moneybot

conversational interface to your finances

by #teamkernel

problem

easily access and analyze my transactions

know my spending habits

my bank and apps like pocketbook tdon't do a good enough job.

Build Classifier

In [6]:
```
clf =  GaussianNB()
fitted = clf.fit(x_train, y_train)
scores = cross_val_score(clf, x_train, y_train, cv=5)
print "Mean score: ", np.mean(scores), "\n", scores
```

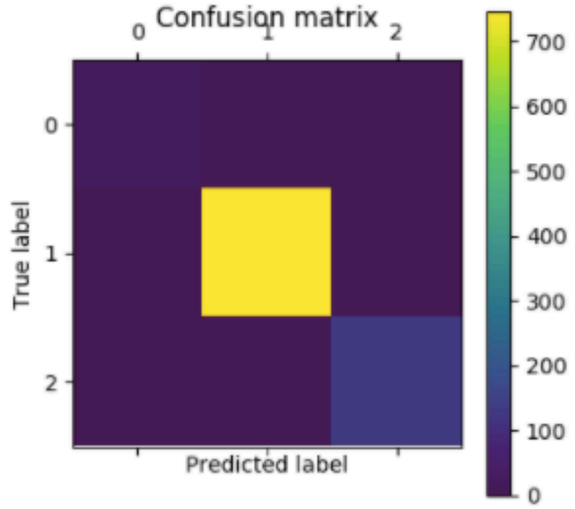
```
Mean score:  0.98797478434
[ 0.99090909  0.98363636  0.98905109  0.98722628  0.98905109]
```

Show confusion matrix

In [7]:
```
def displayConfusionMatrix(cmatrix):
    print cmatrix
    plt.matshow(cmatrix)
    plt.title('Confusion matrix')
    plt.colorbar()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    plt.show()
```

In [8]:
```
y_pred = clf.predict(x_test)
confusion_matrix = confusion_matrix(y_test, y_pred)
displayConfusionMatrix(confusion_matrix)
```

```
[[ 25   6   0]
 [  0 747   0]
 [  0   0 137]]
```



The confusion matrix above shows that Bendigo and CBA can be accurately predicated occassionally be classified incorrectly

know thy incoming data

```
import numpy as np
import pandas as pd
import datetime as dt
import os.path
import parseutils as pu

DATA_FILE_CBA='../data/CBA.csv'
DATA_FILE_BENDIGO='../data/Bendigo.csv'
DATA_FILE_ANZ='../data/ANZ.csv'
DATASET_FILE='../data/Dataset.csv'

COL_NAME_DATE='Date'
COL_NAME_AMOUNT='Amount'
COL_NAME_DESC='Description'
COL_NAME_BANK='Bank'

CSV_TYPE_CBA='CBA'
CSV_TYPE_BENDIGO='BENDIGO'
CSV_TYPE_ANZ='ANZ'
CSV_TYPE_UNKNOWN='UNKNOWN'
```

Check if data-set file already exists

Utility function to build dataset

Build the dataset combining all the raw data and normalizing column names, and adding label for identifying bank

[2]:
```
def buildDataSet():
    print "Building dataset..."
    def loadRawFiles():
        cba_raw=pd.read_csv(DATA_FILE_CBA, header=None)
        bendigo_raw=pd.read_csv(DATA_FILE_BENDIGO, header=None)
        anz_raw=pd.read_csv(DATA_FILE_ANZ, header=None)
        return cba_raw, bendigo_raw, anz_raw

    def addHeaders(df, columns,bankLabel):
        df.columns=columns
        df[COL_NAME_BANK]=bankLabel
        return df
    cba_raw, bendigo_raw, anz_raw = loadRawFiles()

    #add columns and label bank in a separate column
    cba_raw = addHeaders(cba_raw,[COL_NAME_DATE, COL_NAME_AMOUNT, COL_NAME_DESC], CSV_TYPE_CBA)
    bendigo_raw = addHeaders(bendigo_raw,[COL_NAME_DATE, COL_NAME_AMOUNT, COL_NAME_DESC], CSV_TYPE
_BENDIGO)
    anz_raw = addHeaders(anz_raw,[COL_NAME_DATE, COL_NAME_AMOUNT, COL_NAME_DESC], CSV_TYPE_ANZ)
    #print ("CBA\n===")
    #print (cba_raw.head(2))
    #print ("\n\nBENDIGO\n=======")
    #print (bendigo_raw.head(2))
    #print ("\n\nANZ\n===")
    #print (anz_raw.head(2))

    #combine all the resulting data into a single dataframe, and shuffle them
    return (pd.concat([cba_raw, bendigo_raw, anz_raw])).sample(frac=1)
```

Build dataset if necessary

[5]:
```
DATASET_FILE_EXISTS = os.path.isfile(DATASET_FILE)
dataset = None
if DATASET_FILE_EXISTS:
    print "Dataset file %s exists. \nLoading to pandas" % (DATASET_FILE)
    dataset = pd.read_csv(DATASET_FILE)
```


automated
parsing

transaction classification > 80%

```
In [129]: from sklearn.model_selection import train_test_split, cross_val_score
          X_train, X_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.15, random_state=0)
          #X_train, X_test, y_train, y_test = train_test_split(x_reduced, y_data, test_size=0.15, random_state=0)
          clf = sklearn.linear_model.LogisticRegression()
          clf.fit(X_train,y_train)

Out[129]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                    intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                    penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                    verbose=0, warm_start=False)

In [130]: clf.score(X_test, y_test)

Out[130]: 0.85491419656786272

In [131]: scores = cross_val_score(clf, X_test, y_test, cv=5)
          scores

          /Users/ko/anaconda/lib/python3.6/site-packages/sklearn/model_selection/_split.py:581: Warning: The least populated cl
          ass in y has only 1 members, which is too few. The minimum number of groups for any class cannot be less than n_split
          s=5.
            % (min_groups, self.n_splits)), Warning)

Out[131]: array([ 0.80740741,  0.7480916 ,  0.796875  ,  0.8       ,  0.81147541])
```

why does my bank suck at classifying my transactions?

features, features, features![1]

» used word2vec and sklearn to classify

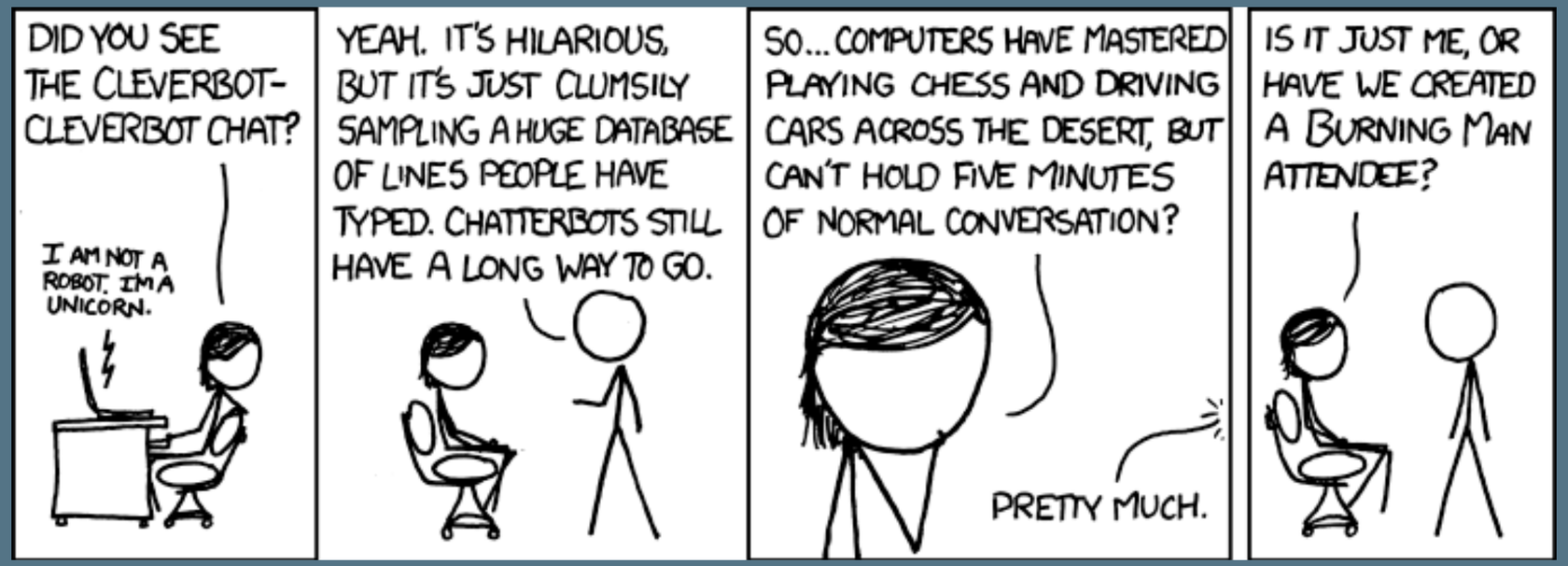
» turns out the big guys use yellow pages and humans

» we used a number of algos from sklearn

---

[1] ok, maybe not so many

need NLP to understand text



» we're using facebook's wit.ai engine to parse text

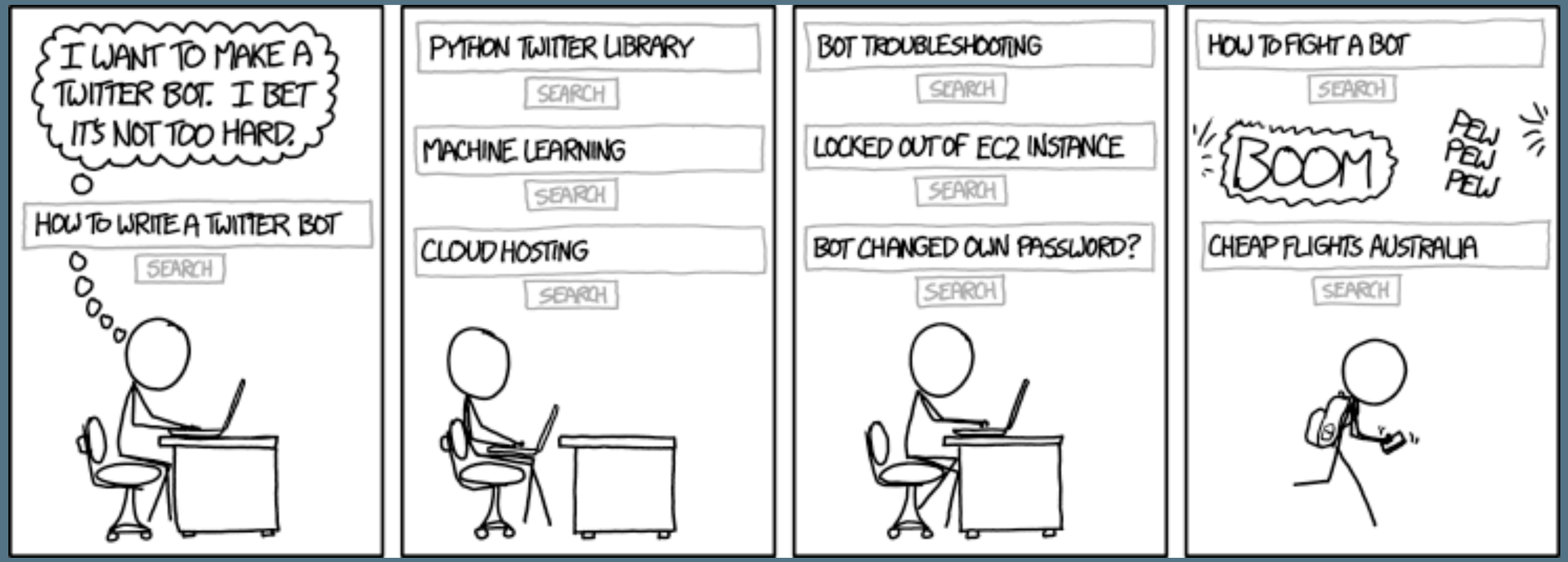
» fb AI promises the world, but needs a lot of work [^fb]

short demo

moneybot, where art thou?

handle more intents in the future

by using a LSTM NN to connect incoming intents to actions with memory



of course, a few years from now…

TeamKernel, 2017

questions?