Classification Project

This dataset is about past loans. The *Loan$_t$rain.csv* data set includes details of 346 customers whose loan are already paid off or defaulted.

```
import itertools
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import seaborn as sns
from matplotlib import rc,font_manager
from sklearn import preprocessing
ticks_font = font_manager.FontProperties(family='Times New Roman', style='normal',
    size=12, weight='normal', stretch='normal')
ax=plt.gca()

#Loading Data and Data Preprocessing:
df=pd.read_csv('D:\Python\edx\Machine Learning\Classification example\loan_train.csv')
with open('class_problem.txt','a') as f:
    print(df.head(),file=f)
    print(df.shape,file=f)

    ## Convert to date time object

df['due_date']=pd.to_datetime(df['due_date'])
df['effective_date']=pd.to_datetime(df['effective_date'])
with open('class_problem.txt','a') as f:
    print(df.head(),file=f)

# Data visulaization (using seaborn) and Preprocessing

status=df['loan_status'].value_counts()
with open('class_problem.txt','a') as f:
    print(status,file=f)

bins=np.linspace(df.Principal.min(),df.Principal.max(),10)
g=sns.FacetGrid(df,col='Gender',hue='loan_status',palette='Set1',col_wrap=2)
g.map(plt.hist,'Principal',bins=bins,ec='k')
g.axes[-1].legend()

bins1=np.linspace(df.age.min(),df.age.max(),10)
g1=sns.FacetGrid(df,col='Gender',hue='loan_status',palette='Set1',col_wrap=2)
g1.map(plt.hist,'age',bins=bins1,ec='k')
```

```
g1.axes[-1].legend()

## Feature Selection and Extraction
# Creating Weekday and weekend columns#

df['dayofweek']=df['effective_date'].dt.dayofweek
bins2=np.linspace(df.dayofweek.min(),df.dayofweek.max(),10)
g2=sns.FacetGrid(df,col='Gender',hue='loan_status',palette='Set1',col_wrap=2)
g2.map(plt.hist,'dayofweek',bins=bins2,ec='k')
g2.axes[-1].legend()
sns.set_style("darkgrid",{'font.sans-serif': ['Arial']})

df['weekend']=df['dayofweek'].apply(lambda x: 1 if (x>3) else 0)
with open('class_problem.txt','a') as f:
    print(df.head(),file=f)

# Converting categorical features to numerical values
### Gender ##
group1=df.groupby(['Gender'])['loan_status'].value_counts(normalize=True)
with open('class_problem.txt','a') as f:
    print(group1,file=f)

#Converting male=0 and female=1

df['Gender'].replace(to_replace=['male','female'],value=[0,1],inplace=True)
with open('class_problem.txt','a') as f:
    print(df.head(),file=f)

### Education ##
group2=df.groupby(['education'])['loan_status'].value_counts(normalize=True)
with open('class_problem.txt','a') as f:
    print(group2,file=f)
#Using one hot encoding technique to conver categorical varables to binary variables and


Feature = df[['Principal','terms','age','Gender','weekend']]
Feature = pd.concat([Feature,pd.get_dummies(df['education'])], axis=1)
Feature.drop(['Master or Above'], axis = 1,inplace=True)
with open('class_problem.txt','a') as f:
    print(Feature.head(),file=f)

#Mentioning X,y
X=Feature
y=df['loan_status'].values
with open('class_problem.txt','a') as f:
    print(X[0:5],file=f)
```

```
    print(y[0:5],file=f)

#Normalizing Data
X = preprocessing.StandardScaler().fit(X).transform(X)
with open('class_problem.txt','a') as f:
    print(X[0:5],file=f)

#####  Classification - KNN, Decision Tree, SVM,LR #####

#Split Data Train/Test
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.2,random_state=4)
with open('class_problem.txt','a') as f:
    print('Train set: ',X_train.shape,y_train.shape,file=f)
    print('Test set: ', X_test.shape,y_test.shape,file=f)

#Modeling
#KNN- first we will find best value of k and then will train the data

from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
Ks=15
acc_mean=np.zeros((Ks-1))
acc_std=np.zeros((Ks-1))
Cmat=[]
for n in range(1,Ks):
    neigh=KNeighborsClassifier(n_neighbors=n).fit(X_train,y_train)
    yhat_KNN=neigh.predict(X_test)
    acc_mean[n-1]=metrics.accuracy_score(y_test,yhat_KNN)

    acc_std[n-1]=np.std(yhat_KNN==y_test)/np.sqrt(yhat_KNN.shape[0])
with open('class_problem.txt','a') as f:
    print(acc_mean,file=f)
    print('The best accuracy was with: ', acc_mean.max(), 'with k= ',acc_mean.argmax()+1

#Plot of k values vs accuracy #
plt.figure()
plt.plot(range(1,Ks),acc_mean,'g')
plt.fill_between(range(1,Ks),acc_mean - 1 * acc_std,acc_mean + 1 * acc_std, alpha=0.10)
plt.legend(('Accuracy ', '+/- 3xstd'))
plt.ylabel('Accuracy ',fontname='Times New Roman',fontsize=12)
plt.xlabel('Number of Nabors (K)',fontname='Times New Roman',fontsize=12)
plt.tight_layout()


k=7 #Best value found above
```

```python
kNN_model = KNeighborsClassifier(n_neighbors=k).fit(X_train,y_train)

#Decision Tree
from sklearn.tree import DecisionTreeClassifier
DT_model = DecisionTreeClassifier(criterion="entropy", max_depth = 4)
DT_model.fit(X_train,y_train)

#SVM
from sklearn import svm
SVM_model = svm.SVC()
SVM_model.fit(X_train, y_train)

#Logitic Regression
from sklearn.linear_model import LogisticRegression
LR_model = LogisticRegression(C=0.01).fit(X_train,y_train)

############ Test Evaluation Data #############

test_df = pd.read_csv('D:\Python\edx\Machine Learning\Classification example\loan_test.c
with open('class_problem.txt','a') as f:
    print('Test Data: \n',test_df.head(),file=f)

#Preprocessing for test data as same as train data
test_df['due_date'] = pd.to_datetime(test_df['due_date'])
test_df['effective_date'] = pd.to_datetime(test_df['effective_date'])
test_df['dayofweek'] = test_df['effective_date'].dt.dayofweek
test_df['weekend'] = test_df['dayofweek'].apply(lambda x: 1 if (x>3)  else 0)
test_df['Gender'].replace(to_replace=['male','female'], value=[0,1],inplace=True)
test_Feature = test_df[['Principal','terms','age','Gender','weekend']]
test_Feature = pd.concat([test_Feature,pd.get_dummies(test_df['education'])], axis=1)
test_Feature.drop(['Master or Above'], axis = 1,inplace=True)
test_X = preprocessing.StandardScaler().fit(test_Feature).transform(test_Feature)
test_y = test_df['loan_status'].values
with open('class_problem.txt','a') as f:
    print(test_X[0:5],file=f)
    print(test_y[0:5],file=f)

#Evaluating accuracy #
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import log_loss

knn_yhat = kNN_model.predict(test_X)
DT_yhat = DT_model.predict(test_X)
SVM_yhat = SVM_model.predict(test_X)
LR_yhat = LR_model.predict(test_X)
```

```
LR_yhat_prob = LR_model.predict_proba(test_X)

with open('class_problem.txt','a') as f:
    print("KNN Jaccard index: ",  accuracy_score(test_y, knn_yhat),file=f)
    print("KNN F1-score: ",  f1_score(test_y, knn_yhat, average='weighted') ,file=f)
    print("DT Jaccard index: ",  accuracy_score(test_y, DT_yhat),file=f)
    print("DT F1-score: ", f1_score(test_y, DT_yhat, average='weighted') ,file=f)
    print("SVM Jaccard index: ", accuracy_score(test_y, SVM_yhat),file=f)
    print("SVM F1-score: ",f1_score(test_y, SVM_yhat, average='weighted') ,file=f)
    print("LR Jaccard index: f", accuracy_score(test_y, LR_yhat),file=f)
    print("LR F1-score: %.2f", f1_score(test_y, LR_yhat, average='weighted'),file=f )
    print("LR LogLoss: %.2f ", log_loss(test_y, LR_yhat_prob),file=f )

#Display plot
plt.show()
```

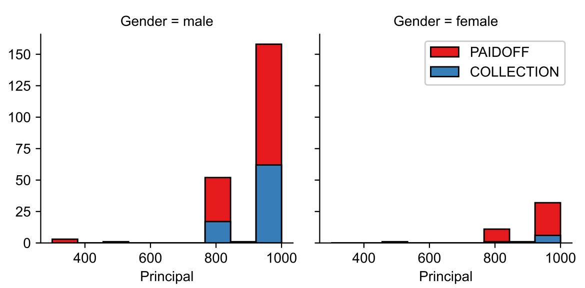
Solution:

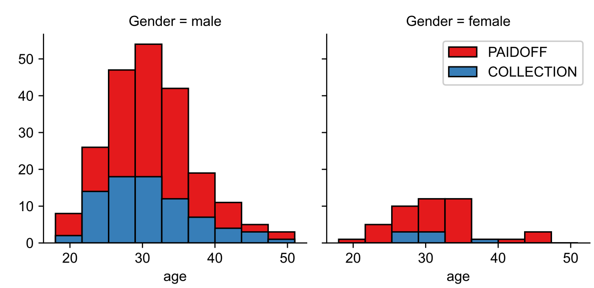
```
   Unnamed: 0  Unnamed: 0.1 loan_status  Principal  terms effective_date   due_date
age             education  Gender
0           0             0     PAIDOFF       1000     30       9/8/2016  10/7/2016
45  High School or Below    male
1           2             2     PAIDOFF       1000     30       9/8/2016  10/7/2016
33              Bechalor  female
2           3             3     PAIDOFF       1000     15       9/8/2016  9/22/2016
27               college    male
3           4             4     PAIDOFF       1000     30       9/9/2016  10/8/2016
28               college  female
4           6             6     PAIDOFF       1000     30       9/9/2016  10/8/2016
29               college    male
(346, 10)
   Unnamed: 0  Unnamed: 0.1 loan_status  Principal  terms effective_date   due_date
age             education  Gender
0           0             0     PAIDOFF       1000     30     2016-09-08 2016-10-07
45  High School or Below    male
1           2             2     PAIDOFF       1000     30     2016-09-08 2016-10-07
33              Bechalor  female
2           3             3     PAIDOFF       1000     15     2016-09-08 2016-09-22
27               college    male
3           4             4     PAIDOFF       1000     30     2016-09-09 2016-10-08
28               college  female
4           6             6     PAIDOFF       1000     30     2016-09-09 2016-10-08
29               college    male
PAIDOFF       260
COLLECTION     86
```

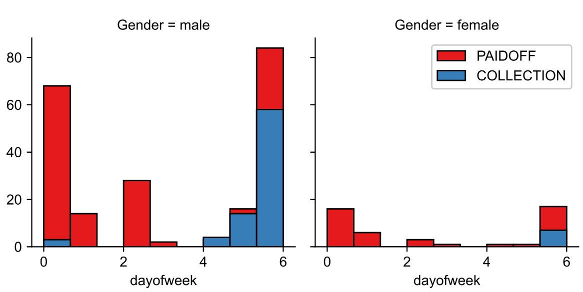
```
Name: loan_status, dtype: int64
   Unnamed: 0  Unnamed: 0.1 loan_status  Principal  terms effective_date   due_date
age             education  Gender  dayofweek  weekend
0           0             0     PAIDOFF       1000     30     2016-09-08 2016-10-07
45  High School or Below     male           3        0
1           2             2     PAIDOFF       1000     30     2016-09-08 2016-10-07
33               Bechalor  female           3        0
2           3             3     PAIDOFF       1000     15     2016-09-08 2016-09-22
27                college    male           3        0
3           4             4     PAIDOFF       1000     30     2016-09-09 2016-10-08
28                college  female           4        1
4           6             6     PAIDOFF       1000     30     2016-09-09 2016-10-08
29                college    male           4        1
Gender  loan_status
female  PAIDOFF        0.865385
        COLLECTION     0.134615
male    PAIDOFF        0.731293
        COLLECTION     0.268707
Name: loan_status, dtype: float64
   Unnamed: 0  Unnamed: 0.1 loan_status  Principal  terms effective_date   due_date
age             education  Gender  dayofweek  weekend
0           0             0     PAIDOFF       1000     30     2016-09-08 2016-10-07
45  High School or Below       0           3        0
1           2             2     PAIDOFF       1000     30     2016-09-08 2016-10-07
33               Bechalor       1           3        0
2           3             3     PAIDOFF       1000     15     2016-09-08 2016-09-22
27                college       0           3        0
3           4             4     PAIDOFF       1000     30     2016-09-09 2016-10-08
28                college       1           4        1
4           6             6     PAIDOFF       1000     30     2016-09-09 2016-10-08
29                college       0           4        1
education             loan_status
Bechalor              PAIDOFF        0.750000
                      COLLECTION     0.250000
High School or Below  PAIDOFF        0.741722
                      COLLECTION     0.258278
Master or Above       COLLECTION     0.500000
                      PAIDOFF        0.500000
college               PAIDOFF        0.765101
                      COLLECTION     0.234899
Name: loan_status, dtype: float64
   Principal  terms  age  Gender  weekend  Bechalor  High School or Below  college
0       1000     30   45       0        0         0                     1
0
1       1000     30   33       1        0         1                     0
0
```

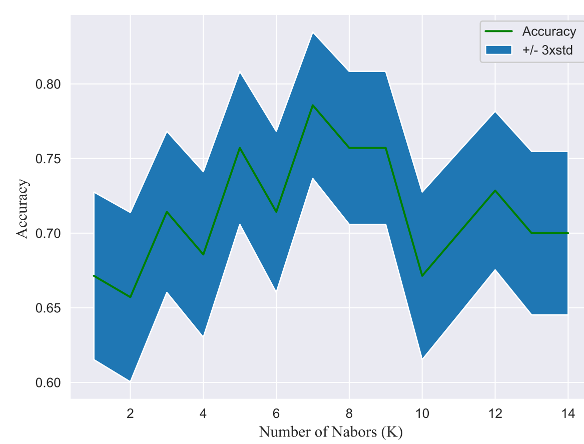
```
2       1000      15    27        0         0          0                      0
1
3       1000      30    28        1         1          0                      0
1
4       1000      30    29        0         1          0                      0
1
   Principal  terms   age  Gender   weekend  Bechalor  High School or Below  college
0       1000      30    45        0         0          0                      1
0
1       1000      30    33        1         0          1                      0
0
2       1000      15    27        0         0          0                      0
1
3       1000      30    28        1         1          0                      0
1
4       1000      30    29        0         1          0                      0
1
['PAIDOFF' 'PAIDOFF' 'PAIDOFF' 'PAIDOFF' 'PAIDOFF']
[[ 0.51578458  0.92071769  2.33152555 -0.42056004 -1.20577805 -0.38170062
   1.13639374 -0.86968108]
 [ 0.51578458  0.92071769  0.34170148  2.37778177 -1.20577805  2.61985426
  -0.87997669 -0.86968108]
 [ 0.51578458 -0.95911111 -0.65321055 -0.42056004 -1.20577805 -0.38170062
  -0.87997669  1.14984679]
 [ 0.51578458  0.92071769 -0.48739188  2.37778177  0.82934003 -0.38170062
  -0.87997669  1.14984679]
 [ 0.51578458  0.92071769 -0.3215732  -0.42056004  0.82934003 -0.38170062
  -0.87997669  1.14984679]]
Train set:  (276, 8) (276,)
Test set:  (70, 8) (70,)
[0.67142857 0.65714286 0.71428571 0.68571429 0.75714286 0.71428571
 0.78571429 0.75714286 0.75714286 0.67142857 0.7        0.72857143
 0.7        0.7       ]
The best accuracy was with:  0.7857142857142857 with k=  7
Test Data:
    Unnamed: 0  Unnamed: 0.1 loan_status  Principal  terms effective_date   due_date
age              education  Gender
0           1             1     PAIDOFF       1000     30       9/8/2016  10/7/2016
50               Bechalor  female
1           5             5     PAIDOFF        300      7       9/9/2016  9/15/2016
35        Master or Above    male
2          21            21     PAIDOFF       1000     30      9/10/2016  10/9/2016
43   High School or Below  female
3          24            24     PAIDOFF       1000     30      9/10/2016  10/9/2016
26                college    male
4          35            35     PAIDOFF        800     15      9/11/2016  9/25/2016
```

```
29               Bechalor    male
[[ 0.49362588  0.92844966  3.05981865  1.97714211 -1.30384048  2.39791576
  -0.79772404 -0.86135677]
 [-3.56269116 -1.70427745  0.53336288 -0.50578054  0.76696499 -0.41702883
  -0.79772404 -0.86135677]
 [ 0.49362588  0.92844966  1.88080596  1.97714211  0.76696499 -0.41702883
   1.25356634 -0.86135677]
 [ 0.49362588  0.92844966 -0.98251057 -0.50578054  0.76696499 -0.41702883
  -0.79772404  1.16095912]
 [-0.66532184 -0.78854628 -0.47721942 -0.50578054  0.76696499  2.39791576
  -0.79772404 -0.86135677]]
['PAIDOFF' 'PAIDOFF' 'PAIDOFF' 'PAIDOFF' 'PAIDOFF']
KNN Jaccard index:  0.6666666666666666
KNN F1-score:  0.6328400281888654
DT Jaccard index:  0.7222222222222222
DT F1-score:  0.7366818873668188
SVM Jaccard index:  0.7962962962962963
SVM F1-score:  0.7583503077293734
LR Jaccard index: f 0.7407407407407407
LR F1-score: %.2f 0.6304176516942475
LR LogLoss: %.2f  0.5163663771215675
```


Gender = male
Gender = female
PAIDOFF
COLLECTION
Principal
Principal


Gender = male
Gender = female
PAIDOFF
COLLECTION
age
age


Gender = male
Gender = female
PAIDOFF
COLLECTION
dayofweek
dayofweek


Accuracy
+/- 3xstd
Accuracy
Number of Nabors (K)