In [1]:

```
import itertools
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import NullFormatter
import pandas as pd
import numpy as np
import matplotlib.ticker as ticker
from sklearn import preprocessing
%matplotlib inline
```

In [2]:

```
!wget -O loan_train.csv https://s3-api.us-geo.objectstorage.softlayer.net/cf-cou
rses-data/CognitiveClass/ML0101ENv3/labs/loan_train.csv
```

```
--2020-02-23 09:45:54--  https://s3-api.us-geo.objectstorage.softlay
er.net/cf-courses-data/CognitiveClass/ML0101ENv3/labs/loan_train.csv
Resolving s3-api.us-geo.objectstorage.softlayer.net (s3-api.us-geo.o
bjectstorage.softlayer.net)... 67.228.254.196
Connecting to s3-api.us-geo.objectstorage.softlayer.net (s3-api.us-g
eo.objectstorage.softlayer.net)|67.228.254.196|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 23101 (23K) [text/csv]
Saving to: ‘loan_train.csv’

loan_train.csv      100%[===================>]  22.56K  --.-KB/s
in 0.01s

2020-02-23 09:45:54 (1.48 MB/s) - ‘loan_train.csv’ saved [23101/2310
1]
```

In [3]:

```
df = pd.read_csv('loan_train.csv')
df.head()
```

Out[3]:

	Unnamed: 0	Unnamed: 0.1	loan_status	Principal	terms	effective_date	due_date	age	educa
0	0	0	PAIDOFF	1000	30	9/8/2016	10/7/2016	45	Scho B
1	2	2	PAIDOFF	1000	30	9/8/2016	10/7/2016	33	Bech
2	3	3	PAIDOFF	1000	15	9/8/2016	9/22/2016	27	col
3	4	4	PAIDOFF	1000	30	9/9/2016	10/8/2016	28	col
4	6	6	PAIDOFF	1000	30	9/9/2016	10/8/2016	29	col

In [4]:

```
df.shape
```

Out[4]:

```
(346, 10)
```

In [5]:

```
df['due_date'] = pd.to_datetime(df['due_date'])
df['effective_date'] = pd.to_datetime(df['effective_date'])
df.head()
```

Out[5]:

	Unnamed: 0	Unnamed: 0.1	loan_status	Principal	terms	effective_date	due_date	age	educa
0	0	0	PAIDOFF	1000	30	2016-09-08	2016-10-07	45	I Scho Be
1	2	2	PAIDOFF	1000	30	2016-09-08	2016-10-07	33	Bech
2	3	3	PAIDOFF	1000	15	2016-09-08	2016-09-22	27	col
3	4	4	PAIDOFF	1000	30	2016-09-09	2016-10-08	28	col
4	6	6	PAIDOFF	1000	30	2016-09-09	2016-10-08	29	col

In [6]:

```
df['loan_status'].value_counts()
```

Out[6]:

```
PAIDOFF       260
COLLECTION     86
Name: loan_status, dtype: int64
```

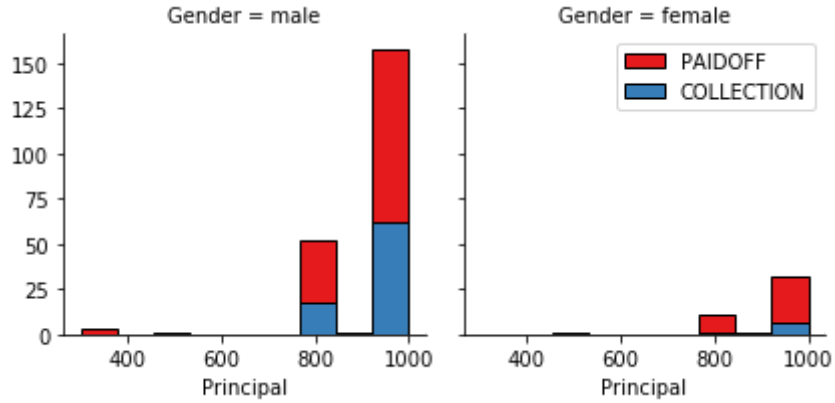
In [7]:

```
!conda install -c anaconda seaborn -y
```

```
Collecting package metadata: done
Solving environment: \ Killed
```

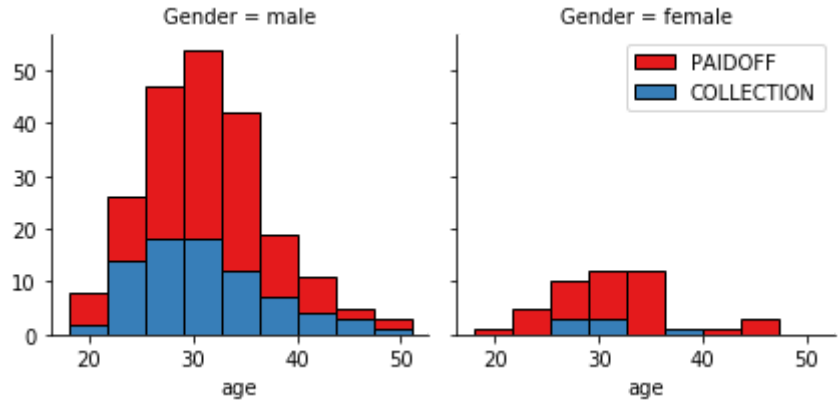
In [8]:

```
import seaborn as sns

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()
plt.show()
```



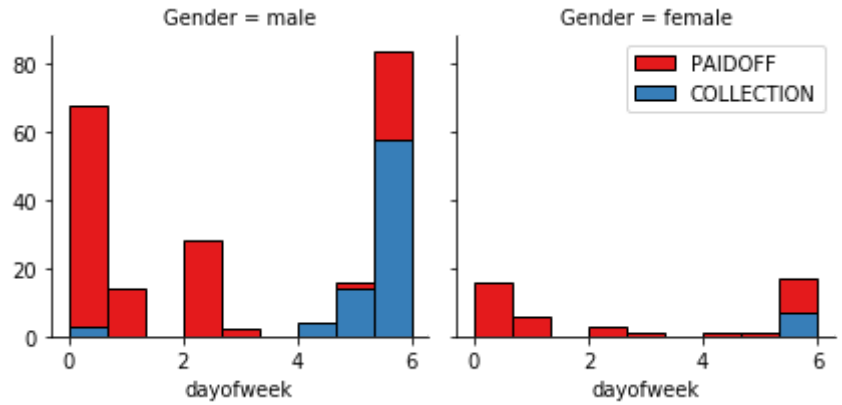
In [9]:

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

g.axes[-1].legend()
plt.show()
```



In [10]:

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



In [11]:

```
df['weekend'] = df['dayofweek'].apply(lambda x: 1 if (x>3)  else 0)
df.head()
```

Out[11]:

	Unnamed: 0	Unnamed: 0.1	loan_status	Principal	terms	effective_date	due_date	age	educa
0	0	0	PAIDOFF	1000	30	2016-09-08	2016-10-07	45	Scho Be
1	2	2	PAIDOFF	1000	30	2016-09-08	2016-10-07	33	Bech
2	3	3	PAIDOFF	1000	15	2016-09-08	2016-09-22	27	col
3	4	4	PAIDOFF	1000	30	2016-09-09	2016-10-08	28	col
4	6	6	PAIDOFF	1000	30	2016-09-09	2016-10-08	29	col

In [12]:

```
df.groupby(['Gender'])['loan_status'].value_counts(normalize=True)
```

Out[12]:

```
Gender  loan_status
female  PAIDOFF        0.865385
        COLLECTION     0.134615
male    PAIDOFF        0.731293
        COLLECTION     0.268707
Name: loan_status, dtype: float64
```

In [13]:

```
df['Gender'].replace(to_replace=['male','female'], value=[0,1],inplace=True)
df.head()
```

Out[13]:

	Unnamed: 0	Unnamed: 0.1	loan_status	Principal	terms	effective_date	due_date	age	educa
0	0	0	PAIDOFF	1000	30	2016-09-08	2016-10-07	45	Scho Be
1	2	2	PAIDOFF	1000	30	2016-09-08	2016-10-07	33	Bech
2	3	3	PAIDOFF	1000	15	2016-09-08	2016-09-22	27	col
3	4	4	PAIDOFF	1000	30	2016-09-09	2016-10-08	28	col
4	6	6	PAIDOFF	1000	30	2016-09-09	2016-10-08	29	col

In [14]:

```
df.groupby(['education'])['loan_status'].value_counts(normalize=True)
```

Out[14]:

```
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
```

In [15]:

```
df[['Principal','terms','age','Gender','education']].head()
```

Out[15]:

	Principal	terms	age	Gender	education
0	1000	30	45	0	High School or Below
1	1000	30	33	1	Bechalor
2	1000	15	27	0	college
3	1000	30	28	1	college
4	1000	30	29	0	college

In [16]:

```
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)
Feature.head()
```

Out[16]:

	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

In [17]:

```
X = Feature
X[0:5]
X.shape
```

Out[17]:

```
(346, 8)
```

In [18]:

```
y = df['loan_status'].values
y[0:5]
y.shape
```

Out[18]:

```
(346,)
```

In [19]:

```
X= preprocessing.StandardScaler().fit(X).transform(X)
X[0:5]
X.shape
```

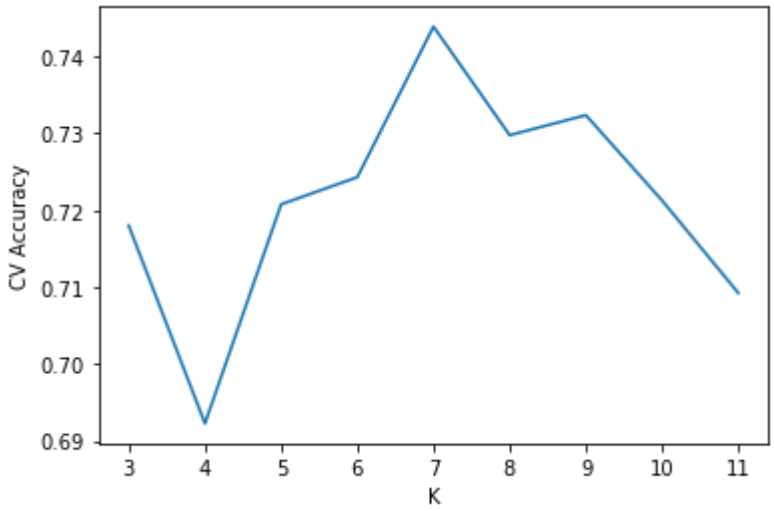
Out[19]:

```
(346, 8)
```

In [20]:

```
# knn
# import scoring methods
from sklearn.metrics import jaccard_similarity_score
from sklearn.metrics import f1_score
from sklearn.metrics import log_loss
from sklearn.model_selection import cross_val_score

X_train, y_train=X,y
# for adding scores
trainScores={}
print(X_train.shape)
print(y_train.shape)

from sklearn.neighbors import KNeighborsClassifier
bestScore=0.0
accList=[]

for k in range(3,12):

    clf_knn = KNeighborsClassifier(n_neighbors=k,algorithm='auto')

    # using 10 fold cross validation for scoring the classifier's accuracy
    scores = cross_val_score(clf_knn, X, y, cv=10)
    score=scores.mean()
    accList.append(score)

    if score > bestScore:
        bestScore=score
        best_clf=clf_knn
        bestK=k

print("Best K is :",bestK,"| Cross validation Accuracy :",bestScore)
clf_knn=best_clf

clf_knn.fit(X_train,y_train)
y_pred=best_clf.predict(X_train)
trainScores['KNN-jaccard']=jaccard_similarity_score(y_train, y_pred)
trainScores['KNN-f1-score']=f1_score(y_train, y_pred, average='weighted')
trainScores

#  plotting the curve for k values
plt.plot(range(3,12),accList)
plt.xlabel('K')
plt.ylabel('CV Accuracy')
plt.show()
```

```
(346, 8)
(346,)
Best K is : 7 | Cross validation Accuracy : 0.7438655462184873

/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/metric
s/classification.py:635: DeprecationWarning: jaccard_similarity_scor
e has been deprecated and replaced with jaccard_score. It will be re
moved in version 0.23. This implementation has surprising behavior f
or binary and multiclass classification tasks.
  'and multiclass classification tasks.', DeprecationWarning)
```



In [21]:

```
# Decision tree
from sklearn import tree

clf_tree = tree.DecisionTreeClassifier()
clf_tree = clf_tree.fit(X_train, y_train)

y_pred=clf_tree.predict(X_train)

trainScores['Tree-jaccard']=jaccard_similarity_score(y_train, y_pred)
trainScores['Tree-f1-score']=f1_score(y_train, y_pred, average='weighted')
trainScores
```

```
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/metric
s/classification.py:635: DeprecationWarning: jaccard_similarity_scor
e has been deprecated and replaced with jaccard_score. It will be re
moved in version 0.23. This implementation has surprising behavior f
or binary and multiclass classification tasks.
  'and multiclass classification tasks.', DeprecationWarning)
```

Out[21]:

```
{'KNN-jaccard': 0.8092485549132948,
 'KNN-f1-score': 0.7955764876306204,
 'Tree-jaccard': 0.8930635838150289,
 'Tree-f1-score': 0.8957643129154805}
```

In [22]:

```
# SVM
# y_train=y_train.astype(float)
from sklearn import svm
clf_svm = svm.LinearSVC(random_state=7)
clf_svm.fit(X_train,y_train)
y_pred = clf_svm.predict(X_train)

trainScores['SVM-jaccard']=jaccard_similarity_score(y_train, y_pred)
trainScores['SVM-f1-score']=f1_score(y_train, y_pred, average='weighted')
trainScores
```

```
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/svm/bas
e.py:929: ConvergenceWarning: Liblinear failed to converge, increase
the number of iterations.
  "the number of iterations.", ConvergenceWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/metric
s/classification.py:635: DeprecationWarning: jaccard_similarity_scor
e has been deprecated and replaced with jaccard_score. It will be re
moved in version 0.23. This implementation has surprising behavior f
or binary and multiclass classification tasks.
  'and multiclass classification tasks.', DeprecationWarning)
```

Out[22]:

```
{'KNN-jaccard': 0.8092485549132948,
 'KNN-f1-score': 0.7955764876306204,
 'Tree-jaccard': 0.8930635838150289,
 'Tree-f1-score': 0.8957643129154805,
 'SVM-jaccard': 0.7398843930635838,
 'SVM-f1-score': 0.6441316913797436}
```

In [23]:

```
# Logistic Regression
from sklearn.linear_model import LogisticRegression
clf_log = LogisticRegression(random_state=0, solver='lbfgs',
                         multi_class='multinomial')
clf_log.fit(X_train, y_train)

y_pred=clf_log.predict(X_train)
y_proba=clf_log.predict_proba(X_train)

trainScores['LogReg-jaccard']=jaccard_similarity_score(y_train, y_pred)
trainScores['LogReg-f1-score']=f1_score(y_train, y_pred, average='weighted')
trainScores['LogReg-logLoss']=log_loss(y_train, y_proba)
trainScores
```

```
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/metric
s/classification.py:635: DeprecationWarning: jaccard_similarity_scor
e has been deprecated and replaced with jaccard_score. It will be re
moved in version 0.23. This implementation has surprising behavior f
or binary and multiclass classification tasks.
  'and multiclass classification tasks.', DeprecationWarning)
```

Out[23]:

```
{'KNN-jaccard': 0.8092485549132948,
 'KNN-f1-score': 0.7955764876306204,
 'Tree-jaccard': 0.8930635838150289,
 'Tree-f1-score': 0.8957643129154805,
 'SVM-jaccard': 0.7398843930635838,
 'SVM-f1-score': 0.6441316913797436,
 'LogReg-jaccard': 0.7543352601156069,
 'LogReg-f1-score': 0.6997291283105058,
 'LogReg-logLoss': 0.43320342487163493}
```

In [24]:

```
# Evaluation
```

In [36]:

```
from sklearn.metrics import jaccard_similarity_score
from sklearn.metrics import f1_score
from sklearn.metrics import log_loss

!wget -O loan_test.csv https://s3-api.us-geo.objectstorage.softlayer.net/cf-cour
ses-data/CognitiveClass/ML0101ENv3/labs/loan_test.csv
```

```
--2020-02-23 10:05:21--  https://s3-api.us-geo.objectstorage.softlay
er.net/cf-courses-data/CognitiveClass/ML0101ENv3/labs/loan_test.csv
Resolving s3-api.us-geo.objectstorage.softlayer.net (s3-api.us-geo.o
bjectstorage.softlayer.net)... 67.228.254.196
Connecting to s3-api.us-geo.objectstorage.softlayer.net (s3-api.us-g
eo.objectstorage.softlayer.net)|67.228.254.196|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3642 (3.6K) [text/csv]
Saving to: ‘loan_test.csv’

loan_test.csv       100%[===================>]   3.56K  --.-KB/s
in 0s

2020-02-23 10:05:21 (426 MB/s) - ‘loan_test.csv’ saved [3642/3642]
```

In [37]:

```
test_df = pd.read_csv('loan_test.csv')
test_df.head()
```

Out[37]:

	Unnamed: 0	Unnamed: 0.1	loan_status	Principal	terms	effective_date	due_date	age	educa
0	1	1	PAIDOFF	1000	30	9/8/2016	10/7/2016	50	Bech
1	5	5	PAIDOFF	300	7	9/9/2016	9/15/2016	35	Mast Al
2	21	21	PAIDOFF	1000	30	9/10/2016	10/9/2016	43	Scho B
3	24	24	PAIDOFF	1000	30	9/10/2016	10/9/2016	26	col
4	35	35	PAIDOFF	800	15	9/11/2016	9/25/2016	29	Bech

In [41]:

```
# import seaborn as sns
# bins = np.linspace(df.Principal.min(), test_df.Principal.max(), 10)
# g = sns.FacetGrid(test_df, col="Gender", hue="loan_status", palette="Set1", co
l_wrap=2)
# g.map(plt.hist, 'Principal', bins=bins, ec="k")
# g.axes[-1].legend()
# plt.show()
```

In [42]:

```
# bins = np.linspace(test_df.age.min(), test_df.age.max(), 10)
# g = sns.FacetGrid(test_df, col="Gender", hue="loan_status", palette="Set1", co
l_wrap=2)
# g.map(plt.hist, 'age', bins=bins, ec="k")

# g.axes[-1].legend()
# plt.show()
```

In [43]:

```
# bins = np.linspace(test_df.dayofweek.min(), test_df.dayofweek.max(), 10)
# g = sns.FacetGrid(test_df, col="Gender", hue="loan_status", palette="Set1", co
l_wrap=2)
# g.map(plt.hist, 'dayofweek', bins=bins, ec="k")
# g.axes[-1].legend()
# plt.show()
```

In [47]:

```
test_df['due_date'] = pd.to_datetime(test_df['due_date'])
test_df['effective_date'] = pd.to_datetime(test_df['effective_date'])
test_df.head()
```

Out[47]:

	Unnamed: 0	Unnamed: 0.1	loan_status	Principal	terms	effective_date	due_date	age	educa
0	1	1	PAIDOFF	1000	30	2016-09-08	2016-10-07	50	Bech
1	5	5	PAIDOFF	300	7	2016-09-09	2016-09-15	35	Maste Ab
2	21	21	PAIDOFF	1000	30	2016-09-10	2016-10-09	43	Scho Be
3	24	24	PAIDOFF	1000	30	2016-09-10	2016-10-09	26	col
4	35	35	PAIDOFF	800	15	2016-09-11	2016-09-25	29	Bech

In [75]:

```
test_df['loan_status'].value_counts()
```

Out[75]:

```
PAIDOFF       40
COLLECTION    14
Name: loan_status, dtype: int64
```

In [54]:

```
test_df['dayofweek'] = df['effective_date'].dt.dayofweek
```

In [55]:

```
#  setting o if week day is above 3 and else 1
test_df['weekend'] = test_df['dayofweek'].apply(lambda x: 1 if (x>3)  else 0)
test_df.head()
```

Out[55]:

	Unnamed: 0	Unnamed: 0.1	loan_status	Principal	terms	effective_date	due_date	age	educa
0	1	1	PAIDOFF	1000	30	2016-09-08	2016-10-07	50	Bech
1	5	5	PAIDOFF	300	7	2016-09-09	2016-09-15	35	Maste Ab
2	21	21	PAIDOFF	1000	30	2016-09-10	2016-10-09	43	Scho Be
3	24	24	PAIDOFF	1000	30	2016-09-10	2016-10-09	26	col
4	35	35	PAIDOFF	800	15	2016-09-11	2016-09-25	29	Bech

In [56]:

```
# # #  setting o if week day is above 3 and else 1
# test_df['weekend'] = test_df['dayofweek'].apply(lambda x: 1 if (x>3)  else 0)
test_df.groupby(['Gender'])['loan_status'].value_counts(normalize=True)
```

Out[56]:

```
Gender  loan_status
female  PAIDOFF       0.727273
        COLLECTION    0.272727
male    PAIDOFF       0.744186
        COLLECTION    0.255814
Name: loan_status, dtype: float64
```

In [57]:

```
#  setting male to 0 and female to 1
test_df['Gender'].replace(to_replace=['male','female'], value=[0,1],inplace=True
)
test_df.head()
```

Out[57]:

	Unnamed: 0	Unnamed: 0.1	loan_status	Principal	terms	effective_date	due_date	age	educa
0	1	1	PAIDOFF	1000	30	2016-09-08	2016-10-07	50	Bech
1	5	5	PAIDOFF	300	7	2016-09-09	2016-09-15	35	Maste Ab
2	21	21	PAIDOFF	1000	30	2016-09-10	2016-10-09	43	Scho Be
3	24	24	PAIDOFF	1000	30	2016-09-10	2016-10-09	26	col
4	35	35	PAIDOFF	800	15	2016-09-11	2016-09-25	29	Bech

In [59]:

```
test_df.groupby(['education'])['loan_status'].value_counts(normalize=True)
```

Out[59]:

```
education             loan_status
Bechalor              PAIDOFF       1.000000
High School or Below  PAIDOFF       0.523810
                      COLLECTION    0.476190
Master or Above       PAIDOFF       1.000000
college               PAIDOFF       0.826087
                      COLLECTION    0.173913
Name: loan_status, dtype: float64
```

In [60]:

```
test_df[['Principal','terms','age','Gender','education']].head()
```

Out[60]:

	Principal	terms	age	Gender	education
0	1000	30	50	1	Bechalor
1	300	7	35	0	Master or Above
2	1000	30	43	1	High School or Below
3	1000	30	26	0	college
4	800	15	29	0	Bechalor

In [61]:

```
Feature = test_df[['Principal','terms','age','Gender','weekend']]
Feature = pd.concat([Feature,pd.get_dummies(test_df['education'])], axis=1)
Feature.drop(['Master or Above'], axis = 1,inplace=True)
Feature.head()
```

Out[61]:

	Principal	terms	age	Gender	weekend	Bechalor	High School or Below	college
0	1000	30	50	1	0	1	0	0
1	300	7	35	0	0	0	0	0
2	1000	30	43	1	0	0	1	0
3	1000	30	26	0	1	0	0	1
4	800	15	29	0	1	1	0	0

In [62]:

```
X_test = Feature
X_test[0:5]
```

Out[62]:

	Principal	terms	age	Gender	weekend	Bechalor	High School or Below	college
0	1000	30	50	1	0	1	0	0
1	300	7	35	0	0	0	0	0
2	1000	30	43	1	0	0	1	0
3	1000	30	26	0	1	0	0	1
4	800	15	29	0	1	1	0	0

In [63]:

```
y_test = df['loan_status'].values
y_test[0:5]
```

Out[63]:

```
array(['PAIDOFF', 'PAIDOFF', 'PAIDOFF', 'PAIDOFF', 'PAIDOFF'],
      dtype=object)
```

In [76]:

```
X_test= preprocessing.StandardScaler().fit(X_test).transform(X_test)
X_test[0:5]
```

Out[76]:

```
array([[ 0.51578458,  0.92071769,  2.33152555, -0.42056004, -1.20577
805,
        -0.38170062,  1.13639374, -0.86968108],
       [ 0.51578458,  0.92071769,  0.34170148,  2.37778177, -1.20577
805,
         2.61985426, -0.87997669, -0.86968108],
       [ 0.51578458, -0.95911111, -0.65321055, -0.42056004, -1.20577
805,
        -0.38170062, -0.87997669,  1.14984679],
       [ 0.51578458,  0.92071769, -0.48739188,  2.37778177,  0.82934
003,
        -0.38170062, -0.87997669,  1.14984679],
       [ 0.51578458,  0.92071769, -0.3215732 , -0.42056004,  0.82934
003,
        -0.38170062, -0.87997669,  1.14984679]])
```

In [77]:

```
from sklearn.metrics import jaccard_similarity_score
from sklearn.metrics import f1_score
from sklearn.metrics import log_loss
from sklearn.model_selection import cross_val_score
```

In [78]:

```
testScores={}
```

In [79]:

```
knn_pred=clf_knn.predict(X_test)
testScores['KNN-jaccard']=jaccard_similarity_score(y_test, knn_pred)
testScores['KNN-f1-score']=f1_score(y_test, knn_pred, average='weighted')
```

```
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/metric
s/classification.py:635: DeprecationWarning: jaccard_similarity_scor
e has been deprecated and replaced with jaccard_score. It will be re
moved in version 0.23. This implementation has surprising behavior f
or binary and multiclass classification tasks.
  'and multiclass classification tasks.', DeprecationWarning)
```

In [80]:

```
tree_pred=clf_tree.predict(X_test)
testScores['Tree-jaccard']=jaccard_similarity_score(y_test, tree_pred)
testScores['Tree-f1-score']=f1_score(y_test, tree_pred, average='weighted')
```

```
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/metric
s/classification.py:635: DeprecationWarning: jaccard_similarity_scor
e has been deprecated and replaced with jaccard_score. It will be re
moved in version 0.23. This implementation has surprising behavior f
or binary and multiclass classification tasks.
  'and multiclass classification tasks.', DeprecationWarning)
```

In [81]:

```
svm_pred=clf_svm.predict(X_test)
testScores['SVM-jaccard']=jaccard_similarity_score(y_test, svm_pred)
testScores['SVM-f1-score']=f1_score(y_test, svm_pred, average='weighted')
```

```
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/metric
s/classification.py:635: DeprecationWarning: jaccard_similarity_scor
e has been deprecated and replaced with jaccard_score. It will be re
moved in version 0.23. This implementation has surprising behavior f
or binary and multiclass classification tasks.
  'and multiclass classification tasks.', DeprecationWarning)
```

In [82]:

```
log_pred=clf_log.predict(X_test)
proba=clf_log.predict_proba(X_test)
testScores['LogReg-jaccard']=jaccard_similarity_score(y_test, log_pred)
testScores['LogReg-f1-score']=f1_score(y_test, log_pred, average='weighted')
testScores['LogReg-logLoss']=log_loss(y_test, proba)
```

```
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/metric
s/classification.py:635: DeprecationWarning: jaccard_similarity_scor
e has been deprecated and replaced with jaccard_score. It will be re
moved in version 0.23. This implementation has surprising behavior f
or binary and multiclass classification tasks.
  'and multiclass classification tasks.', DeprecationWarning)
```

In [83]:

```
trainScores
```

Out[83]:

```
{'KNN-jaccard': 0.8092485549132948,
 'KNN-f1-score': 0.7955764876306204,
 'Tree-jaccard': 0.8930635838150289,
 'Tree-f1-score': 0.8957643129154805,
 'SVM-jaccard': 0.7398843930635838,
 'SVM-f1-score': 0.6441316913797436,
 'LogReg-jaccard': 0.7543352601156069,
 'LogReg-f1-score': 0.6997291283105058,
 'LogReg-logLoss': 0.43320342487163493}
```

In [84]:

```
testScores
```

Out[84]:

```
{'KNN-jaccard': 0.8092485549132948,
 'KNN-f1-score': 0.7929774171543459,
 'Tree-jaccard': 0.8930635838150289,
 'Tree-f1-score': 0.8957643129154805,
 'SVM-jaccard': 0.7398843930635838,
 'SVM-f1-score': 0.6441316913797436,
 'LogReg-jaccard': 0.7543352601156069,
 'LogReg-f1-score': 0.6997291283105058,
 'LogReg-logLoss': 0.43320342487163493}
```

In [ ]:

```
```