Machine Learning project: collaborative filtering

Imports ¶

In [1]:
```
import os
import pickle

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split,KFold

%matplotlib inline
```

Random Seed

In [2]:
```
seed=5543
np.random.seed(seed)
```

Data

In [3]:
```
raw_dir="../../raw/ml-1m"
data_dir="../../data/MovieLens"
```

In [4]:
```
if not (os.path.exists(data_dir)):
    os.mkdir(data_dir)
```

In [5]:
```
filename=raw_dir+"/ratings.dat"
```

In [6]:
```
data_all=pd.read_csv(filename,sep="::",header=None,names=["userId","movieId","rating","TimeStamp"])
```

```
/home/manel/anaconda3/lib/python3.6/site-packages/ipykernel_launcher.py:1: ParserWarning: Falling back to the 'python' engine because the 'c' engine does not support regex separators (separators > 1 char and different from '\s+' are interpreted as regex); you can avoid this warning by specifying engine='python'.
  """Entry point for launching an IPython kernel.
```

In [7]:
```
data_all.head()
```

Out[7]:

	userId	movieId	rating	TimeStamp
0	1	1193	5	978300760
1	1	661	3	978302109
2	1	914	3	978301968
3	1	3408	4	978300275
4	1	2355	5	978824291

Train Test Split

In [8]:
```
userEncoder=LabelEncoder()
movieEncoder=LabelEncoder()
```

In [9]:
```
users_all=userEncoder.fit_transform(data_all[["userId"]].values.ravel())
movies_all=movieEncoder.fit_transform(data_all[["movieId"]].values.ravel
())
ratings_all=data_all[["rating"]].values.ravel()
```

In [10]:
```
users,users_test,movies,movies_test,ratings, ratings_test=train_test_spl
it(users_all,movies_all,ratings_all,test_size=0.15)
users_train,users_val, movies_train,movies_val, ratings_train,ratings_va
l=train_test_split(users,movies,ratings,test_size=0.15)
users_train.shape,users_val.shape,users_test.shape
```

Out[10]:
```
((722650,), (127527,), (150032,))
```

In [11]:
```
unkown_users=~np.isin(users_val,users)
unknown_movies=~np.isin(movies_val,movies)
print(unkown_users.sum(),unknown_movies.sum())
```

```
0 0
```

In [12]:
```
unique_users=userEncoder.classes_
unique_movies=movieEncoder.classes_
```

In [13]:
```
N_users=len(unique_users)
N_movies=len(unique_movies)
print(N_users,N_movies)
```

```
6040 3706
```

Collaborative Filter

Mean Rating

Mean rating over the training set is

$$\mu = \frac{1}{N_{\mathcal{T}}} \sum_{(i,u)\in\mathcal{T}} r_{u,i}$$

And, we define the differential rating

$$\Delta r_{u,i} = r_{u,i} - \mu$$

In [14]:
```
mu=ratings_train.mean()
drating=np.mean((ratings_train-mu)**2)
print(mu,drating)
```

```
3.5812398809935653 1.2487443009293984
```

Parameter Initialization

We implement the following initialization

$$b_u^0 \sim \mathcal{N}(0, 10^{-4})$$
$$b_i^0 \sim \mathcal{N}(0, 10^{-4})$$
$$p_{u,f}^0 \sim \mathcal{N}\left(0, \frac{1}{\max(1, \sqrt{F})}\right)$$
$$q_{i,f}^0 \sim \mathcal{N}\left(0, \frac{1}{\max(1, \sqrt{F})}\right)$$

In [15]:
```
def initialize_params(F,N_users,N_movies):
    b_users=np.random.normal(0,0.0001,N_users)
    b_movies=np.random.normal(0,0.0001,N_movies)
    p_users=np.random.normal(0,1/max(1,np.sqrt(F)),(N_users,F))
    p_movies=np.random.normal(0,1//max(1,np.sqrt(F)),(N_movies,F))
    return b_users,b_movies,p_users,p_movies
```

In [17]:
```
F=2
```

In [18]:
```
b_users,b_movies,p_users,p_movies=initialize_params(F,N_users,N_movies)
params=[mu,b_users,b_movies,p_users,p_movies]
```

Rating Model

The model prediction for r is given by

$$\hat{r}_{i,u} = \mu + b_u + b_i + p_u^T q_i$$

In [19]:
```
def predict_rating(users,movies,params):
    mu,b_users,b_movies,p_users,p_movies=params
    b_u=b_users[users]
    b_m=b_movies[movies]
    p_u=p_users[users]
    p_m=p_movies[movies]
    prod=np.sum(p_u*p_m,axis=1)
    r_hat=mu+b_u+b_m+prod
    return r_hat
```

In [20]:
```
R=predict_rating(users_train,movies_train,params)
R[:10]
```

Out[20]:
```
array([3.5811178 , 3.58137423, 3.58115786, 3.58146872, 3.58122582,
       3.58112864, 3.58093923, 3.58123367, 3.58094814, 3.58119053])
```

Loss Function

The lost function is

$$L(\theta;\{r\}) = \frac{1}{N_S}\sum_{u,i\in S}\left(r_{u,i}-\hat{r}_{u,i}\right)^2$$

In [21]:
```
def rating_error(users,movies,rating,params0):
    dr=rating-predict_rating(users,movies,params0)
    return np.mean(dr**2)
```

In [22]:
```
rating_error(users_train,movies_train,ratings_train,params)
```

Out[22]:
```
1.248742966734941
```

Learning Step

We will implement a step of stochastic gradient descent as

$$\begin{aligned}
b_u &\leftarrow b_u + \gamma\Delta r_{u,i}\\
b_i &\leftarrow b_i + \gamma\Delta r_{u,i}\\
p_u &\leftarrow p_u + \gamma\left(q_i\Delta r_{u,i} - \lambda p_u\right)\\
q_i &\leftarrow q_i + \gamma\left(p_u\Delta r_{u,i} - \lambda q_u\right)
\end{aligned}$$

In [23]:
```
def learning_step(user,movie,rating,parms0,penalty,batch_size):
    N=len(rating)
    mu,b_users,b_movies,p_users,p_movies=parms0
    perm=np.random.permutation(len(rating))
    for i1 in range(0,N,batch_size):
        idx=perm[i1:i1+batch_size]
        u=user[idx]
        m=movie[idx]
        r=rating[idx]
        b_u=b_users[u]
        b_m=b_movies[m]
        p_u=p_users[u]
        p_m=p_movies[m]
        prod=np.sum(p_u*p_m,axis=1)
        r_hat=mu+b_u+b_m+prod
        dr=r-r_hat
        b_users[u] +=learning_rate*(dr)
        b_movies[m]+=learning_rate*(dr)
        p_users[u] +=learning_rate*(dr[:,np.newaxis]*p_m-penalty*p_u)
        p_movies[m]+=learning_rate*(dr[:,np.newaxis]*p_u-penalty*p_m)
    return
```

Training Function

Given the hyperparameters we just train for a fixed number of epochs

In [24]:
```
def fit_ratings(users_train,movies_train,ratings_train,users_val,movies_
val,ratings_val,
                F,learning_rate,penalty,steps,batch_size):

    mu=ratings_train.mean()
    b_users,b_movies,p_users,p_movies=initialize_params(F,N_users,N_movi
es)
    parms=[mu,b_users,b_movies,p_users,p_movies]
    for i1 in range(steps):
        loss=rating_error(users_train,movies_train,ratings_train,parms)
        learning_step(users_train,movies_train,ratings_train,parms,penal
ty,batch_size)
        if i1 % (steps//10)==0:
            val_loss=rating_error(users_val,movies_val,ratings_val,parms
)
            print("\t",i1,loss,val_loss)
    loss=rating_error(users_train,movies_train,ratings_train,parms)
    val_loss=rating_error(users_val,movies_val,ratings_val,parms)
    print("\tFinal",loss,val_loss)
    return val_loss,parms
```

Model Hyper-parameters

In [25]:
```
learning_rate=0.005
penalty=0.1
batch_size=50
steps=200
```

Train Popularity Model

In [26]:
```
loss,params=fit_ratings(users_train,movies_train,ratings_train,
           users_val,  movies_val, ratings_val,
          0,learning_rate,penalty,10,batch_size
          )
```

```
0 1.248746642725307 0.8909740526539852
1 0.8789293209282619 0.8551414073649105
2 0.8413121945710145 0.8422269447426097
3 0.827094512576697 0.8356778554860488
4 0.8199053667259332 0.831931600744101
5 0.8155451055453977 0.8298184162152527
6 0.8130133859324009 0.8284141083235405
7 0.8111815395205997 0.8274597323768608
8 0.8099879740889979 0.8269773267236272
9 0.8090191069252736 0.8264105330218547
Final 0.8081858378145389 0.8264105330218547
```

Train Model with interaction Term

Here we asume the embedding space has dimension $F = 2$.

In [ ]:
```
loss,params=fit_ratings(users_train,movies_train,ratings_train,
           users_val,  movies_val, ratings_val,
          F,learning_rate,penalty,steps,batch_size,
          )
```

```
0 1.2487413272977004 0.8919647278968952
20 0.8005053824001939 0.825574098155686
40 0.788783145543379 0.8150789791241544
60 0.758732024001011 0.7911134361845616
80 0.7444282606397932 0.781016275974047
100 0.737458723836546 0.7765548494867625
120 0.7336375008227668 0.7746195277755271
140 0.7312443829111536 0.7729298978347575
160 0.7299435936606753 0.7722617994295322
180 0.7288682180958788 0.7721030687230466
```

Colaborative Filter Model

We grap the hyperparameters, training and prediction in a single model for ease of use.

In [32]:
```
class Recommender:
    def __init__(self,F,penalty,learning_rate,steps,batch_size):
        self.F=F
        self.penalty=penalty
        self.learning_rate=learning_rate
        self.steps=steps
        self.batch_size=batch_size
    def fit(self,users,movies,ratings, users_val,movies_val,ratings_val
):
        loss,params=fit_ratings(users,movies,ratings,
                                users_val,movies_val,ratings_val,
                                self.F,self.learning_rate,self.penalty,s
elf.steps,self.batch_size
                               )
        self.params=params
        return loss
    def predict(users,movies):
        return predict_ratings(users,movies,self.params)
```

In [33]:
```
model=Recommender(F=5,penalty=0.1,learning_rate=0.05,steps=10,batch_size
=50)
model.fit(users_train,movies_train,ratings_train,users_val,movies_val,ra
tings_val)
```

```
 0 1.2487491435047675 0.890884965767691
 1 0.87400637899907 0.856136779916064
 2 0.8350615442590369 0.8431626271230641
 3 0.8203097886918276 0.8367159809528412
 4 0.8126704877072 0.8329638983922121
 5 0.8082523436197229 0.8307115543495848
 6 0.8053462488336441 0.8291730605342861
 7 0.8032047189870908 0.8278535652366308
 8 0.8017586224206004 0.8271560419529516
 9 0.8006674468903492 0.8265487010644992
Final 0.7997206726910681 0.8265487010644992
```

Out[33]: 0.8265487010644992

Parameter Search

We do a grid search using a single validation set to find the range of penalties and embedding dimension that seems to perform best

In [34]:
```
results=[]
best_loss=1e10
best_F=None
best_penalty=None

if True:
  for F in [1,5,10,20,30,50,100,150]: #[0,1,5,10,20,30,50,75,100,125,15
0]:
    for penalty in [0,0.01,0.05,0.1,0.15,0.2,1]:
        print()
        print(f"F {F}, penalty {penalty} :")
        model=Recommender(F,penalty,learning_rate,steps,batch_size)
        loss=model.fit(users_train,movies_train,ratings_train,
                               users_val,movies_val,ratings_val)
        results.append((F,penalty,loss))
        if loss<best_loss:
            best_loss=loss
            best_F=F
            best_penalty=penalty
        print()
        print(f"==> {F},{penalty},{loss} == best ({best_F},{best_penalt
y},{best_loss}) =============")
```

```
F 1, penalty 0 :
         0 2.2297649479240245 0.9591224199407644
         20 0.800027721408294 0.8317069799262912
         40 0.776291919894767 0.8106944190050412
         60 0.7555207747682228 0.7926947065372381
         80 0.7469745213146038 0.7863286922835332
         100 0.7430728667900206 0.7839665374894451
         120 0.7411201514829009 0.7831638351470969
         140 0.7400930048694438 0.7828983985853032
         160 0.7395459726436394 0.7827856116910031
         180 0.7392379293105136 0.7825441717557904
        Final 0.7389403532602861 0.7826964523321334

==> 1,0,0.7826964523321334 == best (1,0,0.7826964523321334) ===========
==

F 1, penalty 0.01 :
         0 2.2485388893105505 0.9581940121057958
         20 0.7997113100616658 0.8307259255153069
         40 0.7747837595677601 0.8100976893921636
         60 0.7543478579093974 0.7921530045357693
         80 0.7459384269882637 0.7854553987794991
         100 0.7425184607850186 0.783650513294766
         120 0.7408652896808117 0.7822694239538839
         140 0.7399139852292305 0.7821140811268591
         160 0.7394934117556343 0.7825503539970973
         180 0.7392263428227608 0.7825239230733337
        Final 0.7390170296690197 0.7821276975203693

==> 1,0.01,0.7821276975203693 == best (1,0.01,0.7821276975203693) =====
========

F 1, penalty 0.05 :
         0 2.2154831088778963 0.9592911130367108
         20 0.8010559514385066 0.827713097427781
         40 0.7763979811664843 0.805743980172642
         60 0.7552615205972275 0.7885587343816802
         80 0.7484763475861241 0.7836413872939609
         100 0.7456704003100135 0.7821384363287186
         120 0.744411498398097 0.7818650067083212
         140 0.7437123746179637 0.7814440972958807
         160 0.7432726878589414 0.7812729295808487
         180 0.7430652543075422 0.7811696781374082
        Final 0.7428323524085589 0.7812576068726788

==> 1,0.05,0.7812576068726788 == best (1,0.05,0.7812576068726788) =====
========

F 1, penalty 0.1 :
         0 2.2238204163503 0.9478642833671755
         20 0.8023372863511309 0.825990461087711
         40 0.7952462349916753 0.8194818869162359
         60 0.7704335352513414 0.7986592487537698
         80 0.7607881702006991 0.791459277471889
         100 0.7570206933366584 0.789237537778573
         120 0.7552217021813425 0.7883569974231505
         140 0.7542102301979872 0.788084633842837
```

```
         160 0.7537257323472751 0.7876715352542916
         180 0.7534208149985432 0.787475195107407
        Final 0.7532828456768452 0.7870650934155233

==> 1,0.1,0.7870650934155233 == best (1,0.05,0.7812576068726788) ======
=======

F 1, penalty 0.15 :
         0 2.1972671305472855 0.9408464876920525
         20 0.8036870697188484 0.825575206600962
         40 0.8031610810499179 0.8249096699469567
         60 0.8023830168563719 0.8242695362246438
         80 0.7944909112024917 0.8177701056951823
         100 0.7808702922204313 0.8063818483269998
         120 0.7746525614049281 0.8020197479853187
         140 0.7719123283217675 0.8003715363070045
         160 0.7705981995990429 0.7990076731980669
         180 0.769741823521016 0.7984729990627741
        Final 0.7692574607756447 0.7983547788230113

==> 1,0.15,0.7983547788230113 == best (1,0.05,0.7812576068726788) =====
========

F 1, penalty 0.2 :
         0 2.318401155028862 0.9367146534668661
         20 0.8044410973092228 0.825674844305247
         40 0.80427135154376 0.8252772603169289
         60 0.8042330354191863 0.8253786390501066
         80 0.8041665942454344 0.8250016950417323
         100 0.8038152413638864 0.8246554318126814
         120 0.8026995911329999 0.82386599786352
         140 0.800623872346066 0.821969484423668
         160 0.7974916265071534 0.8196273008111714
         180 0.7944235149495998 0.8174307823014593
        Final 0.7922052180920206 0.8153716991062978

==> 1,0.2,0.8153716991062978 == best (1,0.05,0.7812576068726788) ======
=======

F 1, penalty 1 :
         0 2.301518634155183 0.9003442864679203
         20 0.805855367153703 0.8251197309277751
         40 0.8050430325841529 0.8253119194574319
         60 0.8048591277205432 0.8249461848285673
         80 0.8048007035044847 0.82534171920178
         100 0.8047975003374801 0.8252486347395471
         120 0.8047427764378186 0.825598447404809
         140 0.8046922063729888 0.8251386539839692
         160 0.8046988186569561 0.8252105832873753
         180 0.8046784095189267 0.8252470047666113
        Final 0.8046907501460729 0.825193075419845

==> 1,1,0.825193075419845 == best (1,0.05,0.7812576068726788) =========
====

F 5, penalty 0 :
         0 1.2487513098452907 0.8915926124746368
```

```
          20 0.7506424305916863 0.8153694716922872
          40 0.6943352524587739 0.7871835251707676
          60 0.6642455362651379 0.771925055017974
          80 0.6480471883124649 0.7647022734694693
          100 0.6388165598400871 0.7616883659423483
          120 0.6332565178078414 0.7614448223547893
          140 0.6297029368799155 0.7614210290694395
          160 0.6275466993979153 0.7621575866934956
          180 0.625855098790366 0.7629954759541018
         Final 0.6246959958656599 0.7642029964580934

==> 5,0,0.7642029964580934 == best (5,0,0.7642029964580934) ===========
==

F 5, penalty 0.01 :
          0 1.2487471642256454 0.8913347692321165
          20 0.745417471660544 0.8052055596429336
          40 0.6878427422027967 0.7771097871169232
          60 0.6588567253453322 0.7637190211278647
          80 0.6450334735129593 0.7587053013599393
          100 0.6380143657302778 0.7571417785624966
          120 0.6338335818549357 0.7571598558959675
          140 0.6310284123331791 0.7573869648719568
          160 0.6291638025682107 0.7576034409891262
          180 0.6277421351361611 0.7581507480165466
         Final 0.6265942737301966 0.7586331497730149

==> 5,0.01,0.7586331497730149 == best (5,0.01,0.7586331497730149) =====
========

F 5, penalty 0.05 :
          0 1.248743614722122 0.8916199562710234
          20 0.7602456006751701 0.8011332018834963
          40 0.7066383655421384 0.7707269348657033
          60 0.6766142169406445 0.7546132825788663
          80 0.6617551487232808 0.7471239719436471
          100 0.6540292642529685 0.7437805504451162
          120 0.649665849129343 0.7425111804129159
          140 0.6469037317976355 0.7419436873911002
          160 0.6451590158261216 0.7413883024922222
          180 0.643827126221101 0.7414954343593027
         Final 0.6427597259350497 0.7413946506514998

==> 5,0.05,0.7413946506514998 == best (5,0.05,0.7413946506514998) =====
========

F 5, penalty 0.1 :
          0 1.248750834534805 0.8909989336714238
          20 0.7921305465479733 0.8192784616287985
          40 0.7553338817526215 0.7900692906826368
          60 0.7320501263929896 0.7751299967381253
          80 0.7180379035725223 0.7679104126716334
          100 0.7080528559675853 0.7628322861819679
          120 0.7009286314146501 0.7592666825690908
          140 0.6962267260770795 0.7570413163956453
          160 0.6931283504039379 0.7558387808165838
          180 0.6908524720008542 0.7546239608284016
```

```
        Final 0.6892094708683462 0.7540677904376858

==> 5,0.1,0.7540677904376858 == best (5,0.05,0.7413946506514998) ======
=======

F 5, penalty 0.15 :
         0 1.2487415989455257 0.891110341815496
         20 0.8012614803225682 0.8236400795117084
         40 0.7892336451532119 0.8132856060091348
         60 0.7745890592755682 0.801845357809653
         80 0.7675968672008849 0.7973086935079906
         100 0.7627834574796257 0.7941313320099157
         120 0.7593414692794843 0.7919875124437726
         140 0.7571181545729035 0.7906875362638748
         160 0.755324890904747 0.7898477555518162
         180 0.7540532467716009 0.7889422594190699
        Final 0.752653457400857 0.7882177673270312

==> 5,0.15,0.7882177673270312 == best (5,0.05,0.7413946506514998) =====
========

F 5, penalty 0.2 :
         0 1.248745521744386 0.8906877343115223
         20 0.8040482920103087 0.8249552551358608
         40 0.8040804033879162 0.8248379188645961
         60 0.8039663108869334 0.8246300192951864
         80 0.8032265098608842 0.8242920633978179
         100 0.8015708996806268 0.8227151348586291
         120 0.7986637475441465 0.8204833526029717
         140 0.7953231714975244 0.8178644468996659
         160 0.79258090252767 0.8158371048233107
         180 0.790827869911779 0.8147536653799783
        Final 0.7897320915588566 0.8139628716024588

==> 5,0.2,0.8139628716024588 == best (5,0.05,0.7413946506514998) ======
=======

F 5, penalty 1 :
         0 1.2487486354202892 0.8905770827954073
         20 0.8057626259848789 0.8249262516408569
         40 0.8050424492906815 0.8246865403722264
         60 0.8048941633160629 0.824982541411458
         80 0.8048000141402555 0.8253664853638787
         100 0.8047182979411417 0.825434164905228
         120 0.804746532012445 0.8253947116797614
         140 0.8046820601401693 0.8251977840688569
         160 0.8047160739406242 0.8253092936742784
         180 0.8047937635013857 0.8253159085385179
        Final 0.8046890140664014 0.825244826968932

==> 5,1,0.825244826968932 == best (5,0.05,0.7413946506514998) =========
====

F 10, penalty 0 :
         0 1.2487434304128915 0.8909685075195656
         20 0.6994708244748991 0.8077496062483053
         40 0.6195844415236269 0.7884892235234524
```

```
          60 0.5852698712227153 0.7861041347411769
          80 0.5678322612806114 0.7880309905321662
          100 0.557953596979457 0.7920196461534071
          120 0.5517125395907355 0.79657186339414
          140 0.5476926159708011 0.8011968864532487
          160 0.5447004981543985 0.8052805572848525
          180 0.5422889775109342 0.8094709525557573
         Final 0.5406301359755135 0.8130560846720991

==> 10,0,0.8130560846720991 == best (5,0.05,0.7413946506514998) =======
======

F 10, penalty 0.01 :
          0 1.2487408423102824 0.8907739173228147
          20 0.7029405081481221 0.8001532327775172
          40 0.6188467937531164 0.7765614941920417
          60 0.5862814339541935 0.7739454484155872
          80 0.5702575698116711 0.7755615706152033
          100 0.5610566570673127 0.778418043204533
          120 0.5551455234811868 0.7805271846391059
          140 0.5510285430296783 0.782134131019194
          160 0.5480546333758612 0.7846154863500892
          180 0.5458037218894956 0.7863915095668054
         Final 0.5439875140230992 0.787761535071537

==> 10,0.01,0.787761535071537 == best (5,0.05,0.7413946506514998) =====
========

F 10, penalty 0.05 :
          0 1.248740107738926 0.8910287390863945
          20 0.7333265710407397 0.7893641487140519
          40 0.6562368015805509 0.7542763317202331
          60 0.6219045687058006 0.7446058857526998
          80 0.6047251851952693 0.7409398085085032
          100 0.5947424954040679 0.7395964598606629
          120 0.5883028067069098 0.7386363412060822
          140 0.5839921380513079 0.7383020911335697
          160 0.5808736968520171 0.7381772728320816
          180 0.5785264438366965 0.738116763504894
         Final 0.5767074601239571 0.7379060900644814

==> 10,0.05,0.7379060900644814 == best (10,0.05,0.7379060900644814) ===
==========

F 10, penalty 0.1 :
          0 1.2487493406117722 0.8908025169038574
          20 0.776380235560005 0.8070710501129074
          40 0.7415978260060835 0.7828808411040271
          60 0.7157515132903187 0.7677501110080208
          80 0.6982520489849748 0.7595137678040196
          100 0.6857016660755522 0.7546334617549899
          120 0.6764455215434134 0.7515113994723363
          140 0.6697774916655527 0.7495602594936618
          160 0.6649617459341632 0.7480701816554518
          180 0.6614042504276136 0.7472841379692863
         Final 0.6589744514124978 0.7473743897333338
```

```
==> 10,0.1,0.7473743897333338 == best (10,0.05,0.7379060900644814) ====
=========

F 10, penalty 0.15 :
         0 1.2487437319669095 0.8912221015500905
         20 0.8003144781079904 0.8236475535012744
         40 0.7865758967102483 0.8106407698100814
         60 0.7732662629074992 0.8011108006937299
         80 0.7668035923614558 0.7964951402093936
         100 0.7620263961054811 0.7938447236722359
         120 0.7582433747232095 0.7912534196638883
         140 0.755452041168547 0.7897630699872409
         160 0.7532365073575908 0.78880158862176
         180 0.7514676451281093 0.7878286186539085
        Final 0.7498900036476729 0.7869184175973235

==> 10,0.15,0.7869184175973235 == best (10,0.05,0.7379060900644814) ===
==========

F 10, penalty 0.2 :
         0 1.2487418011526814 0.8908189380259148
         20 0.8040060764067867 0.8249668490778914
         40 0.8039912808580106 0.8251124972111745
         60 0.8038741046067422 0.8243883636624083
         80 0.8030322284613356 0.8238302193117605
         100 0.8010281136489429 0.8227922404726449
         120 0.7980500324159354 0.8200725227835295
         140 0.7947121269189724 0.8175319305763245
         160 0.7922332137356777 0.8157312100951482
         180 0.7905859737868868 0.814445441455932
        Final 0.7895844798386167 0.8138667283933928

==> 10,0.2,0.8138667283933928 == best (10,0.05,0.7379060900644814) ====
=========

F 10, penalty 1 :
         0 1.2487410186472037 0.8905847319890536
         20 0.8057276713894312 0.8251346949868907
         40 0.8049964619669907 0.8252331044265948
         60 0.8048027765641171 0.8252892083201053
         80 0.8047830765965948 0.8251495271027721
         100 0.8046850559437471 0.8254641655462872
         120 0.8047943318500226 0.8252341890981867
         140 0.8047074228559737 0.8254139400121977
         160 0.8047132559896168 0.8251865969201847
         180 0.8046912033053225 0.8258003938323721
        Final 0.8045448319042313 0.82527841323026

==> 10,1,0.82527841323026 == best (10,0.05,0.7379060900644814) ========
=====

F 20, penalty 0 :
         0 1.2487426237758974 0.8906897443321168
         20 0.6189473996525631 0.8116125080904204
         40 0.5164342636228899 0.827931207283857
         60 0.4777060171482512 0.8499350560530167
         80 0.4583499538309665 0.8701925422934413
```

```
          100 0.4467922310819076 0.8873112199733059
          120 0.43912508742621065 0.9029208696311046
          140 0.43362245872622907 0.9168985333400173
          160 0.4293803477797884 0.9295267293963125
          180 0.42601645003310684 0.9419310667687372
         Final 0.42330050983734413 0.9528084120972611

==> 20,0,0.9528084120972611 == best (10,0.05,0.7379060900644814) ======
=======

F 20, penalty 0.01 :
          0 1.2487430853275587 0.8905050085869356
          20 0.6293286443253182 0.7909381815538156
          40 0.5222687532311994 0.7961643912324871
          60 0.4830380889400786 0.8115841953594003
          80 0.4639321792414927 0.8250712723467709
          100 0.452727076450459 0.836091454020059
          120 0.4452152205422636 0.8452397254114514
          140 0.4398292345985746 0.8531430215940524
          160 0.4358352794275543 0.8599043780116268
          180 0.4326759181803305 0.8655419088123664
         Final 0.4300770142952947 0.8706731100132421

==> 20,0.01,0.8706731100132421 == best (10,0.05,0.7379060900644814) ===
==========

F 20, penalty 0.05 :
          0 1.2487388985584869 0.8906864995249594
          20 0.7099557456941543 0.78183994881675
          40 0.6058272362616978 0.7479218711624952
          60 0.5562901377787876 0.7426206803899353
          80 0.5319971182944131 0.7431323406829689
          100 0.5182691190182459 0.7447049983399231
          120 0.5098512493511551 0.7458146373360168
          140 0.5040638555737377 0.7471261416821977
          160 0.49983651846355626 0.7478518749382053
          180 0.4966877000054615 0.7489687621127904
         Final 0.49438192126669606 0.7493405999353476

==> 20,0.05,0.7493405999353476 == best (10,0.05,0.7379060900644814) ===
==========

F 20, penalty 0.1 :
          0 1.248744543126167 0.8905281715178839
          20 0.7791465570828208 0.8103223379127166
          40 0.7308563639573025 0.77695185588258
          60 0.7026162397285886 0.7626832404593293
          80 0.6824995020324794 0.7552237199038255
          100 0.667842035493834 0.7506159234480408
          120 0.6573777452251371 0.748220791659236
          140 0.6493408238060101 0.7467576035568825
          160 0.6432394849888388 0.7456600688989157
          180 0.6385528045635731 0.7452131519705636
         Final 0.6347487332040178 0.7447361932087986

==> 20,0.1,0.7447361932087986 == best (10,0.05,0.7379060900644814) ====
=========
```

```
F 20, penalty 0.15 :
         0 1.2487483919353888 0.8906573122450568
         20 0.8006695013697439 0.8239714153664801
         40 0.7888888345073427 0.8128550172416136
         60 0.7737844512242134 0.8016720360246786
         80 0.7666596322490745 0.796705966252126
         100 0.7617499245098531 0.7937058475442704
         120 0.7580122923857459 0.7913184157360154
         140 0.755209455386167 0.789751739387341
         160 0.7529697877390672 0.7885046526460485
         180 0.7512232893876442 0.7876531535497218
        Final 0.7494833714520555 0.7871382703379578

==> 20,0.15,0.7871382703379578 == best (10,0.05,0.7379060900644814) ===
==========

F 20, penalty 0.2 :
         0 1.2487436335171171 0.8905635296386092
         20 0.8037683992568161 0.8247766896870127
         40 0.8036946812016297 0.8248602346340086
         60 0.8031160125871287 0.8241423760043378
         80 0.8014483878117479 0.8225949772967364
         100 0.7985561178286129 0.820219715751625
         120 0.7952201747973894 0.8182571496084989
         140 0.7925443982804365 0.815799545227218
         160 0.7907467870849266 0.8146121740045291
         180 0.7897267114617266 0.8138594060632135
        Final 0.7890506324491433 0.8135564550153783

==> 20,0.2,0.8135564550153783 == best (10,0.05,0.7379060900644814) ====
=========

F 20, penalty 1 :
         0 1.2487421974511048 0.89071870747655
         20 0.8057205202014159 0.8250772159498577
         40 0.8050283798009238 0.8249667987955933
         60 0.8049081625168565 0.825491485057861
         80 0.8048314885189596 0.8250490026681092
         100 0.804680243185528 0.8253760518071959
         120 0.8046997393273124 0.8252075649161376
         140 0.8046499251387409 0.825178165963112
         160 0.8046542614360563 0.8252647663653759
         180 0.8046584866160785 0.8253665584366295
        Final 0.8046698946122812 0.8252896711797886

==> 20,1,0.8252896711797886 == best (10,0.05,0.7379060900644814) ======
=======

F 30, penalty 0 :
         0 1.2487404978608594 0.8911126758405208
         20 0.5646969692556615 0.8202120378803431
         40 0.4427544289049681 0.869161523380187
         60 0.3981217835864418 0.9111158106831609
         80 0.37588032035731334 0.9459062918454698
         100 0.36258952998339944 0.9765851557863413
         120 0.3537032208016508 1.0025413102219385
```

```
          140 0.3471894608570364 1.025512089003799
          160 0.3422890417820217 1.0463367157344274
          180 0.3383783546066635 1.0658894703985182
         Final 0.33528237154528345 1.0832290304915235

==> 30,0,1.0832290304915235 == best (10,0.05,0.7379060900644814) ======
=======

F 30, penalty 0.01 :
          0 1.2487428510543386 0.8904681066673268
          20 0.5847297432272531 0.7935230809208715
          40 0.45487988826260467 0.8216609834940338
          60 0.40873657313014394 0.8491557242462137
          80 0.38610229762643267 0.870232220782774
          100 0.37264834806959934 0.8865629799540112
          120 0.3636555466016144 0.9000983857361838
          140 0.3572439803751318 0.9115758789385032
          160 0.35243369740026004 0.9207825701785373
          180 0.34865430332673036 0.928797768457507
         Final 0.3454936528775615 0.935764351444119

==> 30,0.01,0.935764351444119 == best (10,0.05,0.7379060900644814) ====
=========

F 30, penalty 0.05 :
          0 1.2487395606555187 0.8905031507624955
          20 0.6933827999988861 0.7759309833636263
          40 0.5723023919511411 0.7451693993734535
          60 0.5133852823861236 0.7440709021034572
          80 0.48379851990033496 0.746966112806605
          100 0.4672444250259083 0.7493566174369233
          120 0.4570062355486325 0.7510252682809034
          140 0.4500144271759772 0.7522282767065065
          160 0.4452273110456512 0.7531220104809331
          180 0.44159831488227125 0.7537590689318107
         Final 0.43884350132009925 0.754461070578136

==> 30,0.05,0.754461070578136 == best (10,0.05,0.7379060900644814) ====
=========

F 30, penalty 0.1 :
          0 1.2487446248726406 0.8909384873524869
          20 0.7742305927398241 0.807084338395214
          40 0.7278433921644455 0.776157558079509
          60 0.6963021088637338 0.7604671618590487
          80 0.6753019766130699 0.7527951714620987
          100 0.6597945053553519 0.7482205712375792
          120 0.6482273275847833 0.7456803904610756
          140 0.6390978202012558 0.7444406217352825
          160 0.6322937127416084 0.7436725415615751
          180 0.6268280116211942 0.7431289219881031
         Final 0.6225030690716002 0.7427786502386162

==> 30,0.1,0.7427786502386162 == best (10,0.05,0.7379060900644814) ====
=========

F 30, penalty 0.15 :
```

```
          0 1.248749988077731 0.8904251380144269
          20 0.7998609069053167 0.8233361430546988
          40 0.7858199611083061 0.8107513549133909
          60 0.7726030490087592 0.8010114138666584
          80 0.7662146355707725 0.7966668382012462
          100 0.7613529488490387 0.7933361502350561
          120 0.7574094785033413 0.7911130649045608
          140 0.7545353711751905 0.7897393544254037
          160 0.7523937539139031 0.7881783852414839
          180 0.7504315433909184 0.7872208902165082
         Final 0.7488907030388603 0.786445539675361

==> 30,0.15,0.786445539675361 == best (10,0.05,0.7379060900644814) ====
=========

F 30, penalty 0.2 :
          0 1.2487522393421222 0.8905328105132431
          20 0.8037446735482642 0.8249117771666875
          40 0.8037508748592033 0.8249478694013661
          60 0.8031915185196805 0.8242578729218508
          80 0.8016848324100355 0.8228608081565018
          100 0.7989143868630458 0.8206365684707014
          120 0.7955031558936627 0.81835579678535
          140 0.7928473737427265 0.8161466468860983
          160 0.790840116788553 0.8151832992292947
          180 0.7897834819367895 0.8142192521239794
         Final 0.7890318655502758 0.8136820700914705

==> 30,0.2,0.8136820700914705 == best (10,0.05,0.7379060900644814) ====
=========

F 30, penalty 1 :
          0 1.2487412366753075 0.8906960795879335
          20 0.8058040519459043 0.825043099172956
          40 0.8050495889726083 0.8248728986125821
          60 0.8049416350242505 0.8250945816940364
          80 0.80489820140616 0.8249364288824914
          100 0.8048025843721915 0.825276497311297
          120 0.8046184461104241 0.8254183239832271
          140 0.8047345897369383 0.8255665889122797
          160 0.8046196687713608 0.8253718767941799
          180 0.804630000696624 0.8255421533879798
         Final 0.8047147138763335 0.8256120978939673

==> 30,1,0.8256120978939673 == best (10,0.05,0.7379060900644814) ======
=======

F 50, penalty 0 :
          0 1.2487411830019288 0.8905056743631145
          20 0.4857124419110016 0.8335748755892876
          40 0.33339450563397244 0.9425339237701337
          60 0.28177583778572934 1.0183762909810123
          80 0.2567930433560353 1.0773090493665145
          100 0.24200998648166638 1.1264200723210833
          120 0.23214592342192672 1.169348313872476
          140 0.22499789168750975 1.2076059866300377
          160 0.21952934068261357 1.2420983346064145
```

```
         180 0.21523669201130785 1.2741974684284156
        Final 0.2117072319154468 1.3026232621066003

==> 50,0,1.3026232621066003 == best (10,0.05,0.7379060900644814) ======
=======

F 50, penalty 0.01 :
         0 1.2487470565905419 0.8906009046939354
         20 0.5161424808616566 0.7958148140444338
         40 0.35400855077902776 0.8603653411856049
         60 0.29945364263405094 0.9076337384229787
         80 0.2734610311068477 0.9400055986687632
         100 0.25823442589644485 0.9647814222070012
         120 0.2482960255095816 0.983800725414814
         140 0.24116100070638463 0.9995507477534885
         160 0.23583994996019442 1.0124556549585635
         180 0.23169947035260802 1.02311998449387
        Final 0.22834508722507618 1.0321951397272437

==> 50,0.01,1.0321951397272437 == best (10,0.05,0.7379060900644814) ===
==========

F 50, penalty 0.05 :
         0 1.2487515253735615 0.8902664089633279
         20 0.6776609838974464 0.7689229070640714
         40 0.5319327102129497 0.7369722383993621
         60 0.4571680746637513 0.7377528179647267
         80 0.4202844954062021 0.7420776636811829
         100 0.40041397975754545 0.7457003462867329
         120 0.3883647054959105 0.7486941126374096
         140 0.380486933093903 0.7506675963841589
         160 0.3748313664105529 0.7522644383820764
         180 0.3707376412235852 0.753787983643904
        Final 0.36765453180563007 0.7547541747397597

==> 50,0.05,0.7547541747397597 == best (10,0.05,0.7379060900644814) ===
==========

F 50, penalty 0.1 :
         0 1.2487454857579188 0.8903457710468412
         20 0.7734409521899903 0.8059506010223434
         40 0.7257249028569269 0.7751769256709055
         60 0.693584132108956 0.7595056363774032
         80 0.67057552423068 0.751619199774373
         100 0.6536349739785063 0.7472690667129438
         120 0.6409795518731125 0.7447132321928942
         140 0.6313219667446035 0.7434629201230502
         160 0.6237979662144064 0.7426619812572853
         180 0.6177841089655354 0.7422925130552397
        Final 0.6129286470159068 0.7416743007117844

==> 50,0.1,0.7416743007117844 == best (10,0.05,0.7379060900644814) ====
=========

F 50, penalty 0.15 :
         0 1.248741209557507 0.8905974847301762
         20 0.8000690368009767 0.8232779784427541
```

```
          40 0.7863871949727568 0.8113496665622475
          60 0.7723234470682413 0.8005485471712596
          80 0.7653461108719484 0.7955834963036942
          100 0.7603584303286476 0.792502955906809
          120 0.7567567121071533 0.7904197238516715
          140 0.7540267108631035 0.7892134703842417
          160 0.7519111991254861 0.7882379975024959
          180 0.7499681741917394 0.787092917980116
         Final 0.7485627208718937 0.7861587972314186

==> 50,0.15,0.7861587972314186 == best (10,0.05,0.7379060900644814) ===
==========

F 50, penalty 0.2 :
          0 1.2487483565411743 0.8904301668608242
          20 0.8037802840688322 0.8247633795543221
          40 0.8038803316907026 0.8245145657286945
          60 0.8035196280760888 0.8246461174437322
          80 0.8021620967839985 0.8233790639910104
          100 0.7996102373067114 0.8211017058939286
          120 0.7962283709716358 0.8186175888774799
          140 0.793400033689117 0.8162568075650048
          160 0.7912582448214783 0.8150590973437074
          180 0.7900025278361523 0.8139589860740445
         Final 0.7891745523501638 0.8135250470328513

==> 50,0.2,0.8135250470328513 == best (10,0.05,0.7379060900644814) ====
=========

F 50, penalty 1 :
          0 1.2487491431389606 0.8905468333615276
          20 0.8057779616760875 0.8251076596962563
          40 0.8050773806833567 0.8249956807662538
          60 0.8047999868180554 0.8251015431642867
          80 0.804831702250246 0.8250710971729891
          100 0.8047822024878278 0.8252205548466781
          120 0.8047486571150537 0.8253907458837144
          140 0.8046226183830433 0.8251548207209047
          160 0.8046979703066276 0.825201782572365
          180 0.804568273485381 0.8251143575248835
         Final 0.8045683499161085 0.8253580472888702

==> 50,1,0.8253580472888702 == best (10,0.05,0.7379060900644814) ======
=======

F 100, penalty 0 :
          0 1.2487450614522968 0.8902476509991931
          20 0.3724669813148752 0.8467370664934354
          40 0.1797539260180194 1.0340974132293985
          60 0.12450746496878605 1.1594700137659857
          80 0.10012953165472965 1.2522807905321525
          100 0.08650312695879128 1.3272446104575024
          120 0.07770491971945709 1.3907075109533855
          140 0.07153748096481635 1.44638037809155
          160 0.066931851925765 1.4964246097115375
          180 0.06329813175922755 1.5420511665195873
         Final 0.060390747665515075 1.5821005456321011
```

```
==> 100,0,1.5821005456321011 == best (10,0.05,0.7379060900644814) =====
========

F 100, penalty 0.01 :
         0 1.2487482241055308 0.8904365925412794
         20 0.43152955731674686 0.784711105298026
         40 0.21622261310371252 0.8772954489434904
         60 0.1543893997963666 0.9358607201096283
         80 0.1280616540132085 0.9722551594712924
         100 0.1138245069702643 0.9975231591879773
         120 0.10489431552353777 1.0154248778036723
         140 0.09880415438816571 1.0292241231927635
         160 0.09433823970683937 1.0395513730799868
         180 0.09092255476158692 1.0475232781469013
        Final 0.08821537940745296 1.0537538407067202

==> 100,0.01,1.0537538407067202 == best (10,0.05,0.7379060900644814) ==
===========

F 100, penalty 0.05 :
         0 1.2487416344140596 0.89016477847669
         20 0.6617552900744972 0.7643882355714287
         40 0.4868316953528183 0.7324146131670651
         60 0.3917248191207323 0.7345031971503837
         80 0.34632324542372883 0.7390215080582597
         100 0.3228748077603335 0.7426650827863258
         120 0.3094623515246715 0.7447261497481676
         140 0.3011028475230539 0.7462496292999796
         160 0.2954329060749532 0.7475269275415795
         180 0.29148697806179 0.7482155977568432
        Final 0.28851400434824687 0.7485170561123343

==> 100,0.05,0.7485170561123343 == best (10,0.05,0.7379060900644814) ==
===========

F 100, penalty 0.1 :
         0 1.2487451132878467 0.8908498876187122
         20 0.7735723615501932 0.8069596508758177
         40 0.7229426157514011 0.7742283363629512
         60 0.6889385278959531 0.7579531283565806
         80 0.6653347333455427 0.750154824970553
         100 0.6478255481886275 0.7463200178245588
         120 0.6344398626120694 0.7440581326922153
         140 0.624162362635151 0.7427569921120406
         160 0.6161888151841776 0.7419621067632473
         180 0.6099654845022271 0.7416302373806152
        Final 0.6049121641193705 0.7414862346242342

==> 100,0.1,0.7414862346242342 == best (10,0.05,0.7379060900644814) ===
==========

F 100, penalty 0.15 :
         0 1.2487453004418512 0.8902362924741232
         20 0.7999077165168845 0.8234362319587158
         40 0.7860901691920958 0.8106922172518155
         60 0.7724835200204692 0.8005375785311646
```

```
          80 0.7660300580831789 0.7961832322470599
          100 0.7611486531900442 0.7930922766026198
          120 0.7572147383188237 0.7910037155985464
          140 0.7544915268796702 0.7894507798690841
          160 0.7522466497181732 0.7879520088396453
          180 0.7502985359149388 0.7871840840779913
         Final 0.7488113822888103 0.7864852797908417

==> 100,0.15,0.7864852797908417 == best (10,0.05,0.7379060900644814) ==
===========

F 100, penalty 0.2 :
          0 1.248744476359171 0.8902511742524595
          20 0.80377746039173 0.8247598168166586
          40 0.8038297485684246 0.8248956343767417
          60 0.8033950748352711 0.8244423577994429
          80 0.8019462206272017 0.8229908231279994
          100 0.7992425348375685 0.8210568552595597
          120 0.795864726287642 0.8182022758566609
          140 0.7929768401961903 0.8164308644662843
          160 0.7910793958882452 0.814562446548535
          180 0.7898336215553742 0.8140127044523744
         Final 0.789081105396459 0.8135460161056234

==> 100,0.2,0.8135460161056234 == best (10,0.05,0.7379060900644814) ===
==========

F 100, penalty 1 :
          0 1.2487488482066005 0.890432493551725
          20 0.8057836124259554 0.8250317923143471
          40 0.8051562734870956 0.8251117578124642
          60 0.8049297451253393 0.8253916776698335
          80 0.8047475708702674 0.824938753569981
          100 0.8046842349468073 0.8253359489928873
          120 0.8046618677272529 0.8249268272228665
          140 0.8046667739161485 0.8251029676865453
          160 0.804697397806051 0.8254414201352862
          180 0.8045807305722112 0.8253805316154442
         Final 0.804675144955977 0.8257464692121586

==> 100,1,0.8257464692121586 == best (10,0.05,0.7379060900644814) =====
========

F 150, penalty 0 :
          0 1.2487418456420882 0.890404817668943
          20 0.31431245672365826 0.8338266929972385
          40 0.10731261509425334 1.0292584315050657
          60 0.05775532255249758 1.1572462440851314
          80 0.038826078902308096 1.2463317268626681
          100 0.029250738624563993 1.3145585755944627
          120 0.023520433344176327 1.3698666162197948
          140 0.019718009489064013 1.4164679961371724
          160 0.016999338521971885 1.4568345422462323
          180 0.014965324802996076 1.492610981361956
         Final 0.013378662668130554 1.5229994247972756

==> 150,0,1.5229994247972756 == best (10,0.05,0.7379060900644814) =====
```

```
========

F 150, penalty 0.01 :
         0 1.2487461761349778 0.8900564068638688
         20 0.3860410223572646 0.7782741018755918
         40 0.14967157716578205 0.8673103935815706
         60 0.091271393459932 0.9167478703772921
         80 0.06949033350702684 0.9422422097015166
         100 0.05870157802642984 0.9569047704179858
         120 0.052411865391086096 0.9660562724857407
         140 0.04831354023175787 0.9717310777058363
         160 0.04542913971633942 0.975591640228024
         180 0.04329618114508012 0.9781136274086256
        Final 0.04163869743834605 0.979739304115059

==> 150,0.01,0.979739304115059 == best (10,0.05,0.7379060900644814) ===
==========

F 150, penalty 0.05 :
         0 1.2487446623386098 0.8903965657260069
         20 0.6543243352508199 0.7611843137778104
         40 0.4658153676435412 0.7269435502549191
         60 0.36410840893352364 0.7281735964758181
         80 0.31708628797552263 0.7317956391628535
         100 0.29377608052826654 0.7346158546462738
         120 0.2807044093595937 0.7361700035366813
         140 0.2726919133697445 0.737315788535038
         160 0.2674292551108461 0.7379593934569468
         180 0.263777884854715 0.7382560633300417
        Final 0.26108131038661575 0.7386567764870601

==> 150,0.05,0.7386567764870601 == best (10,0.05,0.7379060900644814) ==
==========

F 150, penalty 0.1 :
         0 1.2487436782134127 0.8901431529916848
         20 0.7704289820142968 0.8041790384678712
         40 0.7213999621207607 0.7729562062751685
         60 0.6875527414471625 0.7571955565006927
         80 0.6636626248391732 0.749591627264993
         100 0.6458287396303315 0.7456066527215758
         120 0.6324759784024767 0.7435188888453946
         140 0.6222054748312242 0.7422724577432904
         160 0.6141053180675078 0.7416298449976029
         180 0.6077285731608622 0.7414986852631797
        Final 0.6027854721649729 0.7411485993007819

==> 150,0.1,0.7411485993007819 == best (10,0.05,0.7379060900644814) ===
==========

F 150, penalty 0.15 :
         0 1.2487430018170327 0.8906223877106585
         20 0.7997383288677898 0.8229907850171079
         40 0.7853281850636283 0.8105270728479007
         60 0.7719461818562541 0.8002406553725515
         80 0.7654255827915234 0.7957295688510567
         100 0.7605406994841589 0.7929565275303634
```

```
          120 0.7567263434295372 0.7904400925240117
          140 0.7540229370669359 0.7893353140285004
          160 0.7518003341396087 0.788068742449398
          180 0.7499127590229777 0.787111536426856
         Final 0.7484085762986435 0.7865825936272818

==> 150,0.15,0.7865825936272818 == best (10,0.05,0.7379060900644814) ==
===========

F 150, penalty 0.2 :
          0 1.248737584238891 0.8905154631542105
          20 0.8038285981615703 0.8249001762435747
          40 0.8037828200019338 0.8246816181209048
          60 0.8033178174626531 0.8241640172384199
          80 0.8018235688590665 0.8230998748411817
          100 0.799234820298462 0.8211914518706656
          120 0.795978960747687 0.8186637418649656
          140 0.7930039459324351 0.8165751048848021
          160 0.7911006455609614 0.8148021789944258
          180 0.7898635474680206 0.8138280795251869
         Final 0.7891392480175354 0.8136717662214284

==> 150,0.2,0.8136717662214284 == best (10,0.05,0.7379060900644814) ===
==========

F 150, penalty 1 :
          0 1.2487467441814784 0.8902256596320252
          20 0.8057661702173866 0.8250666957307523
          40 0.8051231408667391 0.8250516074862483
          60 0.8048897983343494 0.8251749078013332
          80 0.8047775786558076 0.8252300750308507
          100 0.8046998129304759 0.8255357574926846
          120 0.8047072567141516 0.825115389110151
          140 0.8047585980706925 0.8250804204088198
          160 0.8046510318588976 0.825084672943832
          180 0.8046132828740077 0.8254939139267921
         Final 0.8046023105023121 0.8252545455694128

==> 150,1,0.8252545455694128 == best (10,0.05,0.7379060900644814) =====
========
```

In [35]:
```
fits=pd.DataFrame(results,columns=["F","penalty","val_rms"])
fits.head()
```

Out[35]:

	F	penalty	val_rms
0	1	0.00	0.782696
1	1	0.01	0.782128
2	1	0.05	0.781258
3	1	0.10	0.787065
4	1	0.15	0.798355

In [36]:
```
summary=pd.pivot_table(fits,index="F",columns="penalty",values="val_rms"
)
summary
```

Out[36]:

penalty	0.0	0.01	0.05	0.1	0.15	0.2	1.0
F							
1	0.782696	0.782128	0.781258	0.787065	0.798355	0.815372	0.825193
5	0.764203	0.758633	0.741395	0.754068	0.788218	0.813963	0.825245
10	0.813056	0.787762	0.737906	0.747374	0.786918	0.813867	0.825278
20	0.952808	0.870673	0.749341	0.744736	0.787138	0.813556	0.825290
30	1.083229	0.935764	0.754461	0.742779	0.786446	0.813682	0.825612
50	1.302623	1.032195	0.754754	0.741674	0.786159	0.813525	0.825358
100	1.582101	1.053754	0.748517	0.741486	0.786485	0.813546	0.825746
150	1.522999	0.979739	0.738657	0.741149	0.786583	0.813672	0.825255

In [37]:
```
best=fits.iloc[fits["val_rms"].idxmin()]
best
```

Out[37]:
```
F           10.000000
penalty      0.050000
val_rms      0.737906
Name: 16, dtype: float64
```

In [38]:
```
import seaborn as sns

cm = sns.light_palette("#60FF60", reverse=True, as_cmap=True)

s = summary.style.highlight_min(axis=None)
s
```

Out[38]:

penalty	0.0	0.01	0.05	0.1	0.15	0.2	1.0
F							
1	0.782696	0.782128	0.781258	0.787065	0.798355	0.815372	0.825193
5	0.764203	0.758633	0.741395	0.754068	0.788218	0.813963	0.825245
10	0.813056	0.787762	0.737906	0.747374	0.786918	0.813867	0.825278
20	0.952808	0.870673	0.749341	0.744736	0.787138	0.813556	0.82529
30	1.08323	0.935764	0.754461	0.742779	0.786446	0.813682	0.825612
50	1.30262	1.0322	0.754754	0.741674	0.786159	0.813525	0.825358
100	1.5821	1.05375	0.748517	0.741486	0.786485	0.813546	0.825746
150	1.523	0.979739	0.738657	0.741149	0.786583	0.813672	0.825255

In [39]:
```
F_best=int(best["F"])
penalty_best=best["penalty"]
F_best,penalty_best
```

Out[39]: (10, 0.05)

Cross Validation

It seems $F \approx 10$ with penalty around 0.05 gives best results.

We use 5-Fold cross validation to find the optimal value of F

In [40]:
```
K=5
kfold=KFold(K,shuffle=True)
folds=[]
for fold in kfold.split(users):
    folds.append(fold)
```

In [41]:
```
def ratings_cross_validate(model,users,movies,ratings,folds):
    accuracies=[]
    count=0
    for train,val in folds:
        print()
        print("============= Fold",count+1,"===========")
        users_train=users[train]
        movies_train=movies[train]
        ratings_train=ratings[train]
        users_val=users[val]
        movies_val=movies[val]
        ratings_val=ratings[val]
        loss=model.fit(users_train,movies_train,ratings_train,
                                    users_val,movies_val,ratings_val)

        accuracies.append(loss)
        print("======= fold",count+1,"loss",loss,"============")
        print()
        count+=1
    accuracies=np.array(accuracies)
    return accuracies.mean()
```

In [58]:
```
results=[]
best_loss=1e10
best_F=None
steps=100
if True:
  for F in [5,10,15]:
        print()
        print(f"F {F} :")
        model=Recommender(F,best_penalty,learning_rate,steps,batch_size)
        loss=ratings_cross_validate(model,users,movies,ratings,
                                    folds)
        results.append((F,loss))
        if loss<best_loss:
            best_loss=loss
            best_F=F
        print()
        print(f"==> {F},{penalty},{loss} == best ({best_F},{best_penalt
y},{best_loss}) ==============")
```

```
F 5 :

============= Fold 1 ===========
         0 1.2485533530336144 0.8932260820557525
         10 0.792153645151391 0.8284441461561335
         20 0.7596851350083867 0.8073201016757684
         30 0.7291479115139694 0.7905257533022896
         40 0.7057719194647919 0.7788425942450067
         50 0.6892993059143258 0.7706883615736058
         60 0.6777306704150264 0.7652558488252055
         70 0.6692703522347676 0.7611975269798796
         80 0.6627616923298618 0.7585839984881941
         90 0.6580722714403028 0.7560365320775337
        Final 0.6544426100756603 0.754407832790789
======= fold 1 loss 0.754407832790789 ============


============= Fold 2 ===========
         0 1.2478524832474827 0.8962957102144979
         10 0.7922599406899052 0.8333267782960897
         20 0.7670963705646401 0.8174016801086219
         30 0.7343180194957236 0.7978551599139871
         40 0.7100195764880174 0.7845717707513513
         50 0.6914439061011414 0.7748161832518049
         60 0.6778810060704996 0.7676659654676805
         70 0.6686360374904579 0.763427531750924
         80 0.6619130533879918 0.760341507843067
         90 0.6572000510507808 0.758578890886326
        Final 0.6536456243057528 0.757479807183501
======= fold 2 loss 0.757479807183501 ============


============= Fold 3 ===========
         0 1.2492642792078534 0.8947221627768374
         10 0.7939660288211087 0.8291699805414388
         20 0.7698291012192804 0.8140464893229006
         30 0.7347453025168861 0.7937635771912516
         40 0.7106236713523204 0.7813324798833032
         50 0.6920773171208617 0.7719964589021929
         60 0.6783148439958833 0.7642082866316559
         70 0.6685869790970365 0.7587455389347139
         80 0.6615956323960517 0.7553360382804399
         90 0.6566176963256017 0.7525007769328004
        Final 0.6527355226218186 0.7508569211509154
======= fold 3 loss 0.7508569211509154 ============


============= Fold 4 ===========
         0 1.2489324870318703 0.8903585678047412
         10 0.7933966262071291 0.8287365536436028
         20 0.766461055292232 0.8112898690569719
         30 0.7309153312565705 0.7902043186630511
         40 0.7045407033718348 0.7760313864894874
         50 0.6858201256416288 0.7659844730188429
         60 0.6732484689740458 0.7599294383376045
         70 0.6649835345409911 0.7562610060662138
         80 0.6591863975633268 0.7535902453687865
```

```
         90 0.6550099862457914 0.7521867572408737
        Final 0.6519220329742245 0.7515161240344423
======= fold 4 loss 0.7515161240344423 ============


============= Fold 5 ===========
         0 1.2490152427609635 0.893431505158139
         10 0.7911625996998342 0.8283252611915974
         20 0.757712093784779 0.8059846045260528
         30 0.725557802144149 0.7883267681958123
         40 0.703368137343983 0.7778191151003601
         50 0.6877735046280563 0.7705311388024559
         60 0.6768508296167418 0.7661292627484646
         70 0.6692222443367375 0.7629228794068688
         80 0.663647100747699 0.7607534575750667
         90 0.6593872115520141 0.7592617367194755
        Final 0.6561867683883623 0.7583666675663374
======= fold 5 loss 0.7583666675663374 ============


==> 5,1,0.754525470545197 == best (5,0.05,0.754525470545197) ==========
===

F 10 :

============= Fold 1 ===========
         0 1.2485507126207753 0.8926275435426806
         10 0.7862761820062014 0.8274531230444889
         20 0.7421554213612975 0.8011269975382794
         30 0.6926648920182926 0.7775103895572776
         40 0.658036635543289 0.7641476629029821
         50 0.6349463185082386 0.7570958604814425
         60 0.6190352916259858 0.7529427190356769
         70 0.608055937810148 0.7507933399951927
         80 0.5999341315855167 0.7489396281767169
         90 0.59379651970139 0.7479761752063447
        Final 0.5890279745653612 0.7475656630691063
======= fold 1 loss 0.7475656630691063 ============


============= Fold 2 ===========
         0 1.247859104552973 0.8957204059612365
         10 0.7835047094072967 0.8300078580805131
         20 0.7368639330843747 0.8022487223291547
         30 0.6913535941334927 0.7809578633283307
         40 0.6585717706643361 0.7693227383477997
         50 0.6359333672085833 0.7626983160485273
         60 0.620332734651667 0.7591373276489335
         70 0.6090306037263348 0.7569080471475368
         80 0.600838796425524 0.7549008998532963
         90 0.594704880180038 0.7538008933434049
        Final 0.5900114533459271 0.7534480009554136
======= fold 2 loss 0.7534480009554136 ============


============= Fold 3 ===========
         0 1.2492654888087298 0.8938737991331048
```

```
         10 0.7852745813360785 0.8267488383141236
         20 0.7382178205568025 0.7984797417672413
         30 0.6909847923764343 0.7755382680654787
         40 0.6591106994161106 0.7636763214892184
         50 0.6374818819110074 0.7575570871776217
         60 0.6224177350065527 0.7539450368942835
         70 0.6115390408582173 0.7519095881552704
         80 0.6035215644256942 0.7505754032023687
         90 0.5973769296684757 0.7496890075093582
        Final 0.5923796101451001 0.7491827172114203
======= fold 3 loss 0.7491827172114203 ============


============= Fold 4 ===========
         0 1.248936544942687 0.8903058664698152
         10 0.7850506541023992 0.8262847549798703
         20 0.7390474615951647 0.7993596841222609
         30 0.6932186936680107 0.7777196918042024
         40 0.6604324942233092 0.765194332541127
         50 0.6381717546979699 0.7580062311169431
         60 0.6227216234316897 0.7537959638808839
         70 0.6115794888948204 0.7510828998414056
         80 0.6031157790500361 0.748859279382836
         90 0.5967498465396862 0.7476650481571516
        Final 0.5917872397331728 0.7466662162395968
======= fold 4 loss 0.7466662162395968 ============


============= Fold 5 ===========
         0 1.2490140597280115 0.8931897383396855
         10 0.7837105675188053 0.8262348291113716
         20 0.7358247698063187 0.7975254047240382
         30 0.6911313857006126 0.7758901501841615
         40 0.6585728774282388 0.7643239392454375
         50 0.6366861615666882 0.7583383206830782
         60 0.621596387247434 0.7550149550998296
         70 0.6108668027072055 0.7529202861829065
         80 0.6029439490513187 0.751850403982129
         90 0.5969434520571604 0.7508717602564913
        Final 0.5922324822973674 0.7508126736507373
======= fold 5 loss 0.7508126736507373 ============


==> 10,1,0.7495350542252549 == best (10,0.05,0.7495350542252549) ======
=======

F 15 :

============= Fold 1 ===========
         0 1.248555026495998 0.8925883572949387
         10 0.7815127487605379 0.8267788375855819
         20 0.7257008355739724 0.7963911772377493
         30 0.6672511432483502 0.7719355035015518
         40 0.6267449000803704 0.7605881901867618
         50 0.5997348506521214 0.7554991022737152
         60 0.5813603039725534 0.7530228253597514
         70 0.5684495635173628 0.7520492121647997
```

```
         80 0.5587691789585074 0.751436515578365
         90 0.5515788259218983 0.7512336588572562
        Final 0.5459469217516615 0.75120698218398
======= fold 1 loss 0.75120698218398 ============


============= Fold 2 ===========
         0 1.247859609387365 0.8950559853872531
         10 0.7789776413661729 0.8276392457243463
         20 0.7211774614177043 0.7953988421237478
         30 0.6676559502112225 0.7737830441714737
         40 0.6281137586884737 0.7630338601117512
         50 0.6006656381836355 0.7577160515948923
         60 0.5819282025247141 0.75551378961243
         70 0.5688281145154873 0.7543128485519889
         80 0.5592884342776663 0.7539446696917186
         90 0.55196446987085 0.7537506121110705
        Final 0.5464681524216699 0.75371952727639
======= fold 2 loss 0.75371952727639 ============


============= Fold 3 ===========
         0 1.2492635675632229 0.8936881888701315
         10 0.7799398593944412 0.8241693705873789
         20 0.7235722290124241 0.7933482353733734
         30 0.6697933053902676 0.7723410037259685
         40 0.6287872871407577 0.7608228385845908
         50 0.6006014226986377 0.7551594895009263
         60 0.5817025859540833 0.7529412431955143
         70 0.5685972994337738 0.7516858449730692
         80 0.5590187559244967 0.7512947635318404
         90 0.55202091509235 0.7514702451712061
        Final 0.5465233171792638 0.7515600060848673
======= fold 3 loss 0.7515600060848673 ============


============= Fold 4 ===========
         0 1.2489322646078431 0.8901139973061846
         10 0.7814179691981705 0.8265183882253605
         20 0.7292955958754312 0.7995735983292912
         30 0.6725178283082797 0.775798262365037
         40 0.630519258323056 0.7620555451278721
         50 0.6024541019923279 0.7554372593275449
         60 0.5836898564309042 0.7521441242718241
         70 0.5704069960267236 0.7508526278855064
         80 0.5607084226759682 0.7500783305367481
         90 0.553513426035628 0.7498696587409166
        Final 0.5478169683213584 0.749699801930754
======= fold 4 loss 0.749699801930754 ============


============= Fold 5 ===========
         0 1.2490041487546575 0.8929051175356411
         10 0.7782381440270786 0.8254904603460482
         20 0.7184302974484632 0.7937872502051138
         30 0.6662040790897702 0.7733782360750112
         40 0.6277971807705488 0.763338295848253
```

```
          50 0.6009716181301853 0.7580625474067376
          60 0.582379256415953 0.7558669642638263
          70 0.5692704228510136 0.7543656458231661
          80 0.5596501254639026 0.7537558141946534
          90 0.5524028808630612 0.7537394345606137
         Final 0.5466941424695451 0.7537372173462676
======= fold 5 loss 0.7537372173462676 ============


==> 15,1,0.7519847069644519 == best (10,0.05,0.7495350542252549) ======
=======
```

Test Best model

Seems best model is really $F = 10$ with penalty 0.02, so we test performance on the test set

In [59]:
```
model=Recommender(best_F,penalty_best,learning_rate,steps,batch_size)
loss=model.fit(users,movies,ratings,users_test,movies_test,ratings_test)
print(best_F,penalty_best,loss)
```

```
          0 1.2487151147154587 0.8734996712312967
          10 0.7854459353697039 0.8168954476243606
          20 0.7298827082827767 0.7789219089035535
          30 0.6857647008788721 0.7546617468115456
          40 0.6565707493132016 0.7414681641613389
          50 0.6377613938778485 0.7344088473281033
          60 0.6254117611629383 0.7306500346148762
          70 0.6170301391922364 0.7286495844998203
          80 0.611048728955314 0.7274774636837298
          90 0.606653455554716 0.7263877950958051
         Final 0.6032863549775624 0.7261302778562422
10 0.05 0.7261302778562422
```

In [60]:
```
loss
```

Out[60]:
```
0.7261302778562422
```

We have achieved a ≈ 0.73 mean square error, a 12% improvement in accuracy over the 0.83 mean square error of the popularity model.

In [ ]:

In [ ]: