File: ./pca/eigenfaces.py

```
1    """
2    ===================================================
3    Faces recognition example using eigenfaces and SVMs
4    ===================================================
5
6    The dataset used in this example is a preprocessed excerpt of the
7    "Labeled Faces in the Wild", aka LFW_:
8
9      http://vis-www.cs.umass.edu/lfw/lfw-funneled.tgz (233MB)
10
11      .. _LFW: http://vis-www.cs.umass.edu/lfw/
12
13      original source: http://scikit-learn.org/stable/auto_examples/applications/face_recognition.html
14
15    """
16
17
18
19   print(__doc__)
20
21   from time import time
22   import logging
23   import pylab as pl
24   import numpy as np
25
26   from sklearn.model_selection import train_test_split
27   from sklearn.model_selection import GridSearchCV
28   from sklearn.datasets import fetch_lfw_people
29   from sklearn.metrics import classification_report
30   from sklearn.metrics import confusion_matrix
31   from sklearn.decomposition import PCA
32   from sklearn.svm import SVC
33
34   # Display progress logs on stdout
35   logging.basicConfig(level=logging.INFO, format='%(asctime)s %(message)s')
36
37
38   ###############################################################################
39   # Download the data, if not already on disk and load it as numpy arrays
40   lfw_people = fetch_lfw_people(min_faces_per_person=70, resize=0.4)
41
42   # introspect the images arrays to find the shapes (for plotting)
43   n_samples, h, w = lfw_people.images.shape
44   np.random.seed(42)
45
46   # for machine learning we use the data directly (as relative pixel
47   # position info is ignored by this model)
48   X = lfw_people.data
49   n_features = X.shape[1]
50
51   # the label to predict is the id of the person
52   y = lfw_people.target
53   target_names = lfw_people.target_names
54   n_classes = target_names.shape[0]
55
56   print("Total dataset size:")
57   print("n_samples: %d" % n_samples)
58   print("n_features: %d" % n_features)
59   print("n_classes: %d" % n_classes)
60
61
62   ###############################################################################
63   # Split into a training and testing set
64   X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
65
66   ###############################################################################
67   # Compute a PCA (eigenfaces) on the face dataset (treated as unlabeled
68   # dataset): unsupervised feature extraction / dimensionality reduction
69   n_components = 150
70
71   print("Extracting the top %d eigenfaces from %d faces" % (n_components, X_train.shape[0]))
72   t0 = time()
73   pca = PCA(n_components=n_components, whiten=True, svd_solver='randomized').fit(X_train)
74   print("done in %0.3fs" % (time() - t0))
75
76   eigenfaces = pca.components_.reshape((n_components, h, w))
77
78   print("Projecting the input data on the eigenfaces orthonormal basis")
79   t0 = time()
80   X_train_pca = pca.transform(X_train)
81   X_test_pca = pca.transform(X_test)
82   print("done in %0.3fs" % (time() - t0))
83
84
85   ###############################################################################
86   # Train a SVM classification model
87
88   print("Fitting the classifier to the training set")
89   t0 = time()
90   param_grid = {
91            'C': [1e3, 5e3, 1e4, 5e4, 1e5],
92            'gamma': [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1],
93            }
94   # for sklearn version 0.16 or prior, the class_weight parameter value is 'auto'
95   clf = GridSearchCV(SVC(kernel='rbf', class_weight='balanced'), param_grid)
96   clf = clf.fit(X_train_pca, y_train)
97   print("done in %0.3fs" % (time() - t0))
98   print("Best estimator found by grid search:")
99   print(clf.best_estimator_)
100
101
102  ###############################################################################
103  # Quantitative evaluation of the model quality on the test set
104
105  print("Predicting the people names on the testing set")
106  t0 = time()
107  y_pred = clf.predict(X_test_pca)
108  print("done in %0.3fs" % (time() - t0))
109
110  print(classification_report(y_test, y_pred, target_names=target_names))
111  print(confusion_matrix(y_test, y_pred, labels=range(n_classes)))
112
```

```
113
114  ###############################################################################
115  # Qualitative evaluation of the predictions using matplotlib
116
117  def plot_gallery(images, titles, h, w, n_row=3, n_col=4):
118      """Helper function to plot a gallery of portraits"""
119      pl.figure(figsize=(1.8 * n_col, 2.4 * n_row))
120      pl.subplots_adjust(bottom=0, left=.01, right=.99, top=.90, hspace=.35)
121      for i in range(n_row * n_col):
122          pl.subplot(n_row, n_col, i + 1)
123          pl.imshow(images[i].reshape((h, w)), cmap=pl.cm.gray)
124          pl.title(titles[i], size=12)
125          pl.xticks(())
126          pl.yticks(())
127
128
129  # plot the result of the prediction on a portion of the test set
130
131  def title(y_pred, y_test, target_names, i):
132      pred_name = target_names[y_pred[i]].rsplit(' ', 1)[-1]
133      true_name = target_names[y_test[i]].rsplit(' ', 1)[-1]
134      return 'predicted: %s\ntrue:      %s' % (pred_name, true_name)
135
136  prediction_titles = [title(y_pred, y_test, target_names, i)
137                       for i in range(y_pred.shape[0])]
138
139  plot_gallery(X_test, prediction_titles, h, w)
140
141  # plot the gallery of the most significative eigenfaces
142
143  eigenface_titles = ["eigenface %d" % i for i in range(eigenfaces.shape[0])]
144  plot_gallery(eigenfaces, eigenface_titles, h, w)
145
146  pl.show()
```

File: ./naive_bayes/nb_author_id.py

```
1   #!/usr/bin/python
2
3   """
4       This is the code to accompany the Lesson 1 (Naive Bayes) mini-project.
5
6       Use a Naive Bayes Classifier to identify emails by their authors
7
8       authors and labels:
9       Sara has label 0
10      Chris has label 1
11  """
12
13  import sys
14  from time import time
15  sys.path.append("../tools/")
16  from email_preprocess import preprocess
17
18
19  ### features_train and features_test are the features for the training
20  ### and testing datasets, respectively
21  ### labels_train and labels_test are the corresponding item labels
22  features_train, features_test, labels_train, labels_test = preprocess()
23
24
25
26
27  #########################################################
28  ### your code goes here ###
29
30
31  #########################################################
32
33
```

File: ./tools/feature_format.py

```
1   #!/usr/bin/python
2
3   """
4       A general tool for converting data from the
5       dictionary format to an (n x k) python list that's
6       ready for training an sklearn algorithm
7
8       n--no. of key-value pairs in dictonary
9       k--no. of features being extracted
10
11      dictionary keys are names of persons in dataset
12      dictionary values are dictionaries, where each
13          key-value pair in the dict is the name
14          of a feature, and its value for that person
15
16      In addition to converting a dictionary to a numpy
17      array, you may want to separate the labels from the
18      features--this is what targetFeatureSplit is for
19
20      so, if you want to have the poi label as the target,
21      and the features you want to use are the person's
22      salary and bonus, here's what you would do:
23
24      feature_list = ["poi", "salary", "bonus"]
25      data_array = featureFormat( data_dictionary, feature_list )
26      label, features = targetFeatureSplit(data_array)
27
28      the line above (targetFeatureSplit) assumes that the
29      label is the _first_ item in feature_list--very important
30      that poi is listed first!
31  """
32
33
34  import numpy as np
35
36  def featureFormat( dictionary, features, remove_NaN=True, remove_all_zeroes=True, remove_any_zeroes=False, sort_keys = False):
37      """ convert dictionary to numpy array of features
```

```python
            convert dictionary to numpy array of features
            remove_NaN = True will convert "NaN" string to 0.0
            remove_all_zeroes = True will omit any data points for which
                all the features you seek are 0.0
            remove_any_zeroes = True will omit any data points for which
                any of the features you seek are 0.0
            sort_keys = True sorts keys by alphabetical order. Setting the value as
                a string opens the corresponding pickle file with a preset key
                order (this is used for Python 3 compatibility, and sort_keys
                should be left as False for the course mini-projects).
            NOTE: first feature is assumed to be 'poi' and is not checked for
                removal for zero or missing values.
        """


        return_list = []

        # Key order - first branch is for Python 3 compatibility on mini-projects,
        # second branch is for compatibility on final project.
        if isinstance(sort_keys, str):
            import pickle
            keys = pickle.load(open(sort_keys, "rb"))
        elif sort_keys:
            keys = sorted(dictionary.keys())
        else:
            keys = dictionary.keys()

        for key in keys:
            tmp_list = []
            for feature in features:
                try:
                    dictionary[key][feature]
                except KeyError:
                    print ("error: key ", feature, " not present")
                    return
                value = dictionary[key][feature]
                if value=="NaN" and remove_NaN:
                    value = 0
                tmp_list.append( float(value) )

            # Logic for deciding whether or not to add the data point.
            append = True
            # exclude 'poi' class as criteria.
            if features[0] == 'poi':
                test_list = tmp_list[1:]
            else:
                test_list = tmp_list
            ### if all features are zero and you want to remove
            ### data points that are all zero, do that here
            if remove_all_zeroes:
                append = False
                for item in test_list:
                    if item != 0 and item != "NaN":
                        append = True
                        break
            ### if any features for a given data point are zero
            ### and you want to remove data points with any zeroes,

            ### handle that here
            if remove_any_zeroes:
                if 0 in test_list or "NaN" in test_list:
                    append = False
            ### Append the data point if flagged for addition.
            if append:
                return_list.append( np.array(tmp_list) )

    return np.array(return_list)


def targetFeatureSplit( data ):
    """
        given a numpy array like the one returned from
        featureFormat, separate out the first feature
        and put it into its own list (this should be the
        quantity you want to predict)

        return targets and features as separate lists

        (sklearn can generally handle both lists and numpy arrays as
        input formats when training/predicting)
    """

    target = []
    features = []
    for item in data:
        target.append( item[0] )
        features.append( item[1:] )

    return target, features
```

File: ./tools/startup.py

```python
#!/usr/bin/python
import time
import sys

if sys.version_info[0] >= 3:
    from urllib.request import urlretrieve
else:
    # Not Python 3 - today, it is most likely to be Python 2
    # But note that this might need an update when Python 4
    # might be around one day
    from urllib import urlretrieve

print()
print("checking for nltk")
try:
    import nltk
except ImportError:
```

```
18     print("you should install nltk before continuing")
19
20  print("checking for numpy")
21  try:
22      import numpy
23  except ImportError:
24      print("you should install numpy before continuing")
25
26  print("checking for scipy")
27  try:
28      import scipy
29  except:
30      print("you should install scipy before continuing")
31
32  print("checking for sklearn")
33  try:
34      import sklearn
35  except:
36      print("you should install sklearn before continuing")
37
38  print()
39  print("downloading the Enron dataset (this may take a while)")
40  print("to check on progress, you can cd up one level, then execute <ls -lthr>")
41  print("Enron dataset should be last item on the list, along with its current size")
42  print("download will complete at about 423 MB")
43
44
45  def reporthook(count, block_size, total_size):
46      """
47      Callback for displaying progress bar while downloading the enron dataset.
48      Thanks to Pushkaraj Shinde https://github.com/udacity/ud120-projects/pull/55
49      For further information about parameters see documentation of urlretrieve,
50      e.g. https://docs.python.org/3.5/library/urllib.request.html#urllib.request.urlretrieve
51      """
52      global start_time
53      if count == 0:
54          start_time = time.time()
55          return
56      duration = time.time() - start_time
57      progress_size = int(count * block_size)
58      speed = int(progress_size / (1024 * duration))
59      percent = min(int(count * block_size * 100 / total_size), 100)
60      sys.stdout.write("\rEnron dataset: Downloaded %d%%, %d MB, %d KB/s, %d seconds passed" %
61                      (percent, progress_size / (1024 * 1024), speed, duration))
62      sys.stdout.flush()
63
64  url = "https://www.cs.cmu.edu/~./enron/enron_mail_20150507.tgz"
65
66  urlretrieve(url, "../enron_mail_20150507.tgz", reporthook)
67  print("download complete!")
68
69
70  print()
71  print("unzipping Enron dataset (this may take a while)")
72  import tarfile
73  import os
74  os.chdir("..")
75  tfile = tarfile.open("enron_mail_20150507.tar.gz", "r:gz")
76  tfile.extractall(".")
77
78  print("you're ready to go!")
```

File: ./tools/parse_out_email_text.py

```
1   #!/usr/bin/python
2
3   from nltk.stem.snowball import SnowballStemmer
4   import string
5   import re
6
7   def parseOutText(f):
8       """ given an opened email file f, parse out all text below the
9           metadata block at the top
10          (in Part 2, you will also add stemming capabilities)
11          and return a string that contains all the words
12          in the email (space-separated)
13
14          example use case:
15          f = open("email_file_name.txt", "r")
16          text = parseOutText(f)
17
18          """
19
20
21      f.seek(0)  ### go back to beginning of file (annoying)
22      all_text = f.read()
23
24      ### split off metadata
25      content = all_text.split("X-FileName:")
26      words = ""
27      if len(content) > 1:
28          ### remove punctuation
29          #text_string = re.sub('[' + string.punctuation + ']', '', content[1])
30          text_string = content[1].translate(str.maketrans("", "", string.punctuation))
31
32          ### project part 2: comment out the line below
33          #words = text_string
34
35          ### split the text string into individual words, stem each word,
36          ### and append the stemmed word to words (make sure there's a single
37          ### space between each stemmed word)
38          stemmer = SnowballStemmer("english")
39          word_list = []
40
41          for word in text_string.split():
42              word_list.append(stemmer.stem(word))
43
44          words = ' '.join(word_list)
45      return words
46
47
48
```

```
49  def main():
50      ff = open("../text_learning/test_email.txt", "r")
51      text = parseOutText(ff)
52      print(text)
53
54
55
56  if __name__ == '__main__':
57      main()
58
```

File: ./tools/email_preprocess.py

```
1   #!/usr/bin/python
2
3   import pickle
4   #import cPickle
5   import numpy
6
7   from sklearn.feature_extraction.text import TfidfVectorizer
8   from sklearn.feature_selection import SelectPercentile, f_classif
9   from sklearn.model_selection import train_test_split
10
11
12  def preprocess(words_file = "../tools/word_data.pkl", authors_file="../tools/email_authors.pkl"):
13      """
14          this function takes a pre-made list of email texts (by default word_data.pkl)
15          and the corresponding authors (by default email_authors.pkl) and performs
16          a number of preprocessing steps:
17              -- splits into training/testing sets (10% testing)
18              -- vectorizes into tfidf matrix
19              -- selects/keeps most helpful features
20
21          after this, the feaures and labels are put into numpy arrays, which play nice with sklearn functions
22
23          4 objects are returned:
24              -- training/testing features
25              -- training/testing labels
26
27      """
28
29      ### the words (features) and authors (labels), already largely preprocessed
30      ### this preprocessing will be repeated in the text learning mini-project
31      authors_file_handler = open(authors_file, "rb")
32      authors = pickle.load(authors_file_handler)
33      authors_file_handler.close()
34
35      words_file_handler = open(words_file, "rb")
36      word_data = pickle.load(words_file_handler)
37      words_file_handler.close()
38
39      ### test_size is the percentage of events assigned to the test set
40      ### (remainder go into training)
41      features_train, features_test, labels_train, labels_test =\
42          train_test_split(word_data, authors, test_size=0.1, random_state=42)
43
44
45
46      ### text vectorization--go from strings to lists of numbers
47      vectorizer = TfidfVectorizer(sublinear_tf=True, max_df=0.5,
48                                   stop_words='english')
49      features_train_transformed = vectorizer.fit_transform(features_train)
50      features_test_transformed  = vectorizer.transform(features_test)
51
52
53      ### feature selection, because text is super high dimensional and
54      ### can be really computationally chewy as a result
55      selector = SelectPercentile(f_classif, percentile=10)
56      selector.fit(features_train_transformed, labels_train)
57      features_train_transformed = selector.transform(features_train_transformed).toarray()
58      features_test_transformed  = selector.transform(features_test_transformed).toarray()
59
60
61      ### info on the data
62      print ("no. of Chris training emails:", sum(labels_train))
63      print ("no. of Sara training emails:", len(labels_train)-sum(labels_train))
64
65      return features_train_transformed, features_test_transformed, labels_train, labels_test
```

File: ./text_learning/vectorize_text.py

```
1   #!/usr/bin/python
2
3   import os
4   import pickle
5   import re
6   import sys
7
8   sys.path.append( "../tools/" )
9   from parse_out_email_text import parseOutText
10
11  """
12      Starter code to process the emails from Sara and Chris to extract
13      the features and get the documents ready for classification.
14
15      The list of all the emails from Sara are in the from_sara list
16      likewise for emails from Chris (from_chris)
17
18      The actual documents are in the Enron email dataset, which
19      you downloaded/unpacked in Part 0 of the first mini-project. If you have
20      not obtained the Enron email corpus, run startup.py in the tools folder.
21
22      The data is stored in lists and packed away in pickle files at the end.
23  """
24
25
26  from_sara  = open("from_sara.txt", "r")
27  from_chris = open("from_chris.txt", "r")
28
29  from_data = []
```

```
29  from_data = []
30  word_data = []
31
32  ### temp_counter is a way to speed up the development--there are
33  ### thousands of emails from Sara and Chris, so running over all of them
34  ### can take a long time
35  ### temp_counter helps you only look at the first 200 emails in the list so you
36  ### can iterate your modifications quicker
37  temp_counter = 0
38
39
40  for name, from_person in [("sara", from_sara), ("chris", from_chris)]:
41      for path in from_person:
42          ### only look at first 200 emails when developing
43          ### once everything is working, remove this line to run over full dataset
44          temp_counter += 1
45          if temp_counter < 200:
46              path = os.path.join('..', path[:-1])
47              print(path)
48              email = open(path, "r")
49
50              ### use parseOutText to extract the text from the opened email
51
52              ### use str.replace() to remove any instances of the words
53              ### ["sara", "shackleton", "chris", "germani"]
54
55              ### append the text to word_data
56
57              ### append a 0 to from_data if email is from Sara, and 1 if email is from Chris
58
59
60              email.close()
61
62  print("emails processed")
63  from_sara.close()
64  from_chris.close()
65
66  pickle.dump(word_data, open("your_word_data.pkl", "wb"))
67  pickle.dump(from_data, open("your_email_authors.pkl", "wb"))
68
69
70
71
72
73  ### in Part 4, do TfIdf vectorization here
74
75
```

File: ./outliers/outlier_cleaner.py

```
1   #!/usr/bin/python
2
3
4   def outlierCleaner(predictions, ages, net_worths):
5       """
6           Clean away the 10% of points that have the largest
7           residual errors (difference between the prediction
8           and the actual net worth).
9
10          Return a list of tuples named cleaned_data where
11          each tuple is of the form (age, net_worth, error).
12      """
13
14      cleaned_data = []
15
16      ### your code goes here
17
18
19      return cleaned_data
20
```

File: ./outliers/enron_outliers.py

```
1   #!/usr/bin/python
2
3   import pickle
4   import sys
5   import matplotlib.pyplot
6   sys.path.append("../tools/")
7   from feature_format import featureFormat, targetFeatureSplit
8
9
10  ### read in data dictionary, convert to numpy array
11  data_dict = pickle.load( open("../final_project/final_project_dataset.pkl", "rb") )
12  features = ["salary", "bonus"]
13  data = featureFormat(data_dict, features)
14
15
16  ### your code below
17
18
19
```

File: ./outliers/outlier_removal_regression.py

```
1   #!/usr/bin/python
2
3   import random
4   import numpy
5   import matplotlib.pyplot as plt
6   import pickle
7
8   from outlier_cleaner import outlierCleaner
9
10
11  ### load up some practice data with outliers in it
12  ages = pickle.load(open("practice_outliers_ages.pkl", "rb"))
13  net_worths = pickle.load(open("practice_outliers_net_worths.pkl", "rb"))
```

```python
net_worths = pickle.load( open("practice_outliers_net_worths.pkl", "rb") )




### ages and net_worths need to be reshaped into 2D numpy arrays
### second argument of reshape command is a tuple of integers: (n_rows, n_columns)
### by convention, n_rows is the number of data points
### and n_columns is the number of features
ages       = numpy.reshape( numpy.array(ages), (len(ages), 1))
net_worths = numpy.reshape( numpy.array(net_worths), (len(net_worths), 1))
from sklearn.model_selection import train_test_split
ages_train, ages_test, net_worths_train, net_worths_test = train_test_split(ages, net_worths, test_size=0.1, random_state=42)

### fill in a regression here!  Name the regression object reg so that
### the plotting code below works, and you can see what your regression looks like
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
reg.fit(ages_train, net_worths_train)
print("coefficient of determination R^2 on train data {:.3}"
      .format(reg.score(ages_test, net_worths_test)))


try:
    plt.plot(ages, reg.predict(ages), color="blue")
except NameError:
    pass
plt.scatter(ages, net_worths)
plt.show()


### identify and remove the most outlier-y points
cleaned_data = []
try:
    predictions = reg.predict(ages_train)
    cleaned_data = outlierCleaner( predictions, ages_train, net_worths_train )
except NameError:
    print("your regression object doesn't exist, or isn't name reg")
    print("can't make predictions to use in identifying outliers")






### only run this code if cleaned_data is returning data
if len(cleaned_data) > 0:
    ages, net_worths, errors = zip(*cleaned_data)
    ages       = numpy.reshape( numpy.array(ages), (len(ages), 1))
    net_worths = numpy.reshape( numpy.array(net_worths), (len(net_worths), 1))

    ### refit your cleaned data!
    try:
        reg.fit(ages, net_worths)
        plt.plot(ages, reg.predict(ages), color="blue")
        print("coefficient of determination R^2 on train data {:.3}"
            .format(reg.score(ages_test, net_worths_test)))
    except NameError:
        print("you don't seem to have regression imported/created,")
        print("   or else your regression object isn't named reg")
        print("   either way, only draw the scatter plot of the cleaned data")
    plt.scatter(ages, net_worths)
    plt.xlabel("ages")
    plt.ylabel("net worths")
    plt.show()


else:
    print("outlierCleaner() is returning an empty list, no refitting to be done")

```

File: ./validation/validate_poi.py

```python
#!/usr/bin/python


"""
    Starter code for the validation mini-project.
    The first step toward building your POI identifier!

    Start by loading/formatting the data

    After that, it's not our code anymore--it's yours!
"""

import pickle
import sys
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
sys.path.append("../tools/")
from feature_format import featureFormat, targetFeatureSplit
from mylib import fit_and_predict

data_dict = pickle.load(open("../final_project/final_project_dataset.pkl", "rb"))

### first element is our labels, any added elements are predictor
### features. Keep this the same for the mini-project, but you'll
### have a different feature list when you do the final project.
sort_keys = '../tools/python2_lesson13_keys.pkl'
features_list = ["poi", "salary"]

data = featureFormat(data_dict, features_list)
labels, features = targetFeatureSplit(data)

clf = DecisionTreeClassifier(random_state=42)
acc = fit_and_predict(clf, features, features, labels, labels)
print("Overfitted {0:.3}".format(acc))

clf = DecisionTreeClassifier(random_state=42)
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.30, random_state=42, shuffle=False)
acc = fit_and_predict(clf, X_train, X_test, y_train, y_test)
print("Accuracy of test {0:.3}".format(acc))

```

```
41 ### it's all yours from here forward!
42
43
```

File: ./.code2pdf

```
:directories:
- .git

:files:
- .txt
```

File: ./svm/svm_author_id.py

```python
1   #!/usr/bin/python
2
3   """
4       This is the code to accompany the Lesson 2 (SVM) mini-project.
5
6       Use a SVM to identify emails from the Enron corpus by their authors:
7       Sara has label 0
8       Chris has label 1
9   """
10
11  import sys
12  from time import time
13  sys.path.append("../tools/")
14  from email_preprocess import preprocess
15
16
17  ### features_train and features_test are the features for the training
18  ### and testing datasets, respectively
19  ### labels_train and labels_test are the corresponding item labels
20  features_train, features_test, labels_train, labels_test = preprocess()
21
22
23
24
25  #########################################################
26  ### your code goes here ###
27
28  #########################################################
29
30
```

File: ./small_task/count_stop_words.py

```python
1 from nltk.corpus import stopwords
2
3 sw = stopwords.words("english")
4 print(len(sw))
5 print(sw)
```

File: ./datasets_questions/explore_enron_data.py

```python
1   #!/usr/bin/python
2
3   """
4       Starter code for exploring the Enron dataset (emails + finances);
5       loads up the dataset (pickled dict of dicts).
6
7       The dataset has the form:
8       enron_data["LASTNAME FIRSTNAME MIDDLEINITIAL"] = { features_dict }
9
10      {features_dict} is a dictionary of features associated with that person.
11      You should explore features_dict as part of the mini-project,
12      but here's an example to get you started:
13
14      enron_data["SKILLING JEFFREY K"]["bonus"] = 5600000
15
16  """
17
18  import pickle
19
20  enron_data = pickle.load(open("../final_project/final_project_dataset.pkl", "rb"))
21
22
```

File: ./datasets_questions/datasets_questions.ipynb

```
{
"cells": [
{
"cell_type": "code",
"execution_count": 4,
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"Done. Saved 6705 bytes.\n",
"Done. Saved 17578 bytes.\n",
"Done. Saved 6564 bytes.\n",
"Done. Saved 286 bytes.\n"
]
}
],
```

```
"source": [
"#!/usr/bin/env python\n",
"\"\"\"\\n",
"convert dos linefeeds (crlf) to unix (lf)\n",
"usage: dos2unix.py [_____] \n",
"\"\"\"\n",
"original_files=[\"../final_project/final_project_dataset.pkl\",\n",
" \"../tools/email_authors.pkl\",\n",
" \"../final_project/final_project_dataset_modified.pkl\",\n",
" \"../tools/python2_lesson06_keys.pkl\"]\n",
"destination_files =[ \"../final_project/final_project_dataset_py3.pkl\",\n",
" \"../tools/email_authors_py3.pkl\",\n",
" \"../final_project/final_project_dataset_modified_py3.pkl\",\n",
" \"../tools/python2_lesson06_keys_py3.pkl\"]\n",
"\n",
"for original,destination in zip(original_files,destination_files):\n",
" content = ''\n",
" outsize = 0\n",
" with open(original, 'rb') as infile:\n",
" content = infile.read()\n",
" with open(destination, 'wb') as output:\n",
" for line in content.splitlines():\n",
" outsize += outsize+ len(line) + 1\n",
" output.write(line + str.encode('\\n'))\n",
" print(\"Done. Saved %s bytes.\" % (len(content)-outsize))"
]
},
{
"cell_type": "code",
"execution_count": 6,
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"b'\\n'"
]
},
"execution_count": 6,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"str.encode('\\n')"
]
},
{
"cell_type": "code",
"execution_count": 51,
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"../final_project/final_project_dataset.pkl\n",
"../final_project/final_project_dataset_unix.pkl\n",
"../tools/email_authors.pkl\n",
"../tools/email_authors_unix.pkl\n"
]
}
],
"source": [
"original_files=[\"../final_project/final_project_dataset.pkl\",\"../tools/email_authors.pkl\"]\n",
"destination_files =[ \"../final_project/final_project_dataset_unix.pkl\",\"../tools/email_authors_unix.pkl\"]\n",
"for original,destination in zip(original_files,destination_files):\n",
" print(original)\n",
" print(destination)\n"
]
},
{
"cell_type": "code",
"execution_count": 106,
"metadata": {},
"outputs": [],
"source": [
"import pickle"
]
},
{
"cell_type": "code",
"execution_count": 107,
"metadata": {},
"outputs": [],
"source": [
"enron_data = pickle.load(open(\"../final_project/final_project_dataset_py3.pkl\", \"rb\"))"
]
},
{
"cell_type": "code",
"execution_count": 108,
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"dict"
]
},
"execution_count": 108,
"metadata": {},
```

"output_type": "execute_result"
}
],
"source": [
"type(enron_data)"
]
},
{
"cell_type": "code",
"execution_count": 116,
"metadata": {
"collapsed": true
},
"outputs": [
{
"data": {
"text/plain": [
"{'ALLEN PHILLIP K': {'bonus': 4175000,\n",
" 'deferral_payments': 2869717,\n",
" 'deferred_income': -3081055,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'phillip.allen@enron.com',\n",
" 'exercised_stock_options': 1729541,\n",
" 'expenses': 13868,\n",
" 'from_messages': 2195,\n",
" 'from_poi_to_this_person': 47,\n",
" 'from_this_person_to_poi': 65,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 304805,\n",
" 'other': 152,\n",
" 'poi': False,\n",
" 'restricted_stock': 126027,\n",
" 'restricted_stock_deferred': -126027,\n",
" 'salary': 201955,\n",
" 'shared_receipt_with_poi': 1407,\n",
" 'to_messages': 2902,\n",
" 'total_payments': 4484442,\n",
" 'total_stock_value': 1729541},\n",
" 'BADUM JAMES P': {'bonus': 'NaN',\n",
" 'deferral_payments': 178980,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 257817,\n",
" 'expenses': 3486,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 182466,\n",
" 'total_stock_value': 257817},\n",
" 'BANNANTINE JAMES M': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -5104,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'james.bannantine@enron.com',\n",
" 'exercised_stock_options': 4046157,\n",
" 'expenses': 56301,\n",
" 'from_messages': 29,\n",
" 'from_poi_to_this_person': 39,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 864523,\n",
" 'poi': False,\n",
" 'restricted_stock': 1757552,\n",
" 'restricted_stock_deferred': -560222,\n",
" 'salary': 477,\n",
" 'shared_receipt_with_poi': 465,\n",
" 'to_messages': 566,\n",
" 'total_payments': 916197,\n",
" 'total_stock_value': 5243487},\n",
" 'BAXTER JOHN C': {'bonus': 1200000,\n",
" 'deferral_payments': 1295738,\n",
" 'deferred_income': -1386055,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 6680544,\n",
" 'expenses': 11200,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 1586055,\n",
" 'other': 2660303,\n",
" 'poi': False,\n",
" 'restricted_stock': 3942714,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 267102,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 5634343,\n",
" 'total_stock_value': 10623258},\n",
" 'BAY FRANKLIN R': {'bonus': 400000,\n",
" 'deferral_payments': 260455,\n",

"   'deferred_income': -201641,\n",
"   'director_fees': 'NaN',\n",
"   'email_address': 'frank.bay@enron.com',\n",
"   'exercised_stock_options': 'NaN',\n",
"   'expenses': 129142,\n",
"   'from_messages': 'NaN',\n",
"   'from_poi_to_this_person': 'NaN',\n",
"   'from_this_person_to_poi': 'NaN',\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 'NaN',\n",
"   'other': 69,\n",
"   'poi': False,\n",
"   'restricted_stock': 145796,\n",
"   'restricted_stock_deferred': -82782,\n",
"   'salary': 239671,\n",
"   'shared_receipt_with_poi': 'NaN',\n",
"   'to_messages': 'NaN',\n",
"   'total_payments': 827696,\n",
"   'total_stock_value': 63014},\n",
"   'BAZELIDES PHILIP J': {'bonus': 'NaN',\n",
"   'deferral_payments': 684694,\n",
"   'deferred_income': 'NaN',\n",
"   'director_fees': 'NaN',\n",
"   'email_address': 'NaN',\n",
"   'exercised_stock_options': 1599641,\n",
"   'expenses': 'NaN',\n",
"   'from_messages': 'NaN',\n",
"   'from_poi_to_this_person': 'NaN',\n",
"   'from_this_person_to_poi': 'NaN',\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 93750,\n",
"   'other': 874,\n",
"   'poi': False,\n",
"   'restricted_stock': 'NaN',\n",
"   'restricted_stock_deferred': 'NaN',\n",
"   'salary': 80818,\n",
"   'shared_receipt_with_poi': 'NaN',\n",
"   'to_messages': 'NaN',\n",
"   'total_payments': 860136,\n",
"   'total_stock_value': 1599641},\n",
"   'BECK SALLY W': {'bonus': 700000,\n",
"   'deferral_payments': 'NaN',\n",
"   'deferred_income': 'NaN',\n",
"   'director_fees': 'NaN',\n",
"   'email_address': 'sally.beck@enron.com',\n",
"   'exercised_stock_options': 'NaN',\n",
"   'expenses': 37172,\n",
"   'from_messages': 4343,\n",
"   'from_poi_to_this_person': 144,\n",
"   'from_this_person_to_poi': 386,\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 'NaN',\n",
"   'other': 566,\n",
"   'poi': False,\n",
"   'restricted_stock': 126027,\n",
"   'restricted_stock_deferred': 'NaN',\n",
"   'salary': 231330,\n",
"   'shared_receipt_with_poi': 2639,\n",
"   'to_messages': 7315,\n",
"   'total_payments': 969068,\n",
"   'total_stock_value': 126027},\n",
"   'BELDEN TIMOTHY N': {'bonus': 5249999,\n",
"   'deferral_payments': 2144013,\n",
"   'deferred_income': -2334434,\n",
"   'director_fees': 'NaN',\n",
"   'email_address': 'tim.belden@enron.com',\n",
"   'exercised_stock_options': 953136,\n",
"   'expenses': 17355,\n",
"   'from_messages': 484,\n",
"   'from_poi_to_this_person': 228,\n",
"   'from_this_person_to_poi': 108,\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 'NaN',\n",
"   'other': 210698,\n",
"   'poi': True,\n",
"   'restricted_stock': 157569,\n",
"   'restricted_stock_deferred': 'NaN',\n",
"   'salary': 213999,\n",
"   'shared_receipt_with_poi': 5521,\n",
"   'to_messages': 7991,\n",
"   'total_payments': 5501630,\n",
"   'total_stock_value': 1110705},\n",
"   'BELFER ROBERT': {'bonus': 'NaN',\n",
"   'deferral_payments': -102500,\n",
"   'deferred_income': 'NaN',\n",
"   'director_fees': 3285,\n",
"   'email_address': 'NaN',\n",
"   'exercised_stock_options': 3285,\n",
"   'expenses': 'NaN',\n",
"   'from_messages': 'NaN',\n",
"   'from_poi_to_this_person': 'NaN',\n",
"   'from_this_person_to_poi': 'NaN',\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 'NaN',\n",
"   'other': 'NaN',\n",
"   'poi': False,\n",
"   'restricted_stock': 'NaN',\n",
"   'restricted_stock_deferred': 44093,\n",
"   'salary': 'NaN',\n",
"   'shared_receipt_with_poi': 'NaN',\n",
"   'to_messages': 'NaN',\n",
"   'total_payments': 102500,\n",
"   'total_stock_value': -44093},\n",

" 'BERBERIAN DAVID': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'david.berberian@enron.com',\n",
" 'exercised_stock_options': 1624396,\n",
" 'expenses': 11892,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 869220,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 216582,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 228474,\n",
" 'total_stock_value': 2493616},\n",
" 'BERGSIEKER RICHARD P': {'bonus': 250000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -485813,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'rick.bergsieker@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 59175,\n",
" 'from_messages': 59,\n",
" 'from_poi_to_this_person': 4,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 180250,\n",
" 'other': 427316,\n",
" 'poi': False,\n",
" 'restricted_stock': 659249,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 187922,\n",
" 'shared_receipt_with_poi': 233,\n",
" 'to_messages': 383,\n",
" 'total_payments': 618850,\n",
" 'total_stock_value': 659249},\n",
" 'BHATNAGAR SANJAY': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 137864,\n",
" 'email_address': 'sanjay.bhatnagar@enron.com',\n",
" 'exercised_stock_options': 2604490,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 29,\n",
" 'from_poi_to_this_person': 0,\n",
" 'from_this_person_to_poi': 1,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 137864,\n",
" 'poi': False,\n",
" 'restricted_stock': -2604490,\n",
" 'restricted_stock_deferred': 15456290,\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 463,\n",
" 'to_messages': 523,\n",
" 'total_payments': 15456290,\n",
" 'total_stock_value': 'NaN'},\n",
" 'BIBI PHILIPPE A': {'bonus': 1000000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'philippe.bibi@enron.com',\n",
" 'exercised_stock_options': 1465734,\n",
" 'expenses': 38559,\n",
" 'from_messages': 40,\n",
" 'from_poi_to_this_person': 23,\n",
" 'from_this_person_to_poi': 8,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 369721,\n",
" 'other': 425688,\n",
" 'poi': False,\n",
" 'restricted_stock': 378082,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 213625,\n",
" 'shared_receipt_with_poi': 1336,\n",
" 'to_messages': 1607,\n",
" 'total_payments': 2047593,\n",
" 'total_stock_value': 1843816},\n",
" 'BLACHMAN JEREMY M': {'bonus': 850000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'jeremy.blachman@enron.com',\n",
" 'exercised_stock_options': 765313,\n",
" 'expenses': 84208,\n",
" 'from_messages': 14,\n",
" 'from_poi_to_this_person': 25,\n",
" 'from_this_person_to_poi': 2,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 831809,\n",
" 'other': 272,\n",
" 'poi': False,\n",
" 'restricted_stock': 189041,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 248546,\n",
" 'shared_receipt_with_poi': 2326,\n",
" 'to_messages': 2475,\n",

"   'total_payments': 2014835,\n",
"   'total_stock_value': 954354},\n",
" 'BLAKE JR. NORMAN P': {'bonus': 'NaN',\n",
"   'deferral_payments': 'NaN',\n",
"   'deferred_income': -113784,\n",
"   'director_fees': 113784,\n",
"   'email_address': 'NaN',\n",
"   'exercised_stock_options': 'NaN',\n",
"   'expenses': 1279,\n",
"   'from_messages': 'NaN',\n",
"   'from_poi_to_this_person': 'NaN',\n",
"   'from_this_person_to_poi': 'NaN',\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 'NaN',\n",
"   'other': 'NaN',\n",
"   'poi': False,\n",
"   'restricted_stock': 'NaN',\n",
"   'restricted_stock_deferred': 'NaN',\n",
"   'salary': 'NaN',\n",
"   'shared_receipt_with_poi': 'NaN',\n",
"   'to_messages': 'NaN',\n",
"   'total_payments': 1279,\n",
"   'total_stock_value': 'NaN'},\n",
" 'BOWEN JR RAYMOND M': {'bonus': 1350000,\n",
"   'deferral_payments': 'NaN',\n",
"   'deferred_income': -833,\n",
"   'director_fees': 'NaN',\n",
"   'email_address': 'raymond.bowen@enron.com',\n",
"   'exercised_stock_options': 'NaN',\n",
"   'expenses': 65907,\n",
"   'from_messages': 27,\n",
"   'from_poi_to_this_person': 140,\n",
"   'from_this_person_to_poi': 15,\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 974293,\n",
"   'other': 1621,\n",
"   'poi': True,\n",
"   'restricted_stock': 252055,\n",
"   'restricted_stock_deferred': 'NaN',\n",
"   'salary': 278601,\n",
"   'shared_receipt_with_poi': 1593,\n",
"   'to_messages': 1858,\n",
"   'total_payments': 2669589,\n",
"   'total_stock_value': 252055},\n",
" 'BROWN MICHAEL': {'bonus': 'NaN',\n",
"   'deferral_payments': 'NaN',\n",
"   'deferred_income': 'NaN',\n",
"   'director_fees': 'NaN',\n",
"   'email_address': 'michael.brown@enron.com',\n",
"   'exercised_stock_options': 'NaN',\n",
"   'expenses': 49288,\n",
"   'from_messages': 41,\n",
"   'from_poi_to_this_person': 13,\n",
"   'from_this_person_to_poi': 1,\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 'NaN',\n",
"   'other': 'NaN',\n",
"   'poi': False,\n",
"   'restricted_stock': 'NaN',\n",
"   'restricted_stock_deferred': 'NaN',\n",
"   'salary': 'NaN',\n",
"   'shared_receipt_with_poi': 761,\n",
"   'to_messages': 1486,\n",
"   'total_payments': 49288,\n",
"   'total_stock_value': 'NaN'},\n",
" 'BUCHANAN HAROLD G': {'bonus': 500000,\n",
"   'deferral_payments': 'NaN',\n",
"   'deferred_income': 'NaN',\n",
"   'director_fees': 'NaN',\n",
"   'email_address': 'john.buchanan@enron.com',\n",
"   'exercised_stock_options': 825464,\n",
"   'expenses': 600,\n",
"   'from_messages': 125,\n",
"   'from_poi_to_this_person': 0,\n",
"   'from_this_person_to_poi': 0,\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 304805,\n",
"   'other': 1215,\n",
"   'poi': False,\n",
"   'restricted_stock': 189041,\n",
"   'restricted_stock_deferred': 'NaN',\n",
"   'salary': 248017,\n",
"   'shared_receipt_with_poi': 23,\n",
"   'to_messages': 1088,\n",
"   'total_payments': 1054637,\n",
"   'total_stock_value': 1014505},\n",
" 'BUTTS ROBERT H': {'bonus': 750000,\n",
"   'deferral_payments': 'NaN',\n",
"   'deferred_income': -75000,\n",
"   'director_fees': 'NaN',\n",
"   'email_address': 'bob.butts@enron.com',\n",
"   'exercised_stock_options': 'NaN',\n",
"   'expenses': 9410,\n",
"   'from_messages': 'NaN',\n",
"   'from_poi_to_this_person': 'NaN',\n",
"   'from_this_person_to_poi': 'NaN',\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 175000,\n",
"   'other': 150656,\n",
"   'poi': False,\n",
"   'restricted_stock': 417619,\n",
"   'restricted_stock_deferred': 'NaN',\n",
"   'salary': 261516,\n",

" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 1271582,\n",
" 'total_stock_value': 417619},\n",
" 'BUY RICHARD B': {'bonus': 900000,\n",
" 'deferral_payments': 649584,\n",
" 'deferred_income': -694862,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'rick.buy@enron.com',\n",
" 'exercised_stock_options': 2542813,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 1053,\n",
" 'from_poi_to_this_person': 156,\n",
" 'from_this_person_to_poi': 71,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 769862,\n",
" 'other': 400572,\n",
" 'poi': False,\n",
" 'restricted_stock': 901657,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 330546,\n",
" 'shared_receipt_with_poi': 2333,\n",
" 'to_messages': 3523,\n",
" 'total_payments': 2355702,\n",
" 'total_stock_value': 3444470},\n",
" 'CALGER CHRISTOPHER F': {'bonus': 1250000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -262500,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'christopher.calger@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 35818,\n",
" 'from_messages': 144,\n",
" 'from_poi_to_this_person': 199,\n",
" 'from_this_person_to_poi': 25,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 375304,\n",
" 'other': 486,\n",
" 'poi': True,\n",
" 'restricted_stock': 126027,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 240189,\n",
" 'shared_receipt_with_poi': 2188,\n",
" 'to_messages': 2598,\n",
" 'total_payments': 1639297,\n",
" 'total_stock_value': 126027},\n",
" 'CARTER REBECCA C': {'bonus': 300000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -159792,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'rebecca.carter@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 15,\n",
" 'from_poi_to_this_person': 29,\n",
" 'from_this_person_to_poi': 7,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 75000,\n",
" 'other': 540,\n",
" 'poi': False,\n",
" 'restricted_stock': 307301,\n",
" 'restricted_stock_deferred': -307301,\n",
" 'salary': 261809,\n",
" 'shared_receipt_with_poi': 196,\n",
" 'to_messages': 312,\n",
" 'total_payments': 477557,\n",
" 'total_stock_value': 'NaN'},\n",
" 'CAUSEY RICHARD A': {'bonus': 1000000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -235000,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'richard.causey@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 30674,\n",
" 'from_messages': 49,\n",
" 'from_poi_to_this_person': 58,\n",
" 'from_this_person_to_poi': 12,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 350000,\n",
" 'other': 307895,\n",
" 'poi': True,\n",
" 'restricted_stock': 2502063,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 415189,\n",
" 'shared_receipt_with_poi': 1585,\n",
" 'to_messages': 1892,\n",
" 'total_payments': 1868758,\n",
" 'total_stock_value': 2502063},\n",
" 'CHAN RONNIE': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -98784,\n",
" 'director_fees': 98784,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 32460,\n",

" 'restricted_stock_deferred': -32460,\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 'NaN'},\n",
" 'CHRISTODOULOU DIOMEDES': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'diomedes.christodoulou@enron.com',\n",
" 'exercised_stock_options': 5127155,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 950730,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 6077885},\n",
" 'CLINE KENNETH W': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 662086,\n",
" 'restricted_stock_deferred': -472568,\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 189518},\n",
" 'COLWELL WESLEY': {'bonus': 1200000,\n",
" 'deferral_payments': 27610,\n",
" 'deferred_income': -144062,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'wes.colwell@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 16514,\n",
" 'from_messages': 40,\n",
" 'from_poi_to_this_person': 240,\n",
" 'from_this_person_to_poi': 11,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 101740,\n",
" 'poi': True,\n",
" 'restricted_stock': 698242,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 288542,\n",
" 'shared_receipt_with_poi': 1132,\n",
" 'to_messages': 1758,\n",
" 'total_payments': 1490344,\n",
" 'total_stock_value': 698242},\n",
" 'CORDES WILLIAM R': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'bill.cordes@enron.com',\n",
" 'exercised_stock_options': 651850,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 12,\n",
" 'from_poi_to_this_person': 10,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 386335,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 58,\n",
" 'to_messages': 764,\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 1038185},\n",
" 'COX DAVID': {'bonus': 800000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -41250,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'chip.cox@enron.com',\n",
" 'exercised_stock_options': 117551,\n",
" 'expenses': 27861,\n",
" 'from_messages': 33,\n",
" 'from_poi_to_this_person': 0,\n",
" 'from_this_person_to_poi': 4,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 494,\n",

" 'poi': False,\n",
" 'restricted_stock': 378082,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 314288,\n",
" 'shared_receipt_with_poi': 71,\n",
" 'to_messages': 102,\n",
" 'total_payments': 1101393,\n",
" 'total_stock_value': 495633},\n",
" 'CUMBERLAND MICHAEL S': {'bonus': 325000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 22344,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 275000,\n",
" 'other': 713,\n",
" 'poi': False,\n",
" 'restricted_stock': 207940,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 184899,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 807956,\n",
" 'total_stock_value': 207940},\n",
" 'DEFFNER JOSEPH M': {'bonus': 600000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'joseph.deffner@enron.com',\n",
" 'exercised_stock_options': 17378,\n",
" 'expenses': 41626,\n",
" 'from_messages': 74,\n",
" 'from_poi_to_this_person': 115,\n",
" 'from_this_person_to_poi': 4,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 335349,\n",
" 'other': 25553,\n",
" 'poi': False,\n",
" 'restricted_stock': 141833,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 206121,\n",
" 'shared_receipt_with_poi': 552,\n",
" 'to_messages': 714,\n",
" 'total_payments': 1208649,\n",
" 'total_stock_value': 159211},\n",
" 'DELAINEY DAVID W': {'bonus': 3000000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'david.delainey@enron.com',\n",
" 'exercised_stock_options': 2291113,\n",
" 'expenses': 86174,\n",
" 'from_messages': 3069,\n",
" 'from_poi_to_this_person': 66,\n",
" 'from_this_person_to_poi': 609,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 1294981,\n",
" 'other': 1661,\n",
" 'poi': True,\n",
" 'restricted_stock': 1323148,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 365163,\n",
" 'shared_receipt_with_poi': 2097,\n",
" 'to_messages': 3093,\n",
" 'total_payments': 4747979,\n",
" 'total_stock_value': 3614261},\n",
" 'DERRICK JR. JAMES V': {'bonus': 800000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -1284000,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'james.derrick@enron.com',\n",
" 'exercised_stock_options': 8831913,\n",
" 'expenses': 51124,\n",
" 'from_messages': 909,\n",
" 'from_poi_to_this_person': 64,\n",
" 'from_this_person_to_poi': 20,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 484000,\n",
" 'other': 7482,\n",
" 'poi': False,\n",
" 'restricted_stock': 1787380,\n",
" 'restricted_stock_deferred': -1787380,\n",
" 'salary': 492375,\n",
" 'shared_receipt_with_poi': 1401,\n",
" 'to_messages': 2181,\n",
" 'total_payments': 550981,\n",
" 'total_stock_value': 8831913},\n",
" 'DETMERING TIMOTHY J': {'bonus': 425000,\n",
" 'deferral_payments': 875307,\n",
" 'deferred_income': -775241,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'timothy.detmering@enron.com',\n",
" 'exercised_stock_options': 2027865,\n",
" 'expenses': 52255,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",

" 'long_term_incentive': 415657,\n",
" 'other': 1105,\n",
" 'poi': False,\n",
" 'restricted_stock': 315068,\n",
" 'restricted_stock_deferred': -315068,\n",
" 'salary': 210500,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 1204583,\n",
" 'total_stock_value': 2027865},\n",
" 'DIETRICH JANET R': {'bonus': 600000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'janet.dietrich@enron.com',\n",
" 'exercised_stock_options': 1550019,\n",
" 'expenses': 3475,\n",
" 'from_messages': 63,\n",
" 'from_poi_to_this_person': 305,\n",
" 'from_this_person_to_poi': 14,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 556416,\n",
" 'other': 473,\n",
" 'poi': False,\n",
" 'restricted_stock': 315068,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 250100,\n",
" 'shared_receipt_with_poi': 1902,\n",
" 'to_messages': 2572,\n",
" 'total_payments': 1410464,\n",
" 'total_stock_value': 1865087},\n",
" 'DIMICHELE RICHARD G': {'bonus': 1000000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'richard.dimichele@enron.com',\n",
" 'exercised_stock_options': 8191755,\n",
" 'expenses': 35812,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 694862,\n",
" 'other': 374689,\n",
" 'poi': False,\n",
" 'restricted_stock': 126027,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 262788,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 2368151,\n",
" 'total_stock_value': 8317782},\n",
" 'DODSON KEITH': {'bonus': 70000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'keith.dodson@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 28164,\n",
" 'from_messages': 14,\n",
" 'from_poi_to_this_person': 10,\n",
" 'from_this_person_to_poi': 3,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 774,\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 221003,\n",
" 'shared_receipt_with_poi': 114,\n",
" 'to_messages': 176,\n",
" 'total_payments': 319941,\n",
" 'total_stock_value': 'NaN'},\n",
" 'DONAHUE JR JEFFREY M': {'bonus': 800000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -300000,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'jeff.donahue@enron.com',\n",
" 'exercised_stock_options': 765920,\n",
" 'expenses': 96268,\n",
" 'from_messages': 22,\n",
" 'from_poi_to_this_person': 188,\n",
" 'from_this_person_to_poi': 11,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 891,\n",
" 'poi': False,\n",
" 'restricted_stock': 315068,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 278601,\n",
" 'shared_receipt_with_poi': 772,\n",
" 'to_messages': 865,\n",
" 'total_payments': 875760,\n",
" 'total_stock_value': 1080988},\n",
" 'DUNCAN JOHN H': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -25000,\n",
" 'director_fees': 102492,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 371750,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",

" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 77492,\n",
" 'total_stock_value': 371750},\n",
" 'DURAN WILLIAM D': {'bonus': 750000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'w.duran@enron.com',\n",
" 'exercised_stock_options': 1451869,\n",
" 'expenses': 25785,\n",
" 'from_messages': 12,\n",
" 'from_poi_to_this_person': 106,\n",
" 'from_this_person_to_poi': 3,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 1105218,\n",
" 'other': 1568,\n",
" 'poi': False,\n",
" 'restricted_stock': 189041,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 210692,\n",
" 'shared_receipt_with_poi': 599,\n",
" 'to_messages': 904,\n",
" 'total_payments': 2093263,\n",
" 'total_stock_value': 1640910},\n",
" 'ECHOLS JOHN B': {'bonus': 200000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'john.echols@enron.com',\n",
" 'exercised_stock_options': 601438,\n",
" 'expenses': 21530,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 2234774,\n",
" 'other': 53775,\n",
" 'poi': False,\n",
" 'restricted_stock': 407503,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 182245,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 2692324,\n",
" 'total_stock_value': 1008941},\n",
" 'ELLIOTT STEVEN': {'bonus': 350000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -400729,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'steven.elliott@enron.com',\n",
" 'exercised_stock_options': 4890344,\n",
" 'expenses': 78552,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 12961,\n",
" 'poi': False,\n",
" 'restricted_stock': 1788391,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 170941,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 211725,\n",
" 'total_stock_value': 6678735},\n",
" 'FALLON JAMES B': {'bonus': 2500000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'jim.fallon@enron.com',\n",
" 'exercised_stock_options': 940257,\n",
" 'expenses': 95924,\n",
" 'from_messages': 75,\n",
" 'from_poi_to_this_person': 42,\n",
" 'from_this_person_to_poi': 37,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 374347,\n",
" 'other': 401481,\n",
" 'poi': False,\n",
" 'restricted_stock': 1392142,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 304588,\n",
" 'shared_receipt_with_poi': 1604,\n",
" 'to_messages': 1755,\n",
" 'total_payments': 3676340,\n",
" 'total_stock_value': 2332399},\n",
" 'FASTOW ANDREW S': {'bonus': 1300000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -1386055,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'andrew.fastow@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 55921,\n",

" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 1736055,\n",
" 'other': 277464,\n",
" 'poi': True,\n",
" 'restricted_stock': 1794412,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 440698,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 2424083,\n",
" 'total_stock_value': 1794412},\n",
" 'FITZGERALD JAY L': {'bonus': 350000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'jay.fitzgerald@enron.com',\n",
" 'exercised_stock_options': 664461,\n",
" 'expenses': 23870,\n",
" 'from_messages': 16,\n",
" 'from_poi_to_this_person': 1,\n",
" 'from_this_person_to_poi': 8,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 556416,\n",
" 'other': 285414,\n",
" 'poi': False,\n",
" 'restricted_stock': 956775,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 199157,\n",
" 'shared_receipt_with_poi': 723,\n",
" 'to_messages': 936,\n",
" 'total_payments': 1414857,\n",
" 'total_stock_value': 1621236},\n",
" 'FOWLER PEGGY': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'kulvinder.fowler@enron.com',\n",
" 'exercised_stock_options': 1324578,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 36,\n",
" 'from_poi_to_this_person': 0,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 560170,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 10,\n",
" 'to_messages': 517,\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 1884748},\n",
" 'FOY JOE': {'bonus': 'NaN',\n",
" 'deferral_payments': 181755,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'tracy.foy@enron.com',\n",
" 'exercised_stock_options': 343434,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 13,\n",
" 'from_poi_to_this_person': 0,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 2,\n",
" 'to_messages': 57,\n",
" 'total_payments': 181755,\n",
" 'total_stock_value': 343434},\n",
" 'FREVERT MARK A': {'bonus': 2000000,\n",
" 'deferral_payments': 6426990,\n",
" 'deferred_income': -3367011,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'mark.frevert@enron.com',\n",
" 'exercised_stock_options': 10433518,\n",
" 'expenses': 86987,\n",
" 'from_messages': 21,\n",
" 'from_poi_to_this_person': 242,\n",
" 'from_this_person_to_poi': 6,\n",
" 'loan_advances': 2000000,\n",
" 'long_term_incentive': 1617011,\n",
" 'other': 7427621,\n",
" 'poi': False,\n",
" 'restricted_stock': 4188667,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 1060932,\n",
" 'shared_receipt_with_poi': 2979,\n",
" 'to_messages': 3275,\n",
" 'total_payments': 17252530,\n",
" 'total_stock_value': 14622185},\n",
" 'FUGH JOHN L': {'bonus': 'NaN',\n",
" 'deferral_payments': 50591,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",

" 'exercised_stock_options': 176378,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 50591,\n",
" 'total_stock_value': 176378},\n",
" 'GAHN ROBERT S': {'bonus': 509870,\n",
" 'deferral_payments': 73122,\n",
" 'deferred_income': -1042,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 83237,\n",
" 'expenses': 50080,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 76547,\n",
" 'poi': False,\n",
" 'restricted_stock': 235370,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 192008,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 900585,\n",
" 'total_stock_value': 318607},\n",
" 'GARLAND C KEVIN': {'bonus': 850000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'kevin.garland@enron.com',\n",
" 'exercised_stock_options': 636246,\n",
" 'expenses': 48405,\n",
" 'from_messages': 44,\n",
" 'from_poi_to_this_person': 10,\n",
" 'from_this_person_to_poi': 27,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 375304,\n",
" 'other': 60814,\n",
" 'poi': False,\n",
" 'restricted_stock': 259907,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 231946,\n",
" 'shared_receipt_with_poi': 178,\n",
" 'to_messages': 209,\n",
" 'total_payments': 1566469,\n",
" 'total_stock_value': 896153},\n",
" 'GATHMANN WILLIAM D': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 1753766,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 264013,\n",
" 'restricted_stock_deferred': -72419,\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 1945360},\n",
" 'GIBBS DANA R': {'bonus': 'NaN',\n",
" 'deferral_payments': 504610,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'dana.gibbs@enron.com',\n",
" 'exercised_stock_options': 2218275,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 12,\n",
" 'from_poi_to_this_person': 0,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 461912,\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 23,\n",
" 'to_messages': 169,\n",
" 'total_payments': 966522,\n",
" 'total_stock_value': 2218275},\n",
" 'GILLIS JOHN': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",

" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 9803,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 75838,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 85641},\n",
" 'GLISAN JR BEN F': {'bonus': 600000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'ben.glisan@enron.com',\n",
" 'exercised_stock_options': 384728,\n",
" 'expenses': 125978,\n",
" 'from_messages': 16,\n",
" 'from_poi_to_this_person': 52,\n",
" 'from_this_person_to_poi': 6,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 71023,\n",
" 'other': 200308,\n",
" 'poi': True,\n",
" 'restricted_stock': 393818,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 274975,\n",
" 'shared_receipt_with_poi': 874,\n",
" 'to_messages': 873,\n",
" 'total_payments': 1272284,\n",
" 'total_stock_value': 778546},\n",
" 'GOLD JOSEPH': {'bonus': 750000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'joe.gold@enron.com',\n",
" 'exercised_stock_options': 436515,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 304805,\n",
" 'other': 819288,\n",
" 'poi': False,\n",
" 'restricted_stock': 441096,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 272880,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 2146973,\n",
" 'total_stock_value': 877611},\n",
" 'GRAMM WENDY L': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 119292,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 119292,\n",
" 'total_stock_value': 'NaN'},\n",
" 'GRAY RODNEY': {'bonus': 'NaN',\n",
" 'deferral_payments': 93585,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 365625,\n",
" 'other': 680833,\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 6615,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 1146658,\n",
" 'total_stock_value': 'NaN'},\n",
" 'HAEDICKE MARK E': {'bonus': 1150000,\n",

" 'deferral_payments': 2157527,\n",
" 'deferred_income': -934484,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'mark.haedicke@enron.com',\n",
" 'exercised_stock_options': 608750,\n",
" 'expenses': 76169,\n",
" 'from_messages': 1941,\n",
" 'from_poi_to_this_person': 180,\n",
" 'from_this_person_to_poi': 61,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 983346,\n",
" 'other': 52382,\n",
" 'poi': False,\n",
" 'restricted_stock': 524169,\n",
" 'restricted_stock_deferred': -329825,\n",
" 'salary': 374125,\n",
" 'shared_receipt_with_poi': 1847,\n",
" 'to_messages': 4009,\n",
" 'total_payments': 3859065,\n",
" 'total_stock_value': 803094},\n",
" 'HANNON KEVIN P': {'bonus': 1500000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -3117011,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'kevin.hannon@enron.com',\n",
" 'exercised_stock_options': 5538001,\n",
" 'expenses': 34039,\n",
" 'from_messages': 32,\n",
" 'from_poi_to_this_person': 32,\n",
" 'from_this_person_to_poi': 21,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 1617011,\n",
" 'other': 11350,\n",
" 'poi': True,\n",
" 'restricted_stock': 853064,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 243293,\n",
" 'shared_receipt_with_poi': 1035,\n",
" 'to_messages': 1045,\n",
" 'total_payments': 288682,\n",
" 'total_stock_value': 6391065},\n",
" 'HAUG DAVID L': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'david.haug@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 475,\n",
" 'from_messages': 19,\n",
" 'from_poi_to_this_person': 4,\n",
" 'from_this_person_to_poi': 7,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 2217299,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 471,\n",
" 'to_messages': 573,\n",
" 'total_payments': 475,\n",
" 'total_stock_value': 2217299},\n",
" 'HAYES ROBERT E': {'bonus': 'NaN',\n",
" 'deferral_payments': 7961,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'robert.hayes@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 12,\n",
" 'from_poi_to_this_person': 16,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 151418,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 50,\n",
" 'to_messages': 504,\n",
" 'total_payments': 7961,\n",
" 'total_stock_value': 151418},\n",
" 'HAYSLETT RODERICK J': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'rod.hayslett@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 1061,\n",
" 'from_poi_to_this_person': 35,\n",
" 'from_this_person_to_poi': 38,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 346663,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 571,\n",
" 'to_messages': 2649,\n",
" 'total_payments': 'NaN',\n",

" 'total_stock_value': 346663},\n",
" 'HERMANN ROBERT J': {'bonus': 700000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -280000,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'robert.hermann@enron.com',\n",
" 'exercised_stock_options': 187500,\n",
" 'expenses': 48357,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 150000,\n",
" 'other': 416441,\n",
" 'poi': False,\n",
" 'restricted_stock': 480632,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 262663,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 1297461,\n",
" 'total_stock_value': 668132},\n",
" 'HICKERSON GARY J': {'bonus': 1700000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'gary.hickerson@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 98849,\n",
" 'from_messages': 27,\n",
" 'from_poi_to_this_person': 40,\n",
" 'from_this_person_to_poi': 1,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 69223,\n",
" 'other': 1936,\n",
" 'poi': False,\n",
" 'restricted_stock': 441096,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 211788,\n",
" 'shared_receipt_with_poi': 900,\n",
" 'to_messages': 1320,\n",
" 'total_payments': 2081796,\n",
" 'total_stock_value': 441096},\n",
" 'HIRKO JOSEPH': {'bonus': 'NaN',\n",
" 'deferral_payments': 10259,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'joe.hirko@enron.com',\n",
" 'exercised_stock_options': 30766064,\n",
" 'expenses': 77978,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 2856,\n",
" 'poi': True,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 91093,\n",
" 'total_stock_value': 30766064},\n",
" 'HORTON STANLEY C': {'bonus': 'NaN',\n",
" 'deferral_payments': 3131860,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'stanley.horton@enron.com',\n",
" 'exercised_stock_options': 5210569,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 1073,\n",
" 'from_poi_to_this_person': 44,\n",
" 'from_this_person_to_poi': 15,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 2046079,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 1074,\n",
" 'to_messages': 2350,\n",
" 'total_payments': 3131860,\n",
" 'total_stock_value': 7256648},\n",
" 'HUGHES JAMES A': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'james.hughes@enron.com',\n",
" 'exercised_stock_options': 754966,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 34,\n",
" 'from_poi_to_this_person': 35,\n",
" 'from_this_person_to_poi': 5,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 363428,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 589,\n",

" 'to_messages': 719,\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 1118394},\n",
" 'HUMPHREY GENE E': {'bonus': 'NaN',\n",
" 'deferral_payments': 2964506,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'gene.humphrey@enron.com',\n",
" 'exercised_stock_options': 2282768,\n",
" 'expenses': 4994,\n",
" 'from_messages': 17,\n",
" 'from_poi_to_this_person': 10,\n",
" 'from_this_person_to_poi': 17,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 130724,\n",
" 'shared_receipt_with_poi': 119,\n",
" 'to_messages': 128,\n",
" 'total_payments': 3100224,\n",
" 'total_stock_value': 2282768},\n",
" 'IZZO LAWRENCE L': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'larry.izzo@enron.com',\n",
" 'exercised_stock_options': 2165172,\n",
" 'expenses': 28093,\n",
" 'from_messages': 19,\n",
" 'from_poi_to_this_person': 28,\n",
" 'from_this_person_to_poi': 5,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 312500,\n",
" 'other': 1553729,\n",
" 'poi': False,\n",
" 'restricted_stock': 3654808,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 85274,\n",
" 'shared_receipt_with_poi': 437,\n",
" 'to_messages': 496,\n",
" 'total_payments': 1979596,\n",
" 'total_stock_value': 5819980},\n",
" 'JACKSON CHARLENE R': {'bonus': 250000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'charlene.jackson@enron.com',\n",
" 'exercised_stock_options': 185063,\n",
" 'expenses': 10181,\n",
" 'from_messages': 56,\n",
" 'from_poi_to_this_person': 25,\n",
" 'from_this_person_to_poi': 19,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 2435,\n",
" 'poi': False,\n",
" 'restricted_stock': 540672,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 288558,\n",
" 'shared_receipt_with_poi': 117,\n",
" 'to_messages': 258,\n",
" 'total_payments': 551174,\n",
" 'total_stock_value': 725735},\n",
" 'JAEDICKE ROBERT': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -25000,\n",
" 'director_fees': 108750,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 431750,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 44093,\n",
" 'restricted_stock_deferred': -44093,\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 83750,\n",
" 'total_stock_value': 431750},\n",
" 'KAMINSKI WINCENTY J': {'bonus': 400000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'vince.kaminski@enron.com',\n",
" 'exercised_stock_options': 850010,\n",
" 'expenses': 83585,\n",
" 'from_messages': 14368,\n",
" 'from_poi_to_this_person': 41,\n",
" 'from_this_person_to_poi': 171,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 323466,\n",
" 'other': 4669,\n",
" 'poi': False,\n",
" 'restricted_stock': 126027,\n",
" 'restricted_stock_deferred': 'NaN',\n",

" 'salary': 275101,\n",
" 'shared_receipt_with_poi': 583,\n",
" 'to_messages': 4607,\n",
" 'total_payments': 1086821,\n",
" 'total_stock_value': 976037},\n",
" 'KEAN STEVEN J': {'bonus': 1000000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'steven.kean@enron.com',\n",
" 'exercised_stock_options': 2022048,\n",
" 'expenses': 41953,\n",
" 'from_messages': 6759,\n",
" 'from_poi_to_this_person': 140,\n",
" 'from_this_person_to_poi': 387,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 300000,\n",
" 'other': 1231,\n",
" 'poi': False,\n",
" 'restricted_stock': 4131594,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 404338,\n",
" 'shared_receipt_with_poi': 3639,\n",
" 'to_messages': 12754,\n",
" 'total_payments': 1747522,\n",
" 'total_stock_value': 6153642},\n",
" 'KISHKILL JOSEPH G': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -51042,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'joe.kishkill@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 116335,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 465357,\n",
" 'poi': False,\n",
" 'restricted_stock': 1034346,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 174246,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 704896,\n",
" 'total_stock_value': 1034346},\n",
" 'KITCHEN LOUISE': {'bonus': 3100000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'louise.kitchen@enron.com',\n",
" 'exercised_stock_options': 81042,\n",
" 'expenses': 5774,\n",
" 'from_messages': 1728,\n",
" 'from_poi_to_this_person': 251,\n",
" 'from_this_person_to_poi': 194,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 93925,\n",
" 'poi': False,\n",
" 'restricted_stock': 466101,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 271442,\n",
" 'shared_receipt_with_poi': 3669,\n",
" 'to_messages': 8305,\n",
" 'total_payments': 3471141,\n",
" 'total_stock_value': 547143},\n",
" 'KOENIG MARK E': {'bonus': 700000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'mark.koenig@enron.com',\n",
" 'exercised_stock_options': 671737,\n",
" 'expenses': 127017,\n",
" 'from_messages': 61,\n",
" 'from_poi_to_this_person': 53,\n",
" 'from_this_person_to_poi': 15,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 300000,\n",
" 'other': 150458,\n",
" 'poi': True,\n",
" 'restricted_stock': 1248318,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 309946,\n",
" 'shared_receipt_with_poi': 2271,\n",
" 'to_messages': 2374,\n",
" 'total_payments': 1587421,\n",
" 'total_stock_value': 1920055},\n",
" 'KOPPER MICHAEL J': {'bonus': 800000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'michael.kopper@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 118134,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 602671,\n",
" 'other': 907502,\n",
" 'poi': True,\n",

" 'restricted_stock': 985032,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 224305,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 2652612,\n",
" 'total_stock_value': 985032},\n",
" 'LAVORATO JOHN J': {'bonus': 8000000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'john.lavorato@enron.com',\n",
" 'exercised_stock_options': 4158995,\n",
" 'expenses': 49537,\n",
" 'from_messages': 2585,\n",
" 'from_poi_to_this_person': 528,\n",
" 'from_this_person_to_poi': 411,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 2035380,\n",
" 'other': 1552,\n",
" 'poi': False,\n",
" 'restricted_stock': 1008149,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 339288,\n",
" 'shared_receipt_with_poi': 3962,\n",
" 'to_messages': 7259,\n",
" 'total_payments': 10425757,\n",
" 'total_stock_value': 5167144},\n",
" 'LAY KENNETH L': {'bonus': 7000000,\n",
" 'deferral_payments': 202911,\n",
" 'deferred_income': -300000,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'kenneth.lay@enron.com',\n",
" 'exercised_stock_options': 34348384,\n",
" 'expenses': 99832,\n",
" 'from_messages': 36,\n",
" 'from_poi_to_this_person': 123,\n",
" 'from_this_person_to_poi': 16,\n",
" 'loan_advances': 81525000,\n",
" 'long_term_incentive': 3600000,\n",
" 'other': 10359729,\n",
" 'poi': True,\n",
" 'restricted_stock': 14761694,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 1072321,\n",
" 'shared_receipt_with_poi': 2411,\n",
" 'to_messages': 4273,\n",
" 'total_payments': 103559793,\n",
" 'total_stock_value': 49110078},\n",
" 'LEFF DANIEL P': {'bonus': 1000000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'dan.leff@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 63,\n",
" 'from_poi_to_this_person': 67,\n",
" 'from_this_person_to_poi': 14,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 1387399,\n",
" 'other': 3083,\n",
" 'poi': False,\n",
" 'restricted_stock': 360528,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 273746,\n",
" 'shared_receipt_with_poi': 2672,\n",
" 'to_messages': 2822,\n",
" 'total_payments': 2664228,\n",
" 'total_stock_value': 360528},\n",
" 'LEMAISTRE CHARLES': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -25000,\n",
" 'director_fees': 112492,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 412878,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 87492,\n",
" 'total_stock_value': 412878},\n",
" 'LEWIS RICHARD': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'richard.lewis@enron.com',\n",
" 'exercised_stock_options': 850477,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 26,\n",
" 'from_poi_to_this_person': 10,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",

" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 739,\n",
" 'to_messages': 952,\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 850477},\n",
" 'LINDHOLM TOD A': {'bonus': 200000,\n",
" 'deferral_payments': 204075,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'tod.lindholm@enron.com',\n",
" 'exercised_stock_options': 2549361,\n",
" 'expenses': 57727,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 175000,\n",
" 'other': 2630,\n",
" 'poi': False,\n",
" 'restricted_stock': 514847,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 236457,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 875889,\n",
" 'total_stock_value': 3064208},\n",
" 'LOCKHART EUGENE E': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 'NaN'},\n",
" 'LOWRY CHARLES P': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 372205,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 153686,\n",
" 'restricted_stock_deferred': -153686,\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 372205},\n",
" 'MARTIN AMANDA K': {'bonus': 'NaN',\n",
" 'deferral_payments': 85430,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'a..martin@enron.com',\n",
" 'exercised_stock_options': 2070306,\n",
" 'expenses': 8211,\n",
" 'from_messages': 230,\n",
" 'from_poi_to_this_person': 8,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 5145434,\n",
" 'other': 2818454,\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 349487,\n",
" 'shared_receipt_with_poi': 477,\n",
" 'to_messages': 1522,\n",
" 'total_payments': 8407016,\n",
" 'total_stock_value': 2070306},\n",
" 'MCCARTY DANNY J': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'danny.mccarty@enron.com',\n",
" 'exercised_stock_options': 664375,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 215,\n",
" 'from_poi_to_this_person': 25,\n",
" 'from_this_person_to_poi': 2,\n",

" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 94556,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 508,\n",
" 'to_messages': 1433,\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 758931},\n",
" 'MCCLELLAN GEORGE': {'bonus': 900000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -125000,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'george.mcclellan@enron.com',\n",
" 'exercised_stock_options': 506765,\n",
" 'expenses': 228763,\n",
" 'from_messages': 49,\n",
" 'from_poi_to_this_person': 52,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 51587,\n",
" 'poi': False,\n",
" 'restricted_stock': 441096,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 263413,\n",
" 'shared_receipt_with_poi': 1469,\n",
" 'to_messages': 1744,\n",
" 'total_payments': 1318763,\n",
" 'total_stock_value': 947861},\n",
" 'MCCONNELL MICHAEL S': {'bonus': 1100000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'mike.mcconnell@enron.com',\n",
" 'exercised_stock_options': 1623010,\n",
" 'expenses': 81364,\n",
" 'from_messages': 2742,\n",
" 'from_poi_to_this_person': 92,\n",
" 'from_this_person_to_poi': 194,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 554422,\n",
" 'other': 540,\n",
" 'poi': False,\n",
" 'restricted_stock': 1478269,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 365038,\n",
" 'shared_receipt_with_poi': 2189,\n",
" 'to_messages': 3329,\n",
" 'total_payments': 2101364,\n",
" 'total_stock_value': 3101279},\n",
" 'MCDONALD REBECCA': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'rebecca.mcdonald@enron.com',\n",
" 'exercised_stock_options': 757301,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 13,\n",
" 'from_poi_to_this_person': 54,\n",
" 'from_this_person_to_poi': 1,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 934065,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 720,\n",
" 'to_messages': 894,\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 1691366},\n",
" 'MCMAHON JEFFREY': {'bonus': 2600000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'jeffrey.mcmahon@enron.com',\n",
" 'exercised_stock_options': 1104054,\n",
" 'expenses': 137108,\n",
" 'from_messages': 48,\n",
" 'from_poi_to_this_person': 58,\n",
" 'from_this_person_to_poi': 26,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 694862,\n",
" 'other': 297353,\n",
" 'poi': False,\n",
" 'restricted_stock': 558801,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 370448,\n",
" 'shared_receipt_with_poi': 2228,\n",
" 'to_messages': 2355,\n",
" 'total_payments': 4099771,\n",
" 'total_stock_value': 1662855},\n",
" 'MENDELSOHN JOHN': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -103750,\n",
" 'director_fees': 103750,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 148,\n",
" 'from_messages': 'NaN',\n",

" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 148,\n",
" 'total_stock_value': 'NaN'},\n",
" 'METTS MARK': {'bonus': 600000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'mark.metts@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 94299,\n",
" 'from_messages': 29,\n",
" 'from_poi_to_this_person': 38,\n",
" 'from_this_person_to_poi': 1,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 1740,\n",
" 'poi': False,\n",
" 'restricted_stock': 585062,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 365788,\n",
" 'shared_receipt_with_poi': 702,\n",
" 'to_messages': 807,\n",
" 'total_payments': 1061827,\n",
" 'total_stock_value': 585062},\n",
" 'MEYER JEROME J': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -38346,\n",
" 'director_fees': 38346,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 2151,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 2151,\n",
" 'total_stock_value': 'NaN'},\n",
" 'MEYER ROCKFORD G': {'bonus': 'NaN',\n",
" 'deferral_payments': 1848227,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'rockford.meyer@enron.com',\n",
" 'exercised_stock_options': 493489,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 28,\n",
" 'from_poi_to_this_person': 0,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 462384,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 22,\n",
" 'to_messages': 232,\n",
" 'total_payments': 1848227,\n",
" 'total_stock_value': 955873},\n",
" 'MORAN MICHAEL P': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'michael.moran@enron.com',\n",
" 'exercised_stock_options': 59539,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 19,\n",
" 'from_poi_to_this_person': 0,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 161602,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 127,\n",
" 'to_messages': 672,\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 221141},\n",
" 'MORDAUNT KRISTINA M': {'bonus': 325000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'kristina.mordaunt@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",

"   'expenses': 35018,\n",
"   'from_messages': 'NaN',\n",
"   'from_poi_to_this_person': 'NaN',\n",
"   'from_this_person_to_poi': 'NaN',\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 'NaN',\n",
"   'other': 1411,\n",
"   'poi': False,\n",
"   'restricted_stock': 208510,\n",
"   'restricted_stock_deferred': 'NaN',\n",
"   'salary': 267093,\n",
"   'shared_receipt_with_poi': 'NaN',\n",
"   'to_messages': 'NaN',\n",
"   'total_payments': 628522,\n",
"   'total_stock_value': 208510},\n",
" 'MULLER MARK S': {'bonus': 1100000,\n",
"   'deferral_payments': 842924,\n",
"   'deferred_income': -719000,\n",
"   'director_fees': 'NaN',\n",
"   'email_address': 's..muller@enron.com',\n",
"   'exercised_stock_options': 1056320,\n",
"   'expenses': 'NaN',\n",
"   'from_messages': 16,\n",
"   'from_poi_to_this_person': 12,\n",
"   'from_this_person_to_poi': 0,\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 1725545,\n",
"   'other': 947,\n",
"   'poi': False,\n",
"   'restricted_stock': 360528,\n",
"   'restricted_stock_deferred': 'NaN',\n",
"   'salary': 251654,\n",
"   'shared_receipt_with_poi': 114,\n",
"   'to_messages': 136,\n",
"   'total_payments': 3202070,\n",
"   'total_stock_value': 1416848},\n",
" 'MURRAY JULIA H': {'bonus': 400000,\n",
"   'deferral_payments': 'NaN',\n",
"   'deferred_income': 'NaN',\n",
"   'director_fees': 'NaN',\n",
"   'email_address': 'julia.murray@enron.com',\n",
"   'exercised_stock_options': 400478,\n",
"   'expenses': 57580,\n",
"   'from_messages': 45,\n",
"   'from_poi_to_this_person': 11,\n",
"   'from_this_person_to_poi': 2,\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 125000,\n",
"   'other': 330,\n",
"   'poi': False,\n",
"   'restricted_stock': 196983,\n",
"   'restricted_stock_deferred': 'NaN',\n",
"   'salary': 229284,\n",
"   'shared_receipt_with_poi': 395,\n",
"   'to_messages': 2192,\n",
"   'total_payments': 812194,\n",
"   'total_stock_value': 597461},\n",
" 'NOLES JAMES L': {'bonus': 'NaN',\n",
"   'deferral_payments': 774401,\n",
"   'deferred_income': 'NaN',\n",
"   'director_fees': 'NaN',\n",
"   'email_address': 'NaN',\n",
"   'exercised_stock_options': 'NaN',\n",
"   'expenses': 'NaN',\n",
"   'from_messages': 'NaN',\n",
"   'from_poi_to_this_person': 'NaN',\n",
"   'from_this_person_to_poi': 'NaN',\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 'NaN',\n",
"   'other': 'NaN',\n",
"   'poi': False,\n",
"   'restricted_stock': 463261,\n",
"   'restricted_stock_deferred': -94556,\n",
"   'salary': 'NaN',\n",
"   'shared_receipt_with_poi': 'NaN',\n",
"   'to_messages': 'NaN',\n",
"   'total_payments': 774401,\n",
"   'total_stock_value': 368705},\n",
" 'OLSON CINDY K': {'bonus': 750000,\n",
"   'deferral_payments': 77716,\n",
"   'deferred_income': 'NaN',\n",
"   'director_fees': 'NaN',\n",
"   'email_address': 'cindy.olson@enron.com',\n",
"   'exercised_stock_options': 1637034,\n",
"   'expenses': 63791,\n",
"   'from_messages': 52,\n",
"   'from_poi_to_this_person': 20,\n",
"   'from_this_person_to_poi': 15,\n",
"   'loan_advances': 'NaN',\n",
"   'long_term_incentive': 100000,\n",
"   'other': 972,\n",
"   'poi': False,\n",
"   'restricted_stock': 969729,\n",
"   'restricted_stock_deferred': 'NaN',\n",
"   'salary': 329078,\n",
"   'shared_receipt_with_poi': 856,\n",
"   'to_messages': 1184,\n",
"   'total_payments': 1321557,\n",
"   'total_stock_value': 2606763},\n",
" 'OVERDYKE JR JERE C': {'bonus': 'NaN',\n",
"   'deferral_payments': 'NaN',\n",
"   'deferred_income': 'NaN',\n",
"   'director_fees': 'NaN',\n",

" 'email_address': 'jere.overdyke@enron.com',\n",
" 'exercised_stock_options': 5266578,\n",
" 'expenses': 18834,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 135836,\n",
" 'other': 176,\n",
" 'poi': False,\n",
" 'restricted_stock': 2041016,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 94941,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 249787,\n",
" 'total_stock_value': 7307594},\n",
" 'PAI LOU L': {'bonus': 1000000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'lou.pai@enron.com',\n",
" 'exercised_stock_options': 15364167,\n",
" 'expenses': 32047,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 1829457,\n",
" 'poi': False,\n",
" 'restricted_stock': 8453763,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 261879,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 3123383,\n",
" 'total_stock_value': 23817930},\n",
" 'PEREIRA PAULO V. FERRAZ': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -101250,\n",
" 'director_fees': 101250,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 27942,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 27942,\n",
" 'total_stock_value': 'NaN'},\n",
" 'PICKERING MARK R': {'bonus': 300000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'mark.pickering@enron.com',\n",
" 'exercised_stock_options': 28798,\n",
" 'expenses': 31653,\n",
" 'from_messages': 67,\n",
" 'from_poi_to_this_person': 7,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 400000,\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 655037,\n",
" 'shared_receipt_with_poi': 728,\n",
" 'to_messages': 898,\n",
" 'total_payments': 1386690,\n",
" 'total_stock_value': 28798},\n",
" 'PIPER GREGORY F': {'bonus': 400000,\n",
" 'deferral_payments': 1130036,\n",
" 'deferred_income': -33333,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'greg.piper@enron.com',\n",
" 'exercised_stock_options': 880290,\n",
" 'expenses': 43057,\n",
" 'from_messages': 222,\n",
" 'from_poi_to_this_person': 61,\n",
" 'from_this_person_to_poi': 48,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 778,\n",
" 'poi': False,\n",
" 'restricted_stock': 409554,\n",
" 'restricted_stock_deferred': -409554,\n",
" 'salary': 197091,\n",
" 'shared_receipt_with_poi': 742,\n",
" 'to_messages': 1238,\n",
" 'total_payments': 1737629,\n",
" 'total_stock_value': 880290},\n",
" 'PIRO JIM': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",

" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'jim.piro@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 16,\n",
" 'from_poi_to_this_person': 0,\n",
" 'from_this_person_to_poi': 1,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 47304,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 3,\n",
" 'to_messages': 58,\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 47304},\n",
" 'POWERS WILLIAM': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -17500,\n",
" 'director_fees': 17500,\n",
" 'email_address': 'ken.powers@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 26,\n",
" 'from_poi_to_this_person': 0,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 12,\n",
" 'to_messages': 653,\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 'NaN'},\n",
" 'PRENTICE JAMES': {'bonus': 'NaN',\n",
" 'deferral_payments': 564348,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'james.prentice@enron.com',\n",
" 'exercised_stock_options': 886231,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 208809,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 564348,\n",
" 'total_stock_value': 1095040},\n",
" 'REDMOND BRIAN L': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'brian.redmond@enron.com',\n",
" 'exercised_stock_options': 7509039,\n",
" 'expenses': 14689,\n",
" 'from_messages': 221,\n",
" 'from_poi_to_this_person': 204,\n",
" 'from_this_person_to_poi': 49,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 381285,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 96840,\n",
" 'shared_receipt_with_poi': 1063,\n",
" 'to_messages': 1671,\n",
" 'total_payments': 111529,\n",
" 'total_stock_value': 7890324},\n",
" 'REYNOLDS LAWRENCE': {'bonus': 100000,\n",
" 'deferral_payments': 51365,\n",
" 'deferred_income': -200000,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 4160672,\n",
" 'expenses': 8409,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 156250,\n",
" 'other': 202052,\n",
" 'poi': False,\n",
" 'restricted_stock': 201483,\n",
" 'restricted_stock_deferred': -140264,\n",
" 'salary': 76399,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 394475,\n",
" 'total_stock_value': 4221891},\n",

" 'RICE KENNETH D': {'bonus': 1750000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -3504386,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'ken.rice@enron.com',\n",
" 'exercised_stock_options': 19794175,\n",
" 'expenses': 46950,\n",
" 'from_messages': 18,\n",
" 'from_poi_to_this_person': 42,\n",
" 'from_this_person_to_poi': 4,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 1617011,\n",
" 'other': 174839,\n",
" 'poi': True,\n",
" 'restricted_stock': 2748364,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 420636,\n",
" 'shared_receipt_with_poi': 864,\n",
" 'to_messages': 905,\n",
" 'total_payments': 505050,\n",
" 'total_stock_value': 22542539},\n",
" 'RIEKER PAULA H': {'bonus': 700000,\n",
" 'deferral_payments': 214678,\n",
" 'deferred_income': -100000,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'paula.rieker@enron.com',\n",
" 'exercised_stock_options': 1635238,\n",
" 'expenses': 33271,\n",
" 'from_messages': 82,\n",
" 'from_poi_to_this_person': 35,\n",
" 'from_this_person_to_poi': 48,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 1950,\n",
" 'poi': True,\n",
" 'restricted_stock': 283649,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 249201,\n",
" 'shared_receipt_with_poi': 1258,\n",
" 'to_messages': 1328,\n",
" 'total_payments': 1099100,\n",
" 'total_stock_value': 1918887},\n",
" 'SAVAGE FRANK': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -121284,\n",
" 'director_fees': 125034,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 3750,\n",
" 'total_stock_value': 'NaN'},\n",
" 'SCRIMSHAW MATTHEW': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'matthew.scrimshaw@enron.com',\n",
" 'exercised_stock_options': 759557,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 759557},\n",
" 'SHANKMAN JEFFREY A': {'bonus': 2000000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'jeffrey.shankman@enron.com',\n",
" 'exercised_stock_options': 1441898,\n",
" 'expenses': 178979,\n",
" 'from_messages': 2681,\n",
" 'from_poi_to_this_person': 94,\n",
" 'from_this_person_to_poi': 83,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 554422,\n",
" 'other': 1191,\n",
" 'poi': False,\n",
" 'restricted_stock': 630137,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 304110,\n",
" 'shared_receipt_with_poi': 1730,\n",
" 'to_messages': 3221,\n",

" 'total_payments': 3038702,\n",
" 'total_stock_value': 2072035},\n",
" 'SHAPIRO RICHARD S': {'bonus': 650000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'richard.shapiro@enron.com',\n",
" 'exercised_stock_options': 607837,\n",
" 'expenses': 137767,\n",
" 'from_messages': 1215,\n",
" 'from_poi_to_this_person': 74,\n",
" 'from_this_person_to_poi': 65,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 705,\n",
" 'poi': False,\n",
" 'restricted_stock': 379164,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 269076,\n",
" 'shared_receipt_with_poi': 4527,\n",
" 'to_messages': 15149,\n",
" 'total_payments': 1057548,\n",
" 'total_stock_value': 987001},\n",
" 'SHARP VICTORIA T': {'bonus': 600000,\n",
" 'deferral_payments': 187469,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'vicki.sharp@enron.com',\n",
" 'exercised_stock_options': 281073,\n",
" 'expenses': 116337,\n",
" 'from_messages': 136,\n",
" 'from_poi_to_this_person': 24,\n",
" 'from_this_person_to_poi': 6,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 422158,\n",
" 'other': 2401,\n",
" 'poi': False,\n",
" 'restricted_stock': 213063,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 248146,\n",
" 'shared_receipt_with_poi': 2477,\n",
" 'to_messages': 3136,\n",
" 'total_payments': 1576511,\n",
" 'total_stock_value': 494136},\n",
" 'SHELBY REX': {'bonus': 200000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -4167,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'rex.shelby@enron.com',\n",
" 'exercised_stock_options': 1624396,\n",
" 'expenses': 22884,\n",
" 'from_messages': 39,\n",
" 'from_poi_to_this_person': 13,\n",
" 'from_this_person_to_poi': 14,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 1573324,\n",
" 'poi': True,\n",
" 'restricted_stock': 869220,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 211844,\n",
" 'shared_receipt_with_poi': 91,\n",
" 'to_messages': 225,\n",
" 'total_payments': 2003885,\n",
" 'total_stock_value': 2493616},\n",
" 'SHERRICK JEFFREY B': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'jeffrey.sherrick@enron.com',\n",
" 'exercised_stock_options': 1426469,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 25,\n",
" 'from_poi_to_this_person': 39,\n",
" 'from_this_person_to_poi': 18,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 405999,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 583,\n",
" 'to_messages': 613,\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 1832468},\n",
" 'SHERRIFF JOHN R': {'bonus': 1500000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'john.sherriff@enron.com',\n",
" 'exercised_stock_options': 1835558,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 92,\n",
" 'from_poi_to_this_person': 28,\n",
" 'from_this_person_to_poi': 23,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 554422,\n",
" 'other': 1852186,\n",
" 'poi': False,\n",
" 'restricted_stock': 1293424,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 428780,\n",

" 'shared_receipt_with_poi': 2103,\n",
" 'to_messages': 3187,\n",
" 'total_payments': 4335388,\n",
" 'total_stock_value': 3128982},\n",
" 'SKILLING JEFFREY K': {'bonus': 5600000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'jeff.skilling@enron.com',\n",
" 'exercised_stock_options': 19250000,\n",
" 'expenses': 29336,\n",
" 'from_messages': 108,\n",
" 'from_poi_to_this_person': 88,\n",
" 'from_this_person_to_poi': 30,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 1920000,\n",
" 'other': 22122,\n",
" 'poi': True,\n",
" 'restricted_stock': 6843672,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 1111258,\n",
" 'shared_receipt_with_poi': 2042,\n",
" 'to_messages': 3627,\n",
" 'total_payments': 8682716,\n",
" 'total_stock_value': 26093672},\n",
" 'STABLER FRANK': {'bonus': 500000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'frank.stabler@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 16514,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 356071,\n",
" 'poi': False,\n",
" 'restricted_stock': 511734,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 239502,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 1112087,\n",
" 'total_stock_value': 511734},\n",
" 'SULLIVAN-SHAKLOVITZ COLLEEN': {'bonus': 100000,\n",
" 'deferral_payments': 181993,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 1362375,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 554422,\n",
" 'other': 162,\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 162779,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 999356,\n",
" 'total_stock_value': 1362375},\n",
" 'SUNDE MARTIN': {'bonus': 700000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'marty.sunde@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 38,\n",
" 'from_poi_to_this_person': 37,\n",
" 'from_this_person_to_poi': 13,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 476451,\n",
" 'other': 111122,\n",
" 'poi': False,\n",
" 'restricted_stock': 698920,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 257486,\n",
" 'shared_receipt_with_poi': 2565,\n",
" 'to_messages': 2647,\n",
" 'total_payments': 1545059,\n",
" 'total_stock_value': 698920},\n",
" 'TAYLOR MITCHELL S': {'bonus': 600000,\n",
" 'deferral_payments': 227449,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'mitchell.taylor@enron.com',\n",
" 'exercised_stock_options': 3181250,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 29,\n",
" 'from_poi_to_this_person': 0,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 563798,\n",

" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 265214,\n",
" 'shared_receipt_with_poi': 300,\n",
" 'to_messages': 533,\n",
" 'total_payments': 1092663,\n",
" 'total_stock_value': 3745048},\n",
" 'THE TRAVEL AGENCY IN THE PARK': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 362096,\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 362096,\n",
" 'total_stock_value': 'NaN'},\n",
" 'THORN TERENCE H': {'bonus': 'NaN',\n",
" 'deferral_payments': 16586,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'terence.thorn@enron.com',\n",
" 'exercised_stock_options': 4452476,\n",
" 'expenses': 46145,\n",
" 'from_messages': 41,\n",
" 'from_poi_to_this_person': 0,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 200000,\n",
" 'other': 426629,\n",
" 'poi': False,\n",
" 'restricted_stock': 365320,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 222093,\n",
" 'shared_receipt_with_poi': 73,\n",
" 'to_messages': 266,\n",
" 'total_payments': 911453,\n",
" 'total_stock_value': 4817796},\n",
" 'TILNEY ELIZABETH A': {'bonus': 300000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -575000,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'elizabeth.tilney@enron.com',\n",
" 'exercised_stock_options': 591250,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 19,\n",
" 'from_poi_to_this_person': 10,\n",
" 'from_this_person_to_poi': 11,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 275000,\n",
" 'other': 152055,\n",
" 'poi': False,\n",
" 'restricted_stock': 576792,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 247338,\n",
" 'shared_receipt_with_poi': 379,\n",
" 'to_messages': 460,\n",
" 'total_payments': 399393,\n",
" 'total_stock_value': 1168042},\n",
" 'TOTAL': {'bonus': 97343619,\n",
" 'deferral_payments': 32083396,\n",
" 'deferred_income': -27992891,\n",
" 'director_fees': 1398517,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 311764000,\n",
" 'expenses': 5235198,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 83925000,\n",
" 'long_term_incentive': 48521928,\n",
" 'other': 42667589,\n",
" 'poi': False,\n",
" 'restricted_stock': 130322299,\n",
" 'restricted_stock_deferred': -7576788,\n",
" 'salary': 26704229,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 309886585,\n",
" 'total_stock_value': 434509511},\n",
" 'UMANOFF ADAM S': {'bonus': 788750,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'adam.umanoff@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 53122,\n",
" 'from_messages': 18,\n",
" 'from_poi_to_this_person': 12,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",

" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 288589,\n",
" 'shared_receipt_with_poi': 41,\n",
" 'to_messages': 111,\n",
" 'total_payments': 1130461,\n",
" 'total_stock_value': 'NaN'},\n",
" 'URQUHART JOHN A': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -36666,\n",
" 'director_fees': 36666,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 228656,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 228656,\n",
" 'total_stock_value': 'NaN'},\n",
" 'WAKEHAM JOHN': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 109298,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 103773,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 213071,\n",
" 'total_stock_value': 'NaN'},\n",
" 'WALLS JR ROBERT H': {'bonus': 850000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'rob.walls@enron.com',\n",
" 'exercised_stock_options': 4346544,\n",
" 'expenses': 50936,\n",
" 'from_messages': 146,\n",
" 'from_poi_to_this_person': 17,\n",
" 'from_this_person_to_poi': 0,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 540751,\n",
" 'other': 2,\n",
" 'poi': False,\n",
" 'restricted_stock': 1552453,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 357091,\n",
" 'shared_receipt_with_poi': 215,\n",
" 'to_messages': 671,\n",
" 'total_payments': 1798780,\n",
" 'total_stock_value': 5898997},\n",
" 'WALTERS GARETH W': {'bonus': 'NaN',\n",
" 'deferral_payments': 53625,\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 1030329,\n",
" 'expenses': 33785,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 87410,\n",
" 'total_stock_value': 1030329},\n",
" 'WASAFF GEORGE': {'bonus': 325000,\n",
" 'deferral_payments': 831299,\n",
" 'deferred_income': -583325,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'george.wasaff@enron.com',\n",
" 'exercised_stock_options': 1668260,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 30,\n",
" 'from_poi_to_this_person': 22,\n",
" 'from_this_person_to_poi': 7,\n",
" 'loan_advances': 'NaN',\n",

" 'long_term_incentive': 200000,\n",
" 'other': 1425,\n",
" 'poi': False,\n",
" 'restricted_stock': 388167,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 259996,\n",
" 'shared_receipt_with_poi': 337,\n",
" 'to_messages': 400,\n",
" 'total_payments': 1034395,\n",
" 'total_stock_value': 2056427},\n",
" 'WESTFAHL RICHARD K': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -10800,\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'dick.westfahl@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 51870,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 256191,\n",
" 'other': 401130,\n",
" 'poi': False,\n",
" 'restricted_stock': 384930,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 63744,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 762135,\n",
" 'total_stock_value': 384930},\n",
" 'WHALEY DAVID A': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 98718,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 98718},\n",
" 'WHALLEY LAWRENCE G': {'bonus': 3000000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'greg.whalley@enron.com',\n",
" 'exercised_stock_options': 3282960,\n",
" 'expenses': 57838,\n",
" 'from_messages': 556,\n",
" 'from_poi_to_this_person': 186,\n",
" 'from_this_person_to_poi': 24,\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 808346,\n",
" 'other': 301026,\n",
" 'poi': False,\n",
" 'restricted_stock': 2796177,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 510364,\n",
" 'shared_receipt_with_poi': 3920,\n",
" 'to_messages': 6019,\n",
" 'total_payments': 4677574,\n",
" 'total_stock_value': 6079137},\n",
" 'WHITE JR THOMAS E': {'bonus': 450000,\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'thomas.white@enron.com',\n",
" 'exercised_stock_options': 1297049,\n",
" 'expenses': 81353,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 1085463,\n",
" 'poi': False,\n",
" 'restricted_stock': 13847074,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 317543,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 1934359,\n",
" 'total_stock_value': 15144123},\n",
" 'WINOKUR JR. HERBERT S': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': -25000,\n",
" 'director_fees': 108579,\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 1413,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",

" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 84992,\n",
" 'total_stock_value': 'NaN'},\n",
" 'WODRASKA JOHN': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'john.wodraska@enron.com',\n",
" 'exercised_stock_options': 'NaN',\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 189583,\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 189583,\n",
" 'total_stock_value': 'NaN'},\n",
" 'WROBEL BRUCE': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 139130,\n",
" 'expenses': 'NaN',\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 'NaN',\n",
" 'total_stock_value': 139130},\n",
" 'YEAGER F SCOTT': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'scott.yeager@enron.com',\n",
" 'exercised_stock_options': 8308552,\n",
" 'expenses': 53947,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 147950,\n",
" 'poi': True,\n",
" 'restricted_stock': 3576206,\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 158403,\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 360300,\n",
" 'total_stock_value': 11884758},\n",
" 'YEAP SOON': {'bonus': 'NaN',\n",
" 'deferral_payments': 'NaN',\n",
" 'deferred_income': 'NaN',\n",
" 'director_fees': 'NaN',\n",
" 'email_address': 'NaN',\n",
" 'exercised_stock_options': 192758,\n",
" 'expenses': 55097,\n",
" 'from_messages': 'NaN',\n",
" 'from_poi_to_this_person': 'NaN',\n",
" 'from_this_person_to_poi': 'NaN',\n",
" 'loan_advances': 'NaN',\n",
" 'long_term_incentive': 'NaN',\n",
" 'other': 'NaN',\n",
" 'poi': False,\n",
" 'restricted_stock': 'NaN',\n",
" 'restricted_stock_deferred': 'NaN',\n",
" 'salary': 'NaN',\n",
" 'shared_receipt_with_poi': 'NaN',\n",
" 'to_messages': 'NaN',\n",
" 'total_payments': 55097,\n",
" 'total_stock_value': 192758}}"
]
},
"execution_count": 116,
"metadata": {},
"output_type": "execute_result"
}
],

```
  "source": [
  "enron_data."
  ]
  },
  {
  "cell_type": "code",
  "execution_count": 197,
  "metadata": {},
  "outputs": [],
  "source": [
  "def isNaN(value):\n",
  " return value ==\"NaN\"\n",
  " "
  ]
  },
  {
  "cell_type": "code",
  "execution_count": 198,
  "metadata": {},
  "outputs": [],
  "source": [
  "def isNotNaN(value):\n",
  " return not isNaN(value)"
  ]
  },
  {
  "cell_type": "code",
  "execution_count": 201,
  "metadata": {},
  "outputs": [
  {
  "data": {
  "text/plain": [
  "False"
  ]
  },
  "execution_count": 201,
  "metadata": {},
  "output_type": "execute_result"
  }
  ],
  "source": [
  "isNaN(\"D\")"
  ]
  },
  {
  "cell_type": "code",
  "execution_count": 195,
  "metadata": {},
  "outputs": [
  {
  "name": "stdout",
  "output_type": "stream",
  "text": [
  "95\n",
  "111\n"
  ]
  }
  ],
  "source": [
  "sCount=0\n",
  "eCount=0\n",
  "for key in enron_data:\n",
  " #print(enron_data[key])\n",
  " if isNotNaN(enron_data[key][\"salary\"]):\n",
  " sCount+=1\n",
  " if isNotNaN(enron_data[key][\"email_address\"]):\n",
  " eCount+=1\n",
  "print(sCount)\n",
  "print(eCount)\n",
  " \n",
  " \n",
  " \n",
  " \n",
  " "
  ]
  },
  {
  "cell_type": "code",
  "execution_count": 115,
  "metadata": {
  "scrolled": true
  },
  "outputs": [
  {
  "data": {
  "text/plain": [
  "21"
  ]
  },
  "execution_count": 115,
  "metadata": {},
  "output_type": "execute_result"
  }
  ],
  "source": [
  "len(enron_data[\"METTS MARK\"])"
  ]
  },
  {
  "cell_type": "code",
  "execution_count": 3,
  "metadata": {},
```

"outputs": [],
"source": [
"import sys"
]
},
{
"cell_type": "code",
"execution_count": 213,
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"0\n",
"146\n",
"0.0\n"
]
}
],
"source": [
"count=0\n",
"tpNanCnt=0\n",
"for key in enron_data:\n",
" # print(key)\n",
" # print(enron_data[key])\n",
" value=enron_data[key]\n",
" if isNaN(value['total_payments']):\n",
" tpNanCnt+=1\n",
" if value[\"poi\"]:\n",
" count=count+1\n",
"print(count)\n",
"print(len(enron_data))\n",
"print(count/len(enron_data))\n",
"\n",
" "
]
},
{
"cell_type": "code",
"execution_count": 166,
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"dict_keys(['METTS MARK', 'BAXTER JOHN C', 'ELLIOTT STEVEN', 'CORDES WILLIAM R', 'HANNON KEVIN P', 'MORDAUNT KRISTINA M', 'MEYER ROCKFORD G', 'MCMAHON JEFFREY', 'HAEDICKE MARK E', 'PIPER GREGORY F', 'HUMPHREY GENE E', 'NOLES JAMES L', 'BLACHMAN JEREMY M', 'SUNDE MARTIN', 'GIBBS DANA R', 'LOWRY CHARLES P', 'COLWELL WESLEY', 'MULLER MARK S', 'JACKSON CHARLENE R', 'WESTFAHL RICHARD K', 'WALTERS GARETH W', 'WALLS JR ROBERT H', 'KITCHEN LOUISE', 'CHAN RONNIE', 'BELFER ROBERT', 'SHANKMAN JEFFREY A', 'WODRASKA JOHN', 'BERGSIEKER RICHARD P', 'URQUHART JOHN A', 'BIBI PHILIPPE A', 'RIEKER PAULA H', 'WHALEY DAVID A', 'BECK SALLY W', 'HAUG DAVID L', 'ECHOLS JOHN B', 'MENDELSOHN JOHN', 'HICKERSON GARY J', 'CLINE KENNETH W', 'LEWIS RICHARD', 'HAYES ROBERT E', 'KOPPER MICHAEL J', 'LEFF DANIEL P', 'LAVORATO JOHN J', 'BERBERIAN DAVID', 'DETMERING TIMOTHY J', 'WAKEHAM JOHN', 'POWERS WILLIAM', 'GOLD JOSEPH', 'BANNANTINE JAMES M', 'DUNCAN JOHN H', 'SHAPIRO RICHARD S', 'SHERRIFF JOHN R', 'SHELBY REX', 'LEMAISTRE CHARLES', 'DEFFNER JOSEPH M', 'KISHKILL JOSEPH G', 'WHALLEY LAWRENCE G', 'MCCONNELL MICHAEL S', 'PIRO JIM', 'DELAINEY DAVID W', 'SULLIVAN-SHAKLOVITZ COLLEEN', 'WROBEL BRUCE', 'LINDHOLM TOD A', 'MEYER JEROME J', 'LAY KENNETH L', 'BUTTS ROBERT H', 'OLSON CINDY K', 'MCDONALD REBECCA', 'CUMBERLAND MICHAEL S', 'GAHN ROBERT S', 'BADUM JAMES P', 'HERMANN ROBERT J', 'FALLON JAMES B', 'GATHMANN WILLIAM D', 'HORTON STANLEY C', 'BOWEN JR RAYMOND M', 'GILLIS JOHN', 'FITZGERALD JAY L', 'MORAN MICHAEL P', 'REDMOND BRIAN L', 'BAZELIDES PHILIP J', 'BELDEN TIMOTHY N', 'DIMICHELE RICHARD G', 'DURAN WILLIAM D', 'THORN TERENCE H', 'FASTOW ANDREW S', 'FOY JOE', 'CALGER CHRISTOPHER F', 'RICE KENNETH D', 'KAMINSKI WINCENTY J', 'LOCKHART EUGENE E', 'COX DAVID', 'OVERDYKE JR JERE C', 'PEREIRA PAULO V. FERRAZ', 'STABLER FRANK', 'SKILLING JEFFREY K', 'BLAKE JR. NORMAN P', 'SHERRICK JEFFREY B', 'PRENTICE JAMES', 'GRAY RODNEY', 'THE TRAVEL AGENCY IN THE PARK', 'UMANOFF ADAM S', 'KEAN STEVEN J', 'TOTAL', 'FOWLER PEGGY', 'WASAFF GEORGE', 'WHITE JR THOMAS E', 'CHRISTODOULOU DIOMEDES', 'ALLEN PHILLIP K', 'SHARP VICTORIA T', 'JAEDICKE ROBERT', 'WINOKUR JR. HERBERT S', 'BROWN MICHAEL', 'MCCLELLAN GEORGE', 'HUGHES JAMES A', 'REYNOLDS LAWRENCE', 'PICKERING MARK R', 'BHATNAGAR SANJAY', 'CARTER REBECCA C', 'BUCHANAN HAROLD G', 'YEAP SOON', 'MURRAY JULIA H', 'GARLAND C KEVIN', 'DODSON KEITH', 'YEAGER F SCOTT', 'HIRKO JOSEPH', 'DIETRICH JANET R', 'DERRICK JR. JAMES V', 'FREVERT MARK A', 'PAI LOU L', 'HAYSLETT RODERICK J', 'BAY FRANKLIN R', 'MCCARTY DANNY J', 'FUGH JOHN L', 'SCRIMSHAW MATTHEW', 'KOENIG MARK E', 'SAVAGE FRANK', 'IZZO LAWRENCE L', 'TILNEY ELIZABETH A', 'MARTIN AMANDA K', 'BUY RICHARD B', 'GRAMM WENDY L', 'CAUSEY RICHARD A', 'TAYLOR MITCHELL S', 'DONAHUE JR JEFFREY M', 'GLISAN JR BEN F'])"
]
},
"execution_count": 166,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"enron_data.keys()"
]
},
{
"cell_type": "code",
"execution_count": 152,
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"1095040"
]
},
"execution_count": 152,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"enron_data[\"PRENTICE JAMES\"][\"total_stock_value\"]"
]
},
{
"cell_type": "code",
"execution_count": 156,
"metadata": {},

"outputs": [],
"source": [
"xx=(enron_data[\"PRENTICE JAMES\"])"
]
},
{
"cell_type": "code",
"execution_count": 202,
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"dict"
]
},
"execution_count": 202,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"type(enron_data)"
]
},
{
"cell_type": "code",
"execution_count": 161,
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"11"
]
},
"execution_count": 161,
"metadata": {},
"output_type": "execute_result"
}
],
"source": []
}
],
"metadata": {
"kernelspec": {
"display_name": "Python 3",
"language": "python",
"name": "python3"
},
"language_info": {
"codemirror_mode": {
"name": "ipython",
"version": 3
},
"file_extension": ".py",
"mimetype": "text/x-python",
"name": "python",
"nbconvert_exporter": "python",
"pygments_lexer": "ipython3",
"version": "3.6.4"
}
},
"nbformat": 4,
"nbformat_minor": 2
}

File: ./final_project/poi_id.py

```python
1   #!/usr/bin/python
2
3   import sys
4   import pickle
5   sys.path.append("../tools/")
6
7   from feature_format import featureFormat, targetFeatureSplit
8   from tester import dump_classifier_and_data
9
10  ### Task 1: Select what features you'll use.
11  ### features_list is a list of strings, each of which is a feature name.
12  ### The first feature must be "poi".
13  features_list = ['poi','salary'] # You will need to use more features
14
15  ### Load the dictionary containing the dataset
16  with open("final_project_dataset.pkl", "r") as data_file:
17      data_dict = pickle.load(data_file)
18
19  ### Task 2: Remove outliers
20  ### Task 3: Create new feature(s)
21  ### Store to my_dataset for easy export below.
22  my_dataset = data_dict
23
24  ### Extract features and labels from dataset for local testing
25  data = featureFormat(my_dataset, features_list, sort_keys = True)
26  labels, features = targetFeatureSplit(data)
27
28  ### Task 4: Try a varity of classifiers
29  ### Please name your classifier clf for easy export below.
30  ### Note that if you want to do PCA or other multi-stage operations,
31  ### you'll need to use Pipelines. For more info:
32  ### http://scikit-learn.org/stable/modules/pipeline.html
33
34  # Provided to give you a starting point. Try a variety of classifiers.
35  from sklearn.naive_bayes import GaussianNB
36  clf = GaussianNB()
```

```
37
38   ### Task 5: Tune your classifier to achieve better than .3 precision and recall
39   ### using our testing script. Check the tester.py script in the final project
40   ### folder for details on the evaluation method, especially the test_classifier
41   ### function. Because of the small size of the dataset, the script uses
42   ### stratified shuffle split cross validation. For more info:
43   ### http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.StratifiedShuffleSplit.html
44
45   # Example starting point. Try investigating other evaluation techniques!
46   from sklearn.cross_validation import train_test_split
47   features_train, features_test, labels_train, labels_test = \
48       train_test_split(features, labels, test_size=0.3, random_state=42)
49
50   ### Task 6: Dump your classifier, dataset, and features_list so anyone can
51   ### check your results. You do not need to change anything below, but make sure
52   ### that the version of poi_id.py that you submit can be run on its own and
53   ### generates the necessary .pkl files for validating your results.
54
55   dump_classifier_and_data(clf, my_dataset, features_list)
```

File: ./final_project/tester.py

```
1    #!/usr/bin/pickle
2
3    """ a basic script for importing student's POI identifier,
4        and checking the results that they get from it
5
6        requires that the algorithm, dataset, and features list
7        be written to my_classifier.pkl, my_dataset.pkl, and
8        my_feature_list.pkl, respectively
9
10       that process should happen at the end of poi_id.py
11   """
12
13   import pickle
14   import sys
15   from sklearn.cross_validation import StratifiedShuffleSplit
16   sys.path.append("../tools/")
17   from feature_format import featureFormat, targetFeatureSplit
18
19   PERF_FORMAT_STRING = "\
20   \tAccuracy: {:>0.{display_precision}f}\tPrecision: {:>0.{display_precision}f}\t\
21   Recall: {:>0.{display_precision}f}\tF1: {:>0.{display_precision}f}\tF2: {:>0.{display_precision}f}"
22   RESULTS_FORMAT_STRING = "\tTotal predictions: {:4d}\tTrue positives: {:4d}\tFalse positives: {:4d}\
23   \tFalse negatives: {:4d}\tTrue negatives: {:4d}"
24
25   def test_classifier(clf, dataset, feature_list, folds = 1000):
26       data = featureFormat(dataset, feature_list, sort_keys = True)
27       labels, features = targetFeatureSplit(data)
28       cv = StratifiedShuffleSplit(labels, folds, random_state = 42)
29       true_negatives = 0
30       false_negatives = 0
31       true_positives = 0
32       false_positives = 0
33       for train_idx, test_idx in cv:
34           features_train = []
35           features_test  = []
36           labels_train   = []
37           labels_test    = []
38           for ii in train_idx:
39               features_train.append( features[ii] )
40               labels_train.append( labels[ii] )
41           for jj in test_idx:
42               features_test.append( features[jj] )
43               labels_test.append( labels[jj] )
44
45           ### fit the classifier using training set, and test on test set
46           clf.fit(features_train, labels_train)
47           predictions = clf.predict(features_test)
48           for prediction, truth in zip(predictions, labels_test):
49               if prediction == 0 and truth == 0:
50                   true_negatives += 1
51               elif prediction == 0 and truth == 1:
52                   false_negatives += 1
53               elif prediction == 1 and truth == 0:
54                   false_positives += 1
55               elif prediction == 1 and truth == 1:
56                   true_positives += 1
57               else:
58                   print "Warning: Found a predicted label not == 0 or 1."
59                   print "All predictions should take value 0 or 1."
60                   print "Evaluating performance for processed predictions:"
61                   break
62       try:
63           total_predictions = true_negatives + false_negatives + false_positives + true_positives
64           accuracy = 1.0*(true_positives + true_negatives)/total_predictions
65           precision = 1.0*true_positives/(true_positives+false_positives)
66           recall = 1.0*true_positives/(true_positives+false_negatives)
67           f1 = 2.0 * true_positives/(2*true_positives + false_positives+false_negatives)
68           f2 = (1+2.0*2.0) * precision*recall/(4*precision + recall)
69           print clf
70           print PERF_FORMAT_STRING.format(accuracy, precision, recall, f1, f2, display_precision = 5)
71           print RESULTS_FORMAT_STRING.format(total_predictions, true_positives, false_positives, false_negatives, true_negatives)
72           print ""
73       except:
74           print "Got a divide by zero when trying out:", clf
75           print "Precision or recall may be undefined due to a lack of true positive predicitons."
76
77   CLF_PICKLE_FILENAME = "my_classifier.pkl"
78   DATASET_PICKLE_FILENAME = "my_dataset.pkl"
79   FEATURE_LIST_FILENAME = "my_feature_list.pkl"
80
81   def dump_classifier_and_data(clf, dataset, feature_list):
82       with open(CLF_PICKLE_FILENAME, "w") as clf_outfile:
83           pickle.dump(clf, clf_outfile)
84       with open(DATASET_PICKLE_FILENAME, "w") as dataset_outfile:
85           pickle.dump(dataset, dataset_outfile)
86       with open(FEATURE_LIST_FILENAME, "w") as featurelist_outfile:
87           pickle.dump(feature_list, featurelist_outfile)
88
89   def load_classifier_and_data():
90       with open(CLF_PICKLE_FILENAME, "r") as clf_infile:
```

```
 91          clf = pickle.load(clf_infile)
 92      with open(DATASET_PICKLE_FILENAME, "r") as dataset_infile:
 93          dataset = pickle.load(dataset_infile)
 94      with open(FEATURE_LIST_FILENAME, "r") as featurelist_infile:
 95          feature_list = pickle.load(featurelist_infile)
 96      return clf, dataset, feature_list
 97
 98  def main():
 99      ### load up student's classifier, dataset, and feature_list
100      clf, dataset, feature_list = load_classifier_and_data()
101      ### Run testing script
102      test_classifier(clf, dataset, feature_list)
103
104  if __name__ == '__main__':
105      main()
```

File: ./final_project/poi_email_addresses.py

```
 1  def poiEmails():
 2      email_list = ["kenneth_lay@enron.net",
 3              "kenneth_lay@enron.com",
 4              "klay.enron@enron.com",
 5              "kenneth.lay@enron.com",
 6              "klay@enron.com",
 7              "layk@enron.com",
 8              "chairman.ken@enron.com",
 9              "jeffreyskilling@yahoo.com",
10              "jeff_skilling@enron.com",
11              "jskilling@enron.com",
12              "effrey.skilling@enron.com",
13              "skilling@enron.com",
14              "jeffrey.k.skilling@enron.com",
15              "jeff.skilling@enron.com",
16              "kevin_a_howard.enronxgate.enron@enron.net",
17              "kevin.howard@enron.com",
18              "kevin.howard@enron.net",
19              "kevin.howard@gcm.com",
20              "michael.krautz@enron.com"
21              "scott.yeager@enron.com",
22              "syeager@fyi-net.com",
23              "scott_yeager@enron.net",
24              "syeager@flash.net",
25              "joe'.'hirko@enron.com",
26              "joe.hirko@enron.com",
27              "rex.shelby@enron.com",
28              "rex.shelby@enron.nt",
29              "rex_shelby@enron.net",
30              "jbrown@enron.com",
31              "james.brown@enron.com",
32              "rick.causey@enron.com",
33              "richard.causey@enron.com",
34              "rcausey@enron.com",
35              "calger@enron.com",
36              "chris.calger@enron.com",
37              "christopher.calger@enron.com",
38              "ccalger@enron.com",
39              "tim_despain.enronxgate.enron@enron.net",
40              "tim.despain@enron.com",
41              "kevin_hannon@enron.com",
42              "kevin'.'hannon@enron.com",
43              "kevin_hannon@enron.net",
44              "kevin.hannon@enron.com",
45              "mkoenig@enron.com",
46              "mark.koenig@enron.com",
47              "m..forney@enron.com",
48              "ken'.'rice@enron.com",
49              "ken.rice@enron.com",
50              "ken_rice@enron.com",
51              "ken_rice@enron.net",
52              "paula.rieker@enron.com",
53              "prieker@enron.com",
54              "andrew.fastow@enron.com",
55              "lfastow@pdq.net",
56              "andrew.s.fastow@enron.com",
57              "lfastow@pop.pdq.net",
58              "andy.fastow@enron.com",
59              "david.w.delainey@enron.com",
60              "delainey.dave@enron.com",
61              "'delainey@enron.com",
62              "david.delainey@enron.com",
63              "'david.delainey'@enron.com",
64              "dave.delainey@enron.com",
65              "delainey'.'david@enron.com",
66              "ben.glisan@enron.com",
67              "bglisan@enron.com",
68              "ben_f_glisan@enron.com",
69              "ben'.'glisan@enron.com",
70              "jeff.richter@enron.com",
71              "jrichter@nwlink.com",
72              "lawrencelawyer@aol.com",
73              "lawyer'.'larry@enron.com",
74              "larry_lawyer@enron.com",
75              "llawyer@enron.com",
76              "larry.lawyer@enron.com",
77              "lawrence.lawyer@enron.com",
78              "tbelden@enron.com",
79              "tim.belden@enron.com",
80              "tim_belden@pgn.com",
81              "tbelden@ect.enron.com",
82              "michael.kopper@enron.com",
83              "dave.duncan@enron.com",
84              "dave.duncan@cipco.org",
85              "duncan.dave@enron.com",
86              "ray.bowen@enron.com",
87              "raymond.bowen@enron.com",
88              "'bowen@enron.com",
89              "wes.colwell@enron.com",
90              "dan.boyle@enron.com",
91              "cloehr@enron.com",
92              "chris.loehr@enron.com"
93              ]
94      return email_list
```

File: ./README.md

```
1 ud120-projects
2 ==============
3
4 Starter project code for students taking Udacity ud120
```

File: ./decision_tree/dt_author_id.py

```python
1   #!/usr/bin/python
2
3   """
4       This is the code to accompany the Lesson 3 (decision tree) mini-project.
5
6       Use a Decision Tree to identify emails from the Enron corpus by author:
7       Sara has label 0
8       Chris has label 1
9   """
10
11  import sys
12  from time import time
13  sys.path.append("../tools/")
14  from email_preprocess import preprocess
15
16
17  ### features_train and features_test are the features for the training
18  ### and testing datasets, respectively
19  ### labels_train and labels_test are the corresponding item labels
20  features_train, features_test, labels_train, labels_test = preprocess()
21
22
23
24
25  #########################################################
26  ### your code goes here ###
27
28
29  #########################################################
30
31
```

File: ./feature_selection/find_signature.py

```python
1   #!/usr/bin/python
2
3   import pickle
4   import numpy
5   numpy.random.seed(42)
6
7
8   ### The words (features) and authors (labels), already largely processed.
9   ### These files should have been created from the previous (Lesson 10)
10  ### mini-project.
11  words_file = "../text_learning/your_word_data.pkl"
12  authors_file = "../text_learning/your_email_authors.pkl"
13  word_data = pickle.load( open(words_file, "rb"))
14  authors = pickle.load( open(authors_file, "rb") )
15
16
17
18  ### test_size is the percentage of events assigned to the test set (the
19  ### remainder go into training)
20  ### feature matrices changed to dense representations for compatibility with
21  ### classifier functions in versions 0.15.2 and earlier
22  from sklearn.model_selection import train_test_split
23  features_train, features_test, labels_train, labels_test = \
24      train_test_split(word_data, authors, test_size=0.1, random_state=42)
25
26  from sklearn.feature_extraction.text import TfidfVectorizer
27  vectorizer = TfidfVectorizer(sublinear_tf=True, max_df=0.5,
28                               stop_words='english')
29  features_train = vectorizer.fit_transform(features_train)
30  features_test  = vectorizer.transform(features_test).toarray()
31
32
33  ### a classic way to overfit is to use a small number
34  ### of data points and a large number of features;
35  ### train on only 150 events to put ourselves in this regime
36  features_train = features_train[:150].toarray()
37  labels_train   = labels_train[:150]
38
39
40
41  ### your code goes here
42
43
44
```

File: ./regression/finance_regression.ipynb

```
{
"cells": [
{
"cell_type": "code",
"execution_count": 18,
"metadata": {},
"outputs": [],
"source": [
"%matplotlib inline"
]
},
{
```

"cell_type": "code",
"execution_count": 2,
"metadata": {},
"outputs": [],
"source": [
"import sys\n",
"import pickle\n",
"sys.path.append(\"../tools/\")\n",
"from feature_format import featureFormat, targetFeatureSplit"
]
},
{
"cell_type": "code",
"execution_count": 4,
"metadata": {},
"outputs": [],
"source": [
"dictionary = pickle.load(open(\"../final_project/final_project_dataset_modified_py3.pkl\", \"rb\"))"
]
},
{
"cell_type": "code",
"execution_count": 7,
"metadata": {},
"outputs": [],
"source": [
"### list the features you want to look at--first item in the \n",
"### list will be the \"target\" feature\n",
"features_list = [\"bonus\", \"salary\"]\n",
"data = featureFormat(dictionary, features_list, remove_any_zeroes=True,\n",
" sort_keys='../tools/python2_lesson06_keys_py3.pkl')\n",
" # see https://classroom.udacity.com/courses/ud120/lessons/2301748537/concepts/30416086000923"
]
},
{
"cell_type": "code",
"execution_count": 25,
"metadata": {},
"outputs": [],
"source": [
"target, features = targetFeatureSplit(data)"
]
},
{
"cell_type": "code",
"execution_count": 26,
"metadata": {},
"outputs": [],
"source": [
"### training-testing split needed in regression, just like classification\n",
"from sklearn.model_selection import train_test_split\n",
"feature_train, feature_test, target_train, target_test = train_test_split(features, target, test_size=0.5, random_state=42)\n",
"train_color = \"b\"\n",
"test_color = \"r\""
]
},
{
"cell_type": "code",
"execution_count": 29,
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)"
]
},
"execution_count": 29,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"### Your regression goes here!\n",
"### Please name it reg, so that the plotting code below picks it up and \n",
"### plots it correctly. Don't forget to change the test_color above from \"b\" to\n",
"### \"r\" to differentiate training points from test points.\n",
"\n",
"from sklearn import linear_model\n",
"# Create linear regression object\n",
"reg= linear_model.LinearRegression()\n",
"\n",
"# Train the model using the training sets\n",
"reg.fit(feature_train, target_train)"
]
},
{
"cell_type": "code",
"execution_count": 36,
"metadata": {},
"outputs": [
{
"data": {
"image/png":
"iVBORw0KGgoAAAANSUhEUgAAAaEAAAEKCAYAAAC7c+rvAAAABHNCSVQICAgIfAhkiAAAAlwSFlzAAALEgAACxIB0t1+/AAADI0RVh0U29mdHdhcmUAbWF0cG
"text/plain": [
""
]
},
"metadata": {},
"output_type": "display_data"
}
],

```json
"source": [
"### draw the scatterplot, with color-coded training and testing points\n",
"import matplotlib.pyplot as plt\n",
"for feature, target in zip(feature_test, target_test):\n",
" plt.scatter( feature, target, color=test_color ) \n",
"for feature, target in zip(feature_train, target_train):\n",
" plt.scatter( feature, target, color=train_color ) \n",
"\n",
"### labels for the legend\n",
"plt.scatter(feature_test[0], target_test[0], color=test_color, label=\"test\")\n",
"plt.scatter(feature_test[0], target_test[0], color=train_color, label=\"train\")\n",
"\n",
"### draw the regression line, once it's coded\n",
"try:\n",
" plt.plot( feature_test, reg.predict(feature_test) )\n",
"except NameError:\n",
" pass\n",
"plt.xlabel(features_list[1])\n",
"plt.ylabel(features_list[0])\n",
"plt.legend()\n",
"plt.show()\n"
]
},
{
"cell_type": "code",
"execution_count": null,
"metadata": {},
"outputs": [],
"source": []
}
],
"metadata": {
"kernelspec": {
"display_name": "Python 3",
"language": "python",
"name": "python3"
},
"language_info": {
"codemirror_mode": {
"name": "ipython",
"version": 3
},
"file_extension": ".py",
"mimetype": "text/x-python",
"name": "python",
"nbconvert_exporter": "python",
"pygments_lexer": "ipython3",
"version": "3.6.4"
}
},
"nbformat": 4,
"nbformat_minor": 2
}
```

File: ./regression/finance_regression.py

```python
1   #!/usr/bin/python
2
3   """
4       Starter code for the regression mini-project.
5
6       Loads up/formats a modified version of the dataset
7       (why modified?  we've removed some trouble points
8       that you'll find yourself in the outliers mini-project).
9
10      Draws a little scatterplot of the training/testing data
11
12      You fill in the regression code where indicated:
13  """
14
15
16  import sys
17  import pickle
18  sys.path.append("../tools/")
19  from feature_format import featureFormat, targetFeatureSplit
20  dictionary = pickle.load( open("../final_project/final_project_dataset_modified.pkl", "rb") )
21
22  ### list the features you want to look at--first item in the
23  ### list will be the "target" feature
24  features_list = ["bonus", "salary"]
25  data = featureFormat( dictionary, features_list, remove_any_zeroes=True,
26                        sort_keys='../tools/python2_lesson06_keys.pkl')
27                        # see https://classroom.udacity.com/courses/ud120/lessons/2301748537/concepts/30416086000923
28  target, features = targetFeatureSplit( data )
29
30  ### training-testing split needed in regression, just like classification
31  from sklearn.model_selection import train_test_split
32  feature_train, feature_test, target_train, target_test = train_test_split(features, target, test_size=0.5, random_state=42)
33  train_color = "b"
34  test_color = "b"
35
36
37
38  ### Your regression goes here!
39  ### Please name it reg, so that the plotting code below picks it up and
40  ### plots it correctly. Don't forget to change the test_color above from "b" to
41  ### "r" to differentiate training points from test points.
42
43
44
45
46
47
48
49
50  ### draw the scatterplot, with color-coded training and testing points
```

```
51 import matplotlib.pyplot as plt
52 for feature, target in zip(feature_test, target_test):
53     plt.scatter( feature, target, color=test_color )
54 for feature, target in zip(feature_train, target_train):
55     plt.scatter( feature, target, color=train_color )
56
57 ### labels for the legend
58 plt.scatter(feature_test[0], target_test[0], color=test_color, label="test")
59 plt.scatter(feature_test[0], target_test[0], color=train_color, label="train")
60
61
62
63
64 ### draw the regression line, once it's coded
65 try:
66     plt.plot( feature_test, reg.predict(feature_test) )
67 except NameError:
68     pass
69 plt.xlabel(features_list[1])
70 plt.ylabel(features_list[0])
71 plt.legend()
72 plt.show()
```

File: ./.gitignore

```
*.pyc
enron_mail_20110402.tgz
enron_mail_20110402/
enron_mail_20150507.tgz
maildir/
text_learning/your_word_data.pkl
text_learning/your_email_authors.pkl
my_classifier.pkl
my_dataset.pkl
my_feature_list.pkl
.idea
.ipynb_checkpoints
```

File: ./evaluation/evaluate_poi_identifier.py

```
1   #!/usr/bin/python
2
3
4   """
5       Starter code for the evaluation mini-project.
6       Start by copying your trained/tested POI identifier from
7       that which you built in the validation mini-project.
8
9       This is the second step toward building your POI identifier!
10
11      Start by loading/formatting the data...
12  """
13
14 import pickle
15 import sys
16 sys.path.append("../tools/")
17 from feature_format import featureFormat, targetFeatureSplit
18
19 data_dict = pickle.load(open("../final_project/final_project_dataset.pkl", "rb") )
20
21 ### add more features to features_list!
22 features_list = ["poi", "salary"]
23
24 data = featureFormat(data_dict, features_list)
25 labels, features = targetFeatureSplit(data)
26
27
28
29 ### your code goes here
30
31
```

File: ./k_means/k_means_cluster.py

```
1   #!/usr/bin/python
2
3   """
4       Skeleton code for k-means clustering mini-project.
5   """
6
7
8
9
10 import pickle
11 import numpy
12 import matplotlib.pyplot as plt
13 import sys
14 sys.path.append("../tools/")
15 from feature_format import featureFormat, targetFeatureSplit
16
17
18
19
20 def Draw(pred, features, poi, mark_poi=False, name="image.png", f1_name="feature 1", f2_name="feature 2"):
21     """ some plotting code designed to help you visualize your clusters """
22
23     ### plot each cluster with a different color--add more colors for
24     ### drawing more than five clusters
25     colors = ["b", "c", "k", "m", "g"]
26     for ii, pp in enumerate(pred):
27         plt.scatter(features[ii][0], features[ii][1], color = colors[pred[ii]])
28
29     ### if you like, place red stars over points that are POIs (just for funsies)
30     if mark_poi:
31         for ii, pp in enumerate(pred):
32             if poi[ii]:
```

```
33              plt.scatter(features[ii][0], features[ii][1], color="r", marker="*")
34      plt.xlabel(f1_name)
35      plt.ylabel(f2_name)
36      plt.savefig(name)
37      plt.show()
38
39
40
41  ### load in the dict of dicts containing all the data on each person in the dataset
42  data_dict = pickle.load(open("../final_project/final_project_dataset.pkl", "rb"))
43  ### there's an outlier--remove it!
44  data_dict.pop("TOTAL", 0)
45
46
47  ### the input features we want to use
48  ### can be any key in the person-level dictionary (salary, director_fees, etc.)
49  feature_1 = "salary"
50  feature_2 = "exercised_stock_options"
51  poi  = "poi"
52  features_list = [poi, feature_1, feature_2]
53  data = featureFormat(data_dict, features_list )
54  poi, finance_features = targetFeatureSplit( data )
55
56
57  ### in the "clustering with 3 features" part of the mini-project,
58  ### you'll want to change this line to
59  ### for f1, f2, _ in finance_features:
60  ### (as it's currently written, the line below assumes 2 features)
61  for f1, f2 in finance_features:
62      plt.scatter(f1, f2)
63  plt.show()
64
65  ### cluster here; create predictions of the cluster labels
66  ### for the data and store them to a list called pred
67
68
69
70
71  ### rename the "name" parameter when you change the number of features
72  ### so that the figure gets saved to a different file
73  try:
74      Draw(pred, finance_features, poi, mark_poi=False, name="clusters.pdf", f1_name=feature_1, f2_name=feature_2)
75  except NameError:
76      print("no predictions object named pred found, no clusters to plot")
```

File: ./choose_your_own/your_algorithm.py

```
1   #!/usr/bin/python
2
3   import matplotlib.pyplot as plt
4   from prep_terrain_data import makeTerrainData
5   from class_vis import prettyPicture
6
7   features_train, labels_train, features_test, labels_test = makeTerrainData()
8
9
10  ### the training data (features_train, labels_train) have both "fast" and "slow"
11  ### points mixed together--separate them so we can give them different colors
12  ### in the scatterplot and identify them visually
13  grade_fast = [features_train[ii][0] for ii in range(0, len(features_train)) if labels_train[ii]==0]
14  bumpy_fast = [features_train[ii][1] for ii in range(0, len(features_train)) if labels_train[ii]==0]
15  grade_slow = [features_train[ii][0] for ii in range(0, len(features_train)) if labels_train[ii]==1]
16  bumpy_slow = [features_train[ii][1] for ii in range(0, len(features_train)) if labels_train[ii]==1]
17
18
19  #### initial visualization
20  plt.xlim(0.0, 1.0)
21  plt.ylim(0.0, 1.0)
22  plt.scatter(bumpy_fast, grade_fast, color = "b", label="fast")
23  plt.scatter(grade_slow, bumpy_slow, color = "r", label="slow")
24  plt.legend()
25  plt.xlabel("bumpiness")
26  plt.ylabel("grade")
27  plt.show()
28  ################################################################################
29
30
31  ### your code here!  name your classifier object clf if you want the
32  ### visualization code (prettyPicture) to show you the decision boundary
33
34
35
36
37
38
39
40
41  try:
42      prettyPicture(clf, features_test, labels_test)
43  except NameError:
44      pass
```

File: ./choose_your_own/prep_terrain_data.py

```
1   #!/usr/bin/python
2   import random
3
4
5   def makeTerrainData(n_points=1000):
6   ###############################################################################
7   ### make the toy dataset
8       random.seed(42)
9       grade = [random.random() for ii in range(0,n_points)]
10      bumpy = [random.random() for ii in range(0,n_points)]
11      error = [random.random() for ii in range(0,n_points)]
12      y = [round(grade[ii]*bumpy[ii]+0.3+0.1*error[ii]) for ii in range(0,n_points)]
13      for ii in range(0, len(y)):
14          if grade[ii]>0.8 or bumpy[ii]>0.8:
15              y[ii] = 1.0
```

```
16
17 ### split into train/test sets
18     X = [[gg, ss] for gg, ss in zip(grade, bumpy)]
19     split = int(0.75*n_points)
20     X_train = X[0:split]
21     X_test  = X[split:]
22     y_train = y[0:split]
23     y_test  = y[split:]
24
25     grade_sig = [X_train[ii][0] for ii in range(0, len(X_train)) if y_train[ii]==0]
26     bumpy_sig = [X_train[ii][1] for ii in range(0, len(X_train)) if y_train[ii]==0]
27     grade_bkg = [X_train[ii][0] for ii in range(0, len(X_train)) if y_train[ii]==1]
28     bumpy_bkg = [X_train[ii][1] for ii in range(0, len(X_train)) if y_train[ii]==1]
29
30     training_data = {"fast":{"grade":grade_sig, "bumpiness":bumpy_sig}
31             , "slow":{"grade":grade_bkg, "bumpiness":bumpy_bkg}}
32
33
34     grade_sig = [X_test[ii][0] for ii in range(0, len(X_test)) if y_test[ii]==0]
35     bumpy_sig = [X_test[ii][1] for ii in range(0, len(X_test)) if y_test[ii]==0]
36     grade_bkg = [X_test[ii][0] for ii in range(0, len(X_test)) if y_test[ii]==1]
37     bumpy_bkg = [X_test[ii][1] for ii in range(0, len(X_test)) if y_test[ii]==1]
38
39     test_data = {"fast":{"grade":grade_sig, "bumpiness":bumpy_sig}
40             , "slow":{"grade":grade_bkg, "bumpiness":bumpy_bkg}}
41
42     return X_train, y_train, X_test, y_test
43
```

File: ./choose_your_own/class_vis.py

```
1   #!/usr/bin/python
2
3   import numpy as np
4   import matplotlib.pyplot as plt
5   import pylab as pl
6
7   def prettyPicture(clf, X_test, y_test):
8       x_min = 0.0; x_max = 1.0
9       y_min = 0.0; y_max = 1.0
10
11      # Plot the decision boundary. For that, we will assign a color to each
12      # point in the mesh [x_min, m_max]x[y_min, y_max].
13      h = .01  # step size in the mesh
14      xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
15      Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
16
17      # Put the result into a color plot
18      Z = Z.reshape(xx.shape)
19      plt.xlim(xx.min(), xx.max())
20      plt.ylim(yy.min(), yy.max())
21
22      plt.pcolormesh(xx, yy, Z, cmap=pl.cm.seismic)
23
24      # Plot also the test points
25      grade_sig = [X_test[ii][0] for ii in range(0, len(X_test)) if y_test[ii]==0]
26      bumpy_sig = [X_test[ii][1] for ii in range(0, len(X_test)) if y_test[ii]==0]
27      grade_bkg = [X_test[ii][0] for ii in range(0, len(X_test)) if y_test[ii]==1]
28      bumpy_bkg = [X_test[ii][1] for ii in range(0, len(X_test)) if y_test[ii]==1]
29
30      plt.scatter(grade_sig, bumpy_sig, color = "b", label="fast")
31      plt.scatter(grade_bkg, bumpy_bkg, color = "r", label="slow")
32      plt.legend()
33      plt.xlabel("bumpiness")
34      plt.ylabel("grade")
35
36      plt.savefig("test.png")
37
38  import base64
39  import json
40  import subprocess
41
42  def output_image(name, format, bytes):
43      image_start = "BEGIN_IMAGE_f9825uweof8jw9fj4r8"
44      image_end = "END_IMAGE_0238jfw08fjsiufhw8frs"
45      data = {}
46      data['name'] = name
47      data['format'] = format
48      data['bytes'] = base64.encodestring(bytes)
49      print image_start+json.dumps(data)+image_end
50
```