,6).value
    if 'Dual Camera' in temp:
        vari=1
    if 'Dual camera' in temp:
        vari=1
    if 'Dual Front Camera' in temp:
        vari=1
    if 'Dual Back Camera' in temp:
        vari=1
    if 'Dual front Camera' in temp:
        vari=1
    if 'Dual Rear Camera' in temp:
        vari=1
    if 'Dual rear Camera' in temp:
        vari=1
    if 'Dual back Camera' in temp:
```

```
            vari=1
        worksheet.write(i, 0, label = vari)
        vari=0
        if 'Front Camera' in temp:
            vari=1
        if 'front Camera' in temp:
            vari=1
        worksheet.write(i, 1, label = vari)
workbook.save('Excel_Workbook.xls')

import xlrd
import xlwt
ExcelFile=xlrd.open_workbook(r'C:\Users\tianzhy\Desktop\sumai.xlsx')
sheet=ExcelFile.sheet_by_name('速卖')
workbook = xlwt.Workbook(encoding = 'ascii')
worksheet = workbook.add_sheet('My Worksheet')
for i in range (1,2291):
        temp = sheet.cell(i,5).value
        temp = str.split(temp)
        length=len(temp)
        count=0
        for j in range (0,length):
            if temp[j]=='Xiaomi'or temp[j]=='xiaomi'or temp[j]=='XIAOMI':
                count=1
            elif temp[j]=='Huawei'or temp[j]=='HUAWEI'or temp[j]=='huawei':
                count=2
            elif temp[j]=='MEIZU'or temp[j]=='meizu'or temp[j]=='Meizu':
                count=3
            elif temp[j]=='LENOVO'or temp[j]=='Lenovo'or temp[j]=='lenovo':
                count=4
            elif temp[j]=='IPHONE'or temp[j]=='iphone'or temp[j]=='iPhone':
                count=5
            elif temp[j]=='OPPO'or temp[j]=='Oppo'or temp[j]=='oppo':
                count=6
            elif temp[j]=='Vivo'or temp[j]=='vivo'or temp[j]=='VIVO':
                count=7
            elif temp[j]=='Nubia'or temp[j]=='NUBIA'or temp[j]=='nubia':
                count=8
            elif temp[j]=='samsung'or temp[j]=='Samsung'or temp[j]=='SAMSUNG':
                count=9
            elif temp[j]=='ZTE'or temp[j]=='zte':
                count=10
            elif temp[j]=='HOMTOM'or temp[j]=='homtom'or temp[j]=='Homtom':
                count=11
```

```
            elif temp[j]=='DOOGEE'or temp[j]=='Doogee'or temp[j]=='doogee':
                count=12
            elif temp[j]=='Letv'or temp[j]=='LeTv'or temp[j]=='letv' or temp[j]=='LETV':
                count=13
            elif temp[j]=='Blackview'or temp[j]=='BLACKVIEW'or temp[j]=='blackview':
                count=14
            elif temp[j]=='NOKIA'or temp[j]=='Nokia'or temp[j]=='nokia':
                count=15
        worksheet.write(i, 0, label = str(count))
workbook.save('Excel_Workbook.xls')

import pandas as pd
import xgboost as xgb
from sklearn import preprocessing
import numpy as np

train = pd.read_csv(r'D:\XGBoost_learn\click rate\train1.csv', header=0)
tests = pd.read_csv(r'D:\XGBoost_learn\click rate\test_pre.csv', header=0)
# trains=train.iloc[:, 1:].values
# labels=train.iloc[:,:1].values
# test = tests.iloc[:, :].values
'''
train['time_stamp'] = pd.to_datetime(pd.Series(train['time_stamp']))
tests['time_stamp'] = pd.to_datetime(pd.Series(tests['time_stamp']))
train['Year'] = train['time_stamp'].apply(lambda x: x.year)#Year
train['Month'] = train['time_stamp'].apply(lambda x: x.month)#Month
train['weekday'] = train['time_stamp'].dt.dayofweek#weekday
train['time'] = train['time_stamp'].dt.time#time
tests['Year'] = tests['time_stamp'].apply(lambda x: x.year)#Year
tests['Month'] = tests['time_stamp'].apply(lambda x: x.month)#Month
tests['weekday'] = tests['time_stamp'].dt.dayofweek#weekday
tests['time'] = tests['time_stamp'].dt.time#time
train = train.drop('time_stamp', axis=1)
train = train.dropna(axis=0)
tests = tests.drop('time_stamp', axis=1)
tests = tests.fillna(method='pad')
'''

for f in train.columns:
    if train[f].dtype=='object':
        if f != 'shop_id':
            print(f)
            lbl = preprocessing.LabelEncoder()
            lbl.fit(list(train[f].values))
```

```
                train[f] = lbl.transform(list(train[f].values))

for f in tests.columns:
    if tests[f].dtype == 'object':
        print(f)
        lbl = preprocessing.LabelEncoder()
        lbl.fit(list(tests[f].values))
        tests[f] = lbl.transform(list(tests[f].values))
print("test")
print(tests.info())
# for f in train.columns:
#     if f !='':
#         train[f] = train[f].astype(float)

print(train.info())
# train = train.astype(float)
# tests = tests.astype(float)
trains = train.iloc[:, 1:].values
labels = train.iloc[:, :1].values
test = tests.iloc[:, 1:].values

feature_columns_to_use = ['wifi_strong1','wifi_strong2','wifi_strong3']

big_X = train[feature_columns_to_use].append(tests[feature_columns_to_use])
train_X = big_X[0:train.shape[0]].as_matrix()
test_X = big_X[train.shape[0]::].as_matrix()
train_y = train['shop_id']
gbm = xgb.XGBClassifier(silent=1, max_depth=3, n_estimators=300, learning_rate=0.05)
gbm.fit(train_X, train_y)
predictions = gbm.predict(test_X)
submission = pd.DataFrame({'row_id': tests['row_id'],
                           'shop_id': predictions})
print(submission)
submission.to_csv("submission.csv", index=False)
'''
print(trains)
parameters={
    'silent':1,
    'max_depth': 3,
    'n_estimators':300,
    'learning_rate':0.005,
}
feature_types={
```

```
'float','str','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float','float',
}
# feature_types = {
#         'str', 'float',
# }
ft=list(feature_types)
ParamLst = dict(parameters.items())

offset = 15

num_rounds = 1
xgtest = xgb.DMatrix(tests)

print("///////////////////////////")
# 划分训练集与验证集
xgtrain = xgb.DMatrix(trains[:offset,:], label=labels[:offset])
print("///////////////////////////")
xgval = xgb.DMatrix(trains[offset:, :], label=labels[offset:])

# return 训练和验证的错误率
watchlist = [(xgtrain, 'train'), (xgval, 'val')]


# training model
# early_stopping_rounds 当设置的迭代次数较大时，early_stopping_rounds 可在一定的迭代次数内准确率没有提升就停止训练
model = xgb.train(ParamLst, xgtrain, num_rounds, watchlist, early_stopping_rounds=100)
# model.save_model('./model/xgb.model') # 用于存储训练出的模型
preds = model.predict(xgtest, ntree_limit=model.best_iteration)

# 将预测结果写入文件
np.savetxt('submission_xgb_MultiSoftmax.csv', np.c_[range(1, len(test)+1), preds],
                delimiter=',', header='ImageId,Label', comments='', fmt='%d')
'''
```

12.2MATLAB Code

```
%clear B
jitiaoshuju=size(A);
jitiaoshuju=jitiaoshuju(1,1);
R=A(:,1);
xingbie=max(R);
```

```
R=A(:,2);
jigecanhe=max(R);
%AÊÇÔ-Ê¼ÊýÝ
for b=1:xingbie%ÐÔ±ð
B{b}=[];
end
for c=1:jitiaoshuju%¼ÌõÊý¾Ý
for d=1:xingbie%ÐÔ±ð
if A(c,1)==d
B{d}=[B{d};A(c,:)];
end
end
end
for e=1:xingbie%ÐÔ±ð
for f=1:jigecanhe%¼¸ö²Í°Ð
T=B{e}(:,2);
Q=find(T(T==f));
U(e,f)=max(Q);
end
end
%UÊÇÔ-Ê¼¸öÊý£¬±ÈÈçµÚ¶þÐÐµÚÒ»ÁÐ¾ÍÊÇÅ®µÄ²Í°ÐÑ¡ÏîÎª1µÄÓÐ¶àÉÙ¸öÈË
for l=1:xingbie%ÓÐ¼¸öÐÔ±ð
for j=1:jigecanhe%ÓÐ¼¸ö²Í°Ð
for k=1:jigecanhe%ÓÐ¼¸ö²Í°Ð
C{l}(j,k)=U(l,j)/U(l,k);
end
end
end
for i=1:xingbie%ÓÐ¼¸öÐÔ±ð
t=C{i};
[x,lumda]=eig(t);
r=abs(sum(lumda));
n=find(r==max(r));
max_lumda_A(1,i)=lumda(n,n);
max_x_A{i}=x(:,n); %ÌØÕ÷Öµ
max_x_A{i}=max_x_A{i}./sum(max_x_A{i});
end
for p=1:xingbie%ÓÐ¼¸öÐÔ±ð
for q=1:jigecanhe%ÓÐ¼¸ö²Í°Ð
max_x_AB(p,q)=max_x_A{p}(q,1);%ÌØÕ÷ÏòÁ¿£¬µÚÒ»ÐÐÊÇÄÐµÄÄ¸÷ÏîÈ¨ÖØ£¬µÚ¶þ
ÐÐÊÇÅ®µÄÄ¸÷ÏîÈ¨ÖØ
end
end
```

```
for i =1:1324
if A(i)==2
       A(i)=1;
elseif A(i)==4
       A(i)=1;
elseif A(i)==8
       A(i)=2;
elseif A(i)==16
       A(i)=2;
elseif A(i)==32
       A(i)=2;
elseif A(i)==64
       A(i)=3;
elseif A(i)==128
       A(i)=3;
elseif A(i)==256
       A(i)=3;
%%else
      %% A(i)=5;
end
end

p=p';
t=t';
net=newff(minmax(p),[10 1]);
net.trainParam.epochs=1000;
net.trainParam.goal=0.001;
net.trainParam.show=50;
net.trainParam.lr=0.05;
net.trainParam.mc=0.9;
net=train(net,p,t);
A=sim(net,test);
A=A';

[m,n]=size(A);
[p,q]=size(Z);
for j =1:n
    B{j}=A(:,j);
    B{j}=B{j}/std(B{j});

for i=1:(m-1)
       B_{j}(i)=B{j}(i)-B{j}(i+1);
end
end
```

```
for j =1:q
    Y{j}=Z(:,j);
    Y{j}=Y{j}/std(Y{j});

for i=1:(p-1)
       Y_{j}(i)=Y{j}(i)-Y{j}(i+1);
end
end
for i=1:q
    aver(i,1)=mean(Y_{i});
for j = 1:n
       aver(i,(j+1))=abs(sum(Y_{i})-sum(B_{j}));
       final(i,j)=(1+aver(i,1))/(1+aver(i,1)+aver(i,(j+1)));
end
end
%averv0=mean(x0_);
%averv1=averv1/1323;
%final=(1+averv0)/(1+averv0+averv);

B=0:1:15;
B=B';
B_=zeros(16,1);
B=[B B_];
for i=1:2290
    B((A(i,1)+1),2)=B((A(i,1)+1),2)+1;
end
count=0;
for i = 1:16
if B(i,2)~=0
       count=count+1;
       C(count,1)=B(i,1);
       C(count,2)=B(i,2);
end
end

%yangbenµÚÒ»ÁÐÊÇ·ÖÀàÇÃ½øÈ¥
%bÊÇ´ýÅÐµÄÇÃ½øÈ¥£¬gÇÃ½øÈ¥
%iiiÊÇ¸ÅÂÊ£¬½á¹û
%HÊÇ°óÑé¸ÅÂÊ£¬½á¹û
%g-group·ÖÀàÊý£¬°óÀ´Ð´ÁË¸ö×Ô¶¯¼ì²â·ÖÀàÊýµÄ£¬²»»ýÃ»ÔÚmatlabÏÂÐ©£¬°Ç°Ç
[m,n]=size(yangben);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:g
```

```
groupNum(i)=0;
group(i)=0;
for j=1:m
if yangben(j,1)==i
group(i)=group(i)+1;
end
end
if i==1
groupNum(i)=group(i);
else
groupNum(i)=groupNum(i-1)+group(i);
end
end
group;
groupNum; %¼ÆËã·ÖÀà¸öÊýÊý×é
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%¼ÆËã×ÜÆ½¾ùÖµ
% for j=1:n-1
% TotalMean(j)=0;
% for i=1:m
% TotalMean(j)=TotalMean(j)+yangben(i,j+1);
% end
% TotalMean(j)=TotalMean(j)/m;
% end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%
GroupMean=[];
for i=1:g
if i==1
low=1;
up=groupNum(i);
else
low=groupNum(i-1)+1;
up=groupNum(i);
end
matrix=yangben(low:up,:);
MatrixMean=mean(matrix); %¸÷·ÖÀà×éÆ½¾ùÖµ
GroupMean=[GroupMean;MatrixMean];

for u=low:up
for v=2:n
C(u,v-1)=yangben(u,v)-MatrixMean(v);
end
end
```

```
end

C;
GroupMean;
V=C'*C/(m-g);
V_inv=inv(V); %¶Ô¾ØÕóVÇóÄæ
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
GroupMean=GroupMean(:,2:n);
Q1=GroupMean*V_inv;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:g
lnqi(i)=log(group(i)/m);
mat=GroupMean(i,:);
Q2(i)=lnqi(i)-0.5*mat*V_inv*mat';
end
lnqi;
Q2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%

[u,v]=size(b);
result=[];
for i=1:u
x=b(i,:);
yy=Q1*x'+Q2';
result=[result yy];
end
res=result'; %¼ÆËãµÄ´ýÅÐÊý¾Ý¶Ô¸÷±ê×¼Êý¾ÝµÄÏßÐÔ¼ÆËãÖµ

[rows,cols]=size(result);
for i=1:cols
iljj=0;
mlljj=result(:,i);
for j=1:rows
iljj=iljj+exp(result(j,i)-max(mlljj));
end
for j=1:rows
houyangailv(j,i)=exp(result(j,i)-max(mlljj))/iljj;
end
end
H=houyangailv'; %°óÑé¸ÅÂÊ
```

```
iii=[];
for a=1:u
k=max(H(a,:));
for ii=1:g
if k==H(a,ii)
iii=[iii;ii];
end
end
end

clear ccatagorydetectionijkmn
for i=1:7
    c{i}=[];
end
for i=1:442
    catagory=b(i,1);
for j =1:7
       [m,n]=size(c{j});
if n~=0
          detection=0;
for k =1:n

if c{j}(5,k)==a(i,j)
                c{j}(catagory,k)=c{j}(catagory,k)+1;
                detection=1;
end
end
if detection==0
             c{j}(5,(n+1))=a(i,j);
             c{j}(catagory,(n+1))=(c{j}(catagory,(n+1)))+1;
end

end
if n==0
          c{j}(5,1)=a(i,j);
          c{j}(catagory,1)=(c{j}(catagory,1))+1;
end
end
end
for i=1:7
    c{i}=c{i}';
end
```

12.3 Application Result

Bayes Click	Bayes Convert	Principal Click	Principal Convert	BP Click	BP Convert	Boosting Click	Boosting Convert
0.2625	0.225	0.301081	0.295534	0.047775	0.024434	0.0215	0.0223
0.05	0.05	0.140721	0.143245	0.021181	0.021401	0.0241	0.0223
0.2625	0.225	0.121122	0.11836	0.009838	-0.05456	0.0236	0.0226
0.05	0.225	0.099311	0.099699	0.015475	-0.045	0.0198	0.0226
0.05	0.05	1.166903	1.174594	0.013802	-0.07697	0.0213	0.0228
0.05	0.225	0.058086	0.055002	0.012924	-0.05778	0.0213	0.0226
0.05	0.225	0.248493	0.250987	0.015712	-0.04218	0.0212	0.0232
0.05	0.05	0.706112	0.71033	0.021142	0.023265	0.0241	0.0222
0.05	0.225	0.238837	0.239042	0.016657	-0.03991	0.0212	0.0232
0.05	0.05	0.284154	0.284908	0.021321	0.016654	0.0241	0.0223
0.05	0.05	0.153205	0.151119	0.023743	0.016128	0.0242	0.0226
0.05	0.05	0.838394	0.842787	0.013709	-0.06495	0.0213	0.0226
0.2625	0.225	0.050065	0.036009	0.016702	0.008986	0.0247	0.0226
0.05	0.05	0.03951	0.033058	0.020839	0.018061	0.0241	0.0226
0.05	0.05	0.427411	0.431128	0.019589	0.019448	0.0241	0.022
0.05	0.225	0.061471	0.06089	0.015967	-0.04453	0.0241	0.0226
0.05	0.225	0.061778	0.059916	0.02975	-0.05035	0.0242	0.0226
0.05	0.05	0.106117	0.109004	0.029872	-0.03933	0.0242	0.0232
0.05	0.05	0.058695	0.056355	0.022316	0.018414	0.0242	0.0223
0.05	0.05	0.409615	0.413317	0.01951	0.021002	0.0241	0.022
0.2625	0.225	0.054849	0.047701	0.019698	0.011567	0.0205	0.0225
0.05	0.225	0.45968	0.459352	0.018527	0.016732	0.0202	0.0223
0.05	0.05	0.387575	0.390328	0.019166	0.019707	0.0197	0.0223
0.05	0.225	0.044472	0.041542	0.015017	-0.05861	0.0218	0.0239
0.05	0.05	0.919941	0.925382	0.020089	0.020919	0.0241	0.0228
0.05	0.05	0.710686	0.713729	0.012175	-0.07251	0.0213	0.0224
0.05	0.225	0.029344	0.02205	0.01429	-0.0593	0.0218	0.0226
0.05	0.05	0.121712	0.121506	0.017301	0.013074	0.0241	0.0226
0.05	0.05	0.159957	0.15993	0.021647	0.014211	0.0242	0.0226
0.05	0.225	0.057494	0.057361	0.046611	-0.04982	0.0242	0.0226
0.2625	0.225	0.128441	0.129147	0.049592	0.020074	0.0215	0.0223
0.05	0.05	0.0211	0.018417	0.02409	0.011114	0.0242	0.0225
0.05	0.225	0.267234	0.267913	0.008311	-0.06043	0.0212	0.0232
0.05	0.05	0.11542	0.11341	0.022578	0.016628	0.0242	0.0226
0.05	0.05	0.08274	0.079886	0.019898	0.009123	0.0241	0.0226
0.05	0.05	0.035513	0.035484	0.017282	-0.04392	0.0241	0.0226
0.05	0.225	0.272611	0.272038	0.016109	-0.05984	0.0212	0.0232
0.05	0.05	0.070123	0.06746	0.022535	0.012018	0.0242	0.0226
0.05	0.225	0.056996	0.056187	0.016619	-0.0509	0.0175	0.0226
0.05	0.05	0.311108	0.313147	0.017928	-0.04026	0.0246	0.0213
0.05	0.05	0.046446	0.046241	0.049272	0.012992	0.0242	0.0222

0.05	0.05	0.096495	0.096192	0.019702	0.010709	0.0241	0.0226
0.05	0.225	0.023168	0.021322	0.016456	-0.05952	0.0175	0.0226
0.2625	0.225	0.023472	0.015893	0.01577	-0.05274	0.0215	0.0239
0.05	0.225	0.063036	0.06313	0.041782	-0.05208	0.0242	0.0226
0.05	0.05	0.130617	0.132127	0.020563	0.013422	0.0211	0.0226
0.05	0.05	0.056659	0.055431	0.019311	0.009797	0.0241	0.0226
0.05	0.05	0.153791	0.154382	0.021909	0.015791	0.0242	0.0226
0.05	0.225	0.050062	0.047132	0.014266	-0.05669	0.0218	0.0239
0.05	0.05	0.401322	0.396777	0.019591	0.020679	0.0241	0.0223
0.05	0.05	0.042529	0.039504	0.018259	0.006759	0.0242	0.0226
0.05	0.225	0.046048	0.042978	0.013266	-0.05885	0.0213	0.0239
0.05	0.05	0.072228	0.071846	0.019819	0.012318	0.0241	0.0226
0.2625	0.225	0.008559	0.004133	0.018996	0.019257	0.0195	0.0225
0.05	0.05	0.05594	0.055802	0.020961	0.014394	0.0241	0.0223
0.2625	0.225	0.071697	0.064676	0.02177	0.022783	0.0195	0.0221
0.05	0.225	0.032809	0.029794	0.015003	-0.05685	0.0218	0.0226
0.05	0.05	0.303993	0.303849	-0.00484	0.0218	0.0221	0.0223
0.05	0.225	0.02684	0.023795	0.022054	0.015454	0.0242	0.0225
0.2125	0.225	0.010919	0.014113	0.010625	0.014996	0.0247	0.0225
0.05	0.05	0.063761	0.058907	0.037174	0.012365	0.0242	0.0226
0.05	0.05	0.08451	0.084139	0.020973	0.013408	0.0241	0.0226
0.05	0.05	0.957088	0.966342	0.020351	0.026999	0.0241	0.0228
0.05	0.05	0.079281	0.079612	0.022356	0.014837	0.0242	0.0226
0.2125	0.225	0.007936	-0.00086	0.01403	0.015025	0.0247	0.0225
0.2625	0.225	0.065479	0.051174	0.011993	-0.05492	0.0195	0.0226
0.05	0.05	0.081789	0.082327	0.023207	0.012109	0.0242	0.0226
0.05	0.05	0.211924	0.213195	0.023184	-0.02773	0.0241	0.0232
0.05	0.225	0.037688	0.035776	0.021638	0.018533	0.0242	0.0225
0.05	0.225	0.046328	0.045606	0.02291	0.01285	0.0242	0.0222
0.05	0.225	0.046882	0.043287	0.014918	-0.05514	0.0218	0.0239
0.05	0.225	0.023604	0.020592	0.014003	-0.0574	0.0218	0.0239
0.2125	0.225	0.006932	0.003512	0.008363	0.005225	0.0247	0.0225
0.05	0.225	0.107906	0.101205	0.023197	0.016882	0.0242	0.0225
0.2625	0.349	0.046636	0.031028	0.02256	-0.04799	0.0195	0.0226
0.2625	0.225	0.159246	0.146043	0.021014	0.011773	0.023	0.0226
0.05	0.225	0.017755	0.016308	0.017763	-0.05388	0.0224	0.0239
0.05	0.05	0.172193	0.173618	0.021146	0.016857	0.0241	0.0226
0.2125	0.225	0.006541	-0.00187	0.010914	0.015761	0.0247	0.0225
0.05	0.225	0.019879	0.016972	0.015368	-0.05581	0.0198	0.0239
0.05	0.05	0.012198	0.009536	0.034172	-0.00937	0.0242	0.0239
0.05	0.225	0.01658	0.013583	0.014871	-0.05724	0.0218	0.0239
0.05	0.225	0.043114	0.035716	0.016119	-0.05591	0.0241	0.0239
0.2125	0.225	0.010337	0.001918	0.013426	0.013825	0.0247	0.0225

0.2125	0.225	0.010963	0.01043	0.010482	0.007622	0.0247	0.0225
0.05	0.05	0.043176	0.042843	0.019313	0.009654	0.0241	0.0226
0.2125	0.225	0.007149	0.003611	0.008852	0.008215	0.0247	0.0225
0.05	0.225	0.023931	0.021039	0.01376	-0.06077	0.0213	0.0239
0.05	0.05	0.100603	0.10202	0.023325	0.015937	0.0242	0.0213
0.2625	0.349	0.011219	0.003559	0.015098	-0.0486	0.0215	0.0239
0.2125	0.225	0.003645	0.003773	0.001698	-0.03455	0.0199	0.0239
0.05	0.05	0.066361	0.066042	0.020988	0.013059	0.0241	0.0226
0.2125	0.225	0.005905	-0.00243	0.009273	0.00535	0.0247	0.001
0.05	0.225	0.03728	0.037025	0.025724	-0.04938	0.0242	0.0239
0.2125	0.225	0.00517	0.003057	0.003035	-0.03324	0.0221	0.0239
0.2125	0.225	0.003833	-0.00425	0.001923	-0.03413	0.0221	0.0171
0.05	0.05	0.020376	0.020014	0.020601	-0.05363	0.0241	0.0239
0.2125	0.225	0.010629	0.009626	0.003222	0.008785	0.0221	0.0225
0.2125	0.225	0.005152	0.003957	0.002019	-0.03304	0.0221	0.0239
0.05	0.05	0.013796	0.01065	0.021336	0.016562	0.0241	0.0225
0.2125	0.225	0.007317	0.003736	0.009564	0.006794	0.0247	0.0225
0.05	0.05	0.285532	0.287084	0.020283	0.017855	0.0241	0.0223
0.05	0.05	0.029818	0.029556	0.022125	-0.03342	0.0242	0.0226
0.2125	0.225	0.006412	0.008398	0.012103	0.019711	0.0247	0.0225
0.05	0.225	0.056872	0.057289	0.041746	-0.05331	0.0242	0.0226
0.05	0.225	0.02168	0.013711	0.008839	-0.05388	0.0241	0.0239
0.05	0.225	0.27461	0.268199	0.019176	-0.0553	0.0197	0.0232
0.2125	0.225	0.008016	0.004362	0.005632	0.006494	0.0221	0.0225
0.2125	0.225	0.009297	0.008249	0.002735	0.007893	0.0221	0.0195
0.05	0.05	0.026959	0.026361	0.018757	-0.04092	0.0202	0.0226
0.05	0.05	0.028931	0.025484	0.018529	0.007957	0.0202	0.0226
0.05	0.225	0.048058	0.045066	0.022431	0.016809	0.0242	0.0222
0.2125	0.225	0.011357	0.010473	0.003062	0.008683	0.0221	0.0225
0.2125	0.225	0.008931	0.007993	0.001995	0.00784	0.0221	0.0195
0.05	0.225	0.020309	0.017448	0.013675	-0.0585	0.0213	0.0239
0.2125	0.225	0.007707	0.003987	0.033202	0.011352	0.0234	0.0225
0.05	0.225	0.008246	0.007104	0.018907	-0.05212	0.0202	0.0239
0.05	0.05	0.021423	0.021151	0.023556	-0.03263	0.0242	0.0239
0.2625	0.225	0.029511	0.021986	0.020884	0.012993	0.0195	0.021
0.2125	0.225	0.005237	0.001894	0.000943	-0.03714	0.0221	0.0239
0.05	0.225	0.041358	0.040852	0.01934	-0.0451	0.0241	0.0239
0.05	0.05	0.029407	0.028856	0.02031	0.012507	0.0241	0.0212
0.05	0.225	0.023048	0.02133	0.016487	-0.05954	0.0175	0.0226
0.2125	0.225	0.007227	-0.00093	0.009995	0.006268	0.0247	0.001
0.382	0.349	0.000474	-0.00617	0.029938	-0.03327	0.0226	0.001
0.2125	0.225	0.003404	-0.00492	0.001145	-0.03817	0.0221	0.001
0.2125	0.225	0.005929	-0.00214	0.002944	-0.0385	0.0221	0.001

0.2125	0.225	0.007107	0.006384	-0.00476	-0.02763	0.0214	0.0239
0.2125	0.225	0.007211	0.006589	0.001433	-0.03111	0.0221	0.0239
0.2125	0.225	0.008844	0.008095	-0.00581	-0.02622	0.0214	0.0239
0.2125	0.225	0.008742	0.007927	-0.00664	-0.0261	0.0214	0.0239
0.2125	0.225	0.004588	-0.00359	0.002976	-0.03823	0.0221	0.001
0.2125	0.225	0.009737	0.008954	-0.00538	-0.03678	0.0214	0.0239
0.2125	0.225	0.005842	0.002583	0.00356	-0.03568	0.0221	0.0239
0.2125	0.225	0.009736	0.008953	-0.00537	-0.03649	0.0214	0.0239
0.2125	0.225	0.006041	-0.00238	0.00819	0.005465	0.0247	0.001
0.05	0.225	0.051646	0.049489	0.021027	0.012499	0.0241	0.0226
0.2125	0.225	0.010266	0.00932	-0.00707	-0.03118	0.0214	0.0239
0.2125	0.225	0.003554	-0.00456	0.003841	-0.03775	0.0221	0.001
0.2125	0.225	0.005441	0.00203	0.00256	-0.03863	0.0221	0.0239
0.2125	0.225	0.007262	0.005939	-0.00623	-0.0318	0.0214	0.0239
0.05	0.05	0.020607	0.020204	0.012958	0.010416	0.0213	0.0195
0.05	0.05	0.022177	0.020814	0.018936	0.00952	0.0202	0.0195
0.2125	0.225	0.009875	0.001768	0.009749	0.005862	0.0247	0.001
0.05	0.05	0.114214	0.115638	0.023639	0.015651	0.0242	0.0226
0.2125	0.225	0.01193	0.010972	0.002416	0.006727	0.0221	0.0225
0.2125	0.225	0.009059	0.007769	-0.00621	-0.06008	0.0214	0.0239
0.05	0.05	0.069129	0.068707	0.019155	0.005347	0.0227	0.0226
0.05	0.05	0.212041	0.213253	0.020847	0.017145	0.0241	0.0227
0.2125	0.225	0.00864	0.00916	0.008708	0.019435	0.0247	0.0225
0.2125	0.225	0.006305	-0.00132	0.006431	-0.03179	0.0202	0.001
0.2625	0.225	0.015216	0.011977	0.018604	0.020035	0.0209	0.0225
0.05	0.05	0.022746	0.021435	0.021955	0.010881	0.0242	0.0226
0.2625	0.225	0.004393	-0.00268	0.017402	0.016233	0.0236	0.0225
0.2125	0.225	0.008666	0.000453	0.008176	0.005651	0.0247	0.001
0.05	0.225	0.014505	0.006801	0.014164	-0.05842	0.0218	0.0239
0.05	0.225	0.029027	0.02591	0.014523	-0.05599	0.0218	0.0226
0.2125	0.225	0.007448	0.003982	0.00731	0.006281	0.0221	0.0225
0.2625	0.225	0.018527	0.017602	0.020393	0.020422	0.0195	0.0225
0.05	0.05	0.042026	0.039118	0.017737	0.007654	0.0224	0.0226
0.05	0.225	0.331541	0.333914	0.014941	-0.06625	0.0212	0.0232
0.2125	0.225	0.005455	-0.00266	0.00119	-0.03827	0.0221	0.0149
0.2125	0.225	0.012391	0.011502	0.010113	0.008919	0.0247	0.0225
0.05	0.225	0.014426	0.00058	0.012753	-0.05659	0.0213	0.001
0.05	0.225	0.011038	0.007793	0.015099	-0.05155	0.0218	0.0239
0.2125	0.225	0.007116	0.006546	0.004305	-0.02703	0.0221	0.0239
0.2125	0.225	0.006364	-0.00174	0.000389	-0.03959	0.0221	0.0171
0.05	0.05	0.030969	0.03046	0.016133	-0.09231	0.0241	0.0226
0.05	0.05	0.041989	0.041225	0.024206	0.015106	0.0242	0.02
0.2125	0.225	0.006068	0.002641	0.001369	-0.04098	0.0221	0.0239

0.2125	0.225	0.010283	0.008887	0.002416	0.006727	0.0221	0.0225
0.05	0.225	0.014355	0.013712	0.016826	-0.04724	0.0175	0.0239
0.05	0.225	0.009614	0.006398	0.013621	-0.05867	0.0215	0.0239
0.05	0.05	0.011085	0.010683	0.011712	-0.04207	0.0213	0.0239
0.05	0.05	0.024787	0.024539	0.019065	0.01	0.0202	0.0226
0.2625	0.225	0.01189	0.004122	0.018492	0.018244	0.0239	0.0225
0.2125	0.225	0.01139	0.010615	0.00267	0.008315	0.0221	0.0225
0.2625	0.225	0.01581	0.008043	0.021035	0.014318	0.023	0.0225
0.05	0.05	0.027528	0.026768	0.018935	0.01077	0.0202	0.0226
0.05	0.05	0.146195	0.147953	0.02167	0.014909	0.0242	0.0226
0.05	0.225	0.012646	0.009517	0.01619	-0.05372	0.0241	0.0239
0.05	0.225	0.013892	0.0128	0.016021	-0.05959	0.0241	0.0239
0.2125	0.225	0.007266	0.006005	0.002201	-0.03094	0.0221	0.0239
0.05	0.05	0.021308	0.020514	0.01933	0.008859	0.0241	0.0195
0.05	0.225	0.189112	0.192334	0.015699	-0.06461	0.0241	0.0232
0.05	0.05	0.142136	0.142049	0.015453	-0.04105	0.0225	0.0226
0.05	0.05	0.148518	0.148921	0.021764	0.014453	0.0242	0.0213
0.05	0.05	0.276366	0.264585	0.052117	0.023249	0.0234	0.0223
0.2625	0.225	0.010303	0.002507	0.009749	-0.05589	0.0215	0.0239
0.2125	0.225	0.006215	-0.00817	-0.00032	-0.03466	0.0221	0.001
0.2125	0.225	0.007926	########	0.012253	0.014051	0.0247	0.0225
0.2125	0.225	0.009719	0.009138	0.000374	-0.08663	0.0221	0.0239
0.2125	0.225	0.010238	0.009419	0.000391	-0.02953	0.0221	0.0239
0.2125	0.225	0.006739	0.005964	-0.00528	-0.02343	0.0214	0.0239
0.05	0.05	0.022889	0.022175	0.021974	0.011524	0.0242	0.0226
0.05	0.05	0.021889	0.020806	0.018825	0.009917	0.0202	0.0195
0.05	0.05	0.102282	0.101981	0.019771	0.011515	0.0241	0.0226
0.2125	0.225	0.008776	0.007424	0.001543	-0.02451	0.0221	0.0239
0.2125	0.225	0.012034	0.010843	0.006964	0.011446	0.0221	0.0225
0.2125	0.225	0.012645	0.011759	0.032645	0.014862	0.0234	0.0225
0.05	0.225	0.01525	0.013787	0.020646	0.011275	0.0241	0.0225
0.2125	0.225	0.006153	0.00279	0.003873	-0.04123	0.0221	0.0239
0.2125	0.225	0.005238	-0.00272	0.001981	-0.03653	0.0221	0.001
0.2125	0.225	0.009704	0.001139	0.006595	0.004754	0.0202	0.001
0.2125	0.225	0.007505	0.004021	0.011899	0.011975	0.0247	0.0225
0.2125	0.225	0.004383	0.002958	0.00147	-0.03563	0.0221	0.0239
0.05	0.05	0.016457	0.013194	0.016896	0.019142	0.0241	0.0225
0.05	0.225	0.06849	0.061434	0.021317	0.013423	0.0241	0.0226
0.2125	0.225	0.007851	-0.00034	0.008375	0.005534	0.0247	0.001
0.05	0.225	0.019538	0.016346	0.014395	-0.0515	0.0218	0.0239
0.2125	0.225	0.008866	0.008007	0.002504	0.007833	0.0221	0.0195
0.2125	0.225	0.008521	0.000354	0.008581	0.005284	0.0247	0.001
0.2125	0.225	0.00776	0.004291	0.008908	0.010604	0.0247	0.0135

0.2125	0.225	0.008879	0.005321	0.009061	0.007696	0.0247	0.0189
0.05	0.225	0.01688	0.013538	0.013647	-0.05888	0.0213	0.0239
0.2125	0.225	0.004865	0.001278	0.001244	-0.0387	0.0221	0.001
0.05	0.05	0.073389	0.073277	0.02152	0.013877	0.0242	0.0208
0.2625	0.225	0.069466	0.061968	0.020799	0.019072	0.0195	0.0225
0.2125	0.225	0.005393	-0.00247	0.004086	-0.04003	0.0221	0.001
0.2125	0.225	0.01215	0.011037	0.006082	0.008364	0.0202	0.0225
0.05	0.225	0.02346	0.019896	0.015425	-0.05364	0.0198	0.0239
0.05	0.05	0.01633	0.013126	0.018571	0.009475	0.0202	0.0195
0.05	0.225	0.021668	0.020445	0.017369	0.004842	0.0218	0.021
0.2625	0.225	0.008397	0.005027	0.01419	0.01422	0.0215	0.0225
0.05	0.225	0.022944	0.015037	0.013643	-0.0581	0.0215	0.0239
0.05	0.225	0.018746	0.019121	0.013199	-0.05789	0.0213	0.0239
0.05	0.05	0.042536	0.041844	0.01938	0.011102	0.0241	0.0226
0.2125	0.225	0.007773	0.007257	0.000346	-0.01889	0.0221	0.0239
0.2125	0.225	0.012191	0.011324	0.005658	0.012083	0.0221	0.0225
0.05	0.225	0.045833	0.042801	0.022026	0.016382	0.0242	0.0209
0.05	0.225	0.014539	0.013379	0.03326	-0.05816	0.0242	0.0239
0.05	0.05	1.842754	1.851109	0.02723	-0.01368	0.0215	0.0234
0.2625	0.225	0.01866	0.010826	0.010803	-0.05841	0.0195	0.0239
0.2125	0.225	0.005478	0.002011	-0.00463	-0.03717	0.0214	0.0239
0.2125	0.225	0.005205	0.001767	0.000531	-0.03617	0.0221	0.001
0.2125	0.225	0.008186	3.88E-07	0.01224	0.013073	0.0247	0.0135
0.05	0.225	0.062721	0.060188	0.021876	0.015631	0.0242	0.0226
0.2125	0.225	0.006413	0.002731	0.000125	-0.00961	0.0221	0.0239
0.05	0.05	0.023908	0.023306	0.018539	0.004969	0.0202	0.0226
0.2125	0.225	0.012638	0.011516	0.007637	0.014536	0.0215	0.0225
0.05	0.225	0.052156	0.050381	0.020638	0.008122	0.0241	0.0226
0.2125	0.225	0.008049	-0.00015	0.008539	0.005569	0.0247	0.001
0.2125	0.225	0.010283	0.008887	0.002416	0.006727	0.0221	0.0225
0.05	0.05	0.033099	0.033057	0.017914	-0.04353	0.0224	0.0226
0.05	0.225	0.012346	-0.00199	0.013945	-0.05816	0.0218	0.001
0.05	0.225	0.021146	0.020853	0.04742	0.017029	0.0242	0.0225
0.05	0.225	0.02212	0.018965	0.01802	0.008293	0.0224	0.021
0.05	0.05	0.125575	0.122975	0.02215	0.010401	0.0242	0.0226
0.2625	0.225	0.049448	0.036176	0.020347	0.021365	0.0195	0.0225
0.05	0.05	0.356942	0.359282	0.023274	0.020159	0.0215	0.0223
0.05	0.05	0.168361	0.169295	0.021903	0.015031	0.0242	0.0226
0.05	0.05	0.151001	0.152199	0.02252	0.015042	0.0242	0.0226
0.05	0.05	0.103999	0.105048	0.020955	0.012843	0.0241	0.0222
0.05	0.05	0.844454	0.844337	0.025795	0.000313	0.0215	0.0226
0.2125	0.225	0.005866	-0.00886	0.011111	0.016296	0.0247	0.0225
0.05	0.05	0.10245	0.102947	0.020947	0.010108	0.0241	0.0226

0.05	0.225	0.059872	0.052483	0.011717	-0.05989	0.0213	0.0226
0.05	0.05	0.204978	0.206755	0.023622	0.020508	0.0215	0.0223
0.05	0.05	0.053262	0.052931	0.020776	0.012347	0.0241	0.0226
0.05	0.225	0.025138	0.025226	0.022422	0.017912	0.0242	0.0225
0.05	0.225	0.026865	0.026295	0.017613	-0.04441	0.0224	0.0226
0.05	0.225	0.033728	0.032893	0.021447	-0.05059	0.0242	0.0226
0.05	0.05	0.03963	0.039072	0.022987	-0.06509	0.0242	0.0226
0.2125	0.225	0.005303	-0.00274	0.004478	-0.03539	0.0221	0.001
0.2625	0.349	9.09616	9.118142	0.034491	-0.1052	0.0215	0.0255
0.2625	0.05	0.785604	0.816868	0.035319	-0.04666	0.0215	0.0226
0.05	0.05	0.671339	0.675279	0.019997	0.024817	0.0241	0.0225
0.2625	0.225	0.883821	0.880353	0.010231	-0.06141	0.0212	0.0229
0.2625	0.349	6.919327	6.945776	0.00938	-0.08818	0.0212	0.0255
0.2625	0.225	0.846089	0.842904	0.010162	-0.0601	0.0212	0.0224
0.382	0.349	7.16185	7.179173	0.012326	-0.09859	0.0226	0.0255
0.2625	0.225	0.945303	0.94349	0.009658	-0.06012	0.0235	0.0229
0.2625	0.05	0.373528	0.383608	0.009096	-0.04901	0.0212	0.0223
0.2625	0.225	0.063014	0.05562	0.014447	-0.05349	0.0215	0.0226
0.382	0.349	0.07983	0.06655	0.018882	-0.04924	0.0204	0.0226
0.2625	0.225	0.054785	0.04724	0.014298	-0.05379	0.0218	0.0239
0.05	0.05	0.460305	0.46433	0.021374	0.020999	0.0241	0.0223
0.382	0.349	0.041972	0.0348	0.029855	0.02318	0.0226	0.0225
0.382	0.349	11.4282	11.4552	0.0365	-0.11111	0.0226	0.0255
0.2625	0.225	3.048716	3.057682	0.005921	-0.09288	0.0212	0.0238
0.2625	0.349	12.524	12.5511	0.035871	-0.11862	0.0215	0.0301
0.2625	0.225	1.612981	1.610383	0.001919	-0.05607	0.0212	0.0229
0.05	0.05	0.294202	0.288847	0.046674	0.017109	0.0234	0.0223
0.2625	0.05	1.144992	1.144997	0.018258	0.025127	0.0251	0.023
0.2625	0.225	0.127119	0.11991	0.015408	-0.05705	0.0225	0.0226
0.05	0.225	0.042112	0.039941	0.015921	-0.05273	0.0241	0.0239
0.05	0.225	0.116847	0.113293	0.022334	0.021449	0.0242	0.0226
0.382	0.349	7.549541	7.578432	0.010822	-0.08613	0.0226	0.0255
0.2625	0.05	0.583085	0.595301	0.00789	-0.05132	0.0251	0.0232
0.2625	0.225	0.10881	0.101731	0.008454	-0.05752	0.0236	0.0224
0.2625	0.225	0.101932	0.094958	0.020866	0.020915	0.0206	0.0226
0.2625	0.225	0.113682	0.106418	0.021067	0.013923	0.023	0.0222
0.2625	0.225	0.609268	0.603864	0.010462	-0.05297	0.0212	0.0232
0.05	0.05	0.784498	0.792061	0.013634	-0.05421	0.0215	0.0226
0.05	0.05	5.877811	5.903869	0.010725	-0.04756	0.0212	0.0255
0.2625	0.225	1.346343	1.347062	0.007878	-0.06269	0.0251	0.0229
0.2625	0.349	3.003834	3.015559	0.006	-0.07736	0.0202	0.0238
0.2625	0.225	0.081122	0.067548	0.020647	0.022424	0.0195	0.0221