In [129]:
```
#import python and machine Learning Libraries
import os
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
from pandas import read_csv
from sklearn.model_selection import KFold
from sklearn.preprocessing import Normalizer
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import KFold
from pandas.plotting import scatter_matrix
from matplotlib import pyplot
from sklearn.model_selection import train_test_split
from sklearn.model_selection import KFold
from sklearn.externals import joblib
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

In [101]:
```
#import the csv file/dataset using pandas
import pandas as pd
df=pd.read_csv("HR_data.csv")
```

In [102]:
```
#print info with a few rows to get the feel of data along with
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 10 columns):
satisfaction_level       14999 non-null float64
last_evaluation          14999 non-null float64
number_project           14999 non-null int64
average_montly_hours     14999 non-null int64
time_spend_company       14999 non-null int64
Work_accident            14999 non-null int64
left                     14999 non-null int64
promotion_last_5years    14999 non-null int64
department               14999 non-null object
salary scale             14999 non-null object
dtypes: float64(2), int64(6), object(2)
memory usage: 1.1+ MB
```

In [103]:
```
#print the first 5 rows to get the feel of the dataset
df.head()
```

Out[103]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_compan
0	0.38	0.53	2	157	
1	0.80	0.86	5	262	
2	0.11	0.88	7	272	
3	0.72	0.87	5	223	
4	0.37	0.52	2	159	

In [104]:
```
#lets find the null values in the data set
df[df['department'].isnull()]
```

Out[104]:

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company

I am going to try to predict people who left and variables like satisfaction level, evaluation, salary scale, deparment, accidents and time spent will help me predict the people who left

In [105]:
```
dff=df[['satisfaction_level', 'department','left','salary scale','promotion_last_5years','Work_accident','number_project','last_evaluation','average_montly_hours']]
```

exploring features and converting categorical data to numerical values because machine learning models work best with numerical values as opposed to categorical values

Exploring salary scale

In [106]:
```
dff['salary scale'].unique()
```

Out[106]: array(['low', 'medium', 'high'], dtype=object)

In [107]:
```
lb_make = LabelEncoder()
dff["salary_code"] = lb_make.fit_transform(dff["salary scale"])
```

In [108]:
```
dff.head(5)
```

Out[108]:

	satisfaction_level	department	left	salary scale	promotion_last_5years	Work_accident	number_pr
0	0.38	sales	1	low	0	0	
1	0.80	sales	1	medium	0	0	
2	0.11	sales	1	medium	0	0	
3	0.72	sales	1	low	0	0	
4	0.37	sales	1	low	0	0	

exploring departments

In [109]:
```
dff['department'].unique()
```

Out[109]:
```
array(['sales', 'accounting', 'hr', 'technical', 'support', 'management',
       'IT', 'product_mng', 'marketing', 'RandD'], dtype=object)
```

In [110]:
```
lb_make = LabelEncoder()
dff["department_code"] = lb_make.fit_transform(dff["department"])
```

In [111]:
```
dff["department_code"].unique()
```

Out[111]:
```
array([7, 2, 3, 9, 8, 4, 0, 6, 5, 1], dtype=int64)
```

In [112]:
```
dff.head(10)
```

Out[112]:

	satisfaction_level	department	left	salary scale	promotion_last_5years	Work_accident	number_pr
0	0.38	sales	1	low	0	0	
1	0.80	sales	1	medium	0	0	
2	0.11	sales	1	medium	0	0	
3	0.72	sales	1	low	0	0	
4	0.37	sales	1	low	0	0	
5	0.41	sales	1	low	0	0	
6	0.10	sales	1	low	0	0	
7	0.92	sales	1	low	0	0	
8	0.89	sales	1	low	0	0	
9	0.42	sales	1	low	0	0	

since weve already converted categorical data to numerical data, we need to drop department and salary scale

In [113]:
```
#exploring the shape of the dataset
dff.shape
```

Out[113]:
```
(14999, 11)
```

In [114]:
```
# descriptions of the dataset showing all the variables
print(dff.describe())
```

```
       satisfaction_level          left  promotion_last_5years  Work_accident  \
count        14999.000000  14999.000000           14999.000000   14999.000000
mean             0.612834      0.238083               0.021268       0.144610
std              0.248631      0.425924               0.144281       0.351719
min              0.090000      0.000000               0.000000       0.000000
25%              0.440000      0.000000               0.000000       0.000000
50%              0.640000      0.000000               0.000000       0.000000
75%              0.820000      0.000000               0.000000       0.000000
max              1.000000      1.000000               1.000000       1.000000

       number_project  last_evaluation  average_montly_hours   salary_code  \
count    14999.000000     14999.000000          14999.000000  14999.000000
mean         3.803054         0.716102            201.050337      1.347290
std          1.232592         0.171169             49.943099      0.625819
min          2.000000         0.360000             96.000000      0.000000
25%          3.000000         0.560000            156.000000      1.000000
50%          4.000000         0.720000            200.000000      1.000000
75%          5.000000         0.870000            245.000000      2.000000
max          7.000000         1.000000            310.000000      2.000000

       department_code
count     14999.000000
mean          5.870525
std           2.868786
min           0.000000
25%           4.000000
50%           7.000000
75%           8.000000
max           9.000000
```

In [115]:
```
# class distribution
print(dff.groupby('department').size())
```

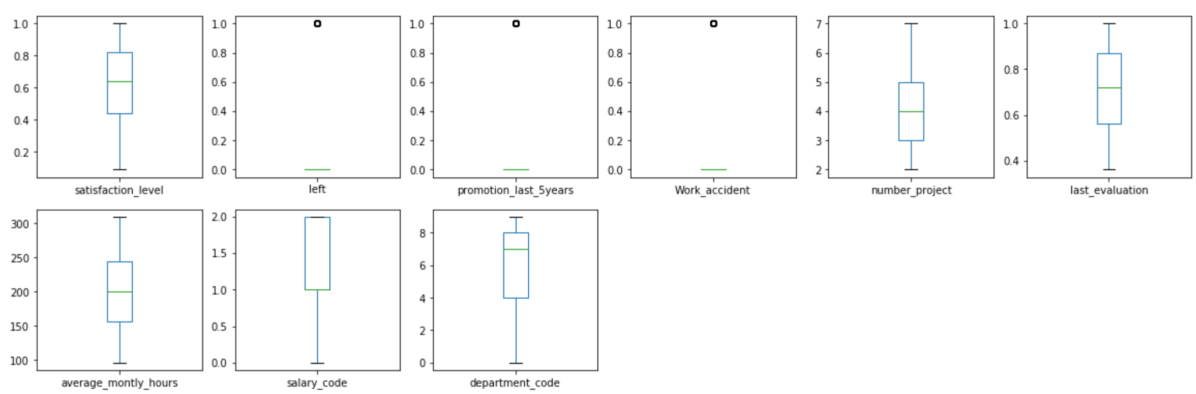
```
department
IT             1227
RandD           787
accounting      767
hr              739
management      630
marketing       858
product_mng     902
sales          4140
support        2229
technical      2720
dtype: int64
```

In [116]:
```
# class distribution
print(dff.groupby('salary scale').size())
```

```
salary scale
high      1237
low       7316
medium    6446
dtype: int64
```

Univariate Plots

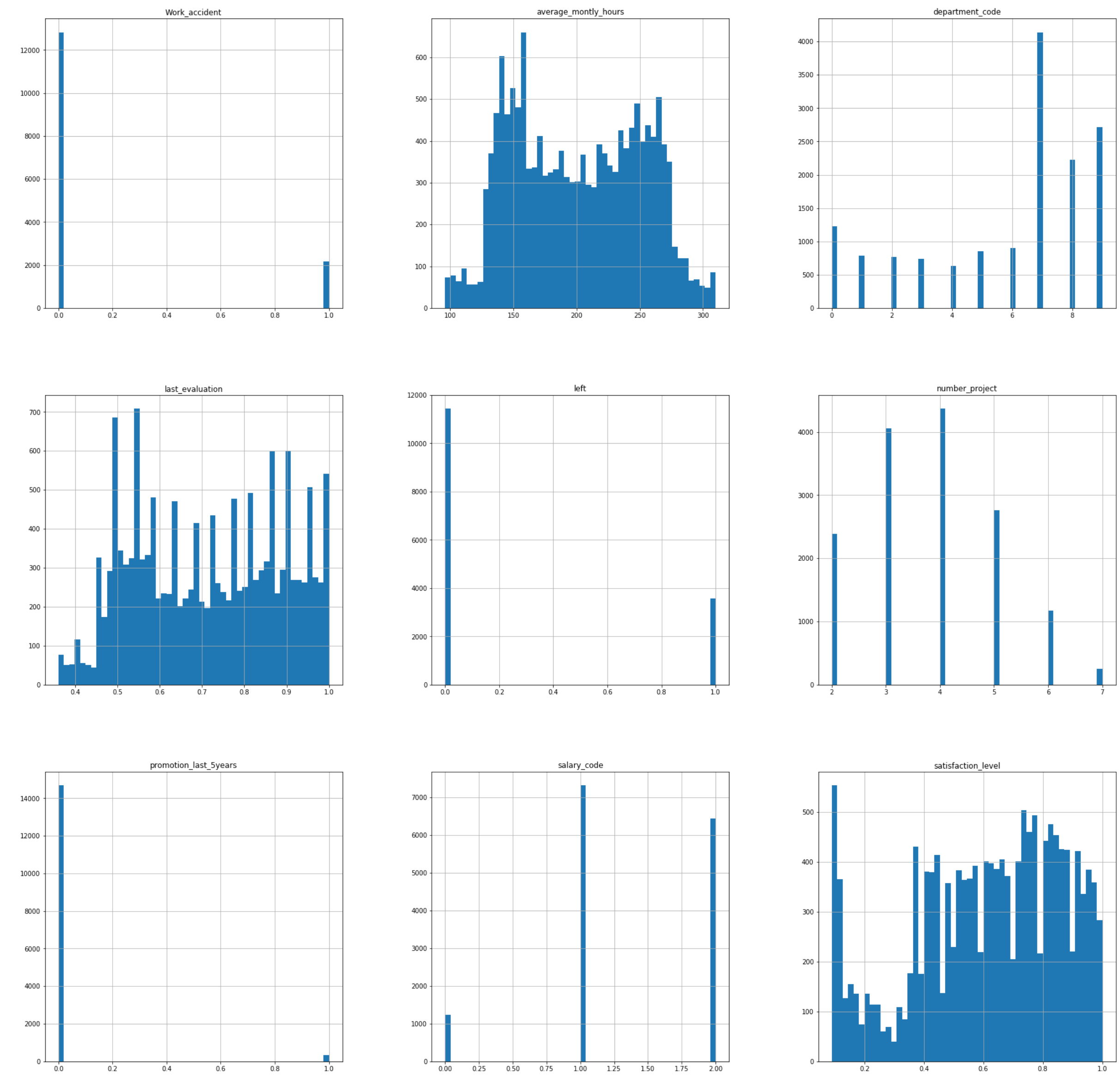
In [117]:
```
# box and whisker plots
dff.plot(kind='box',layout=(6,6), subplots=True, sharex=False, sharey=False,figsize=(20,20))
plt.show()
```



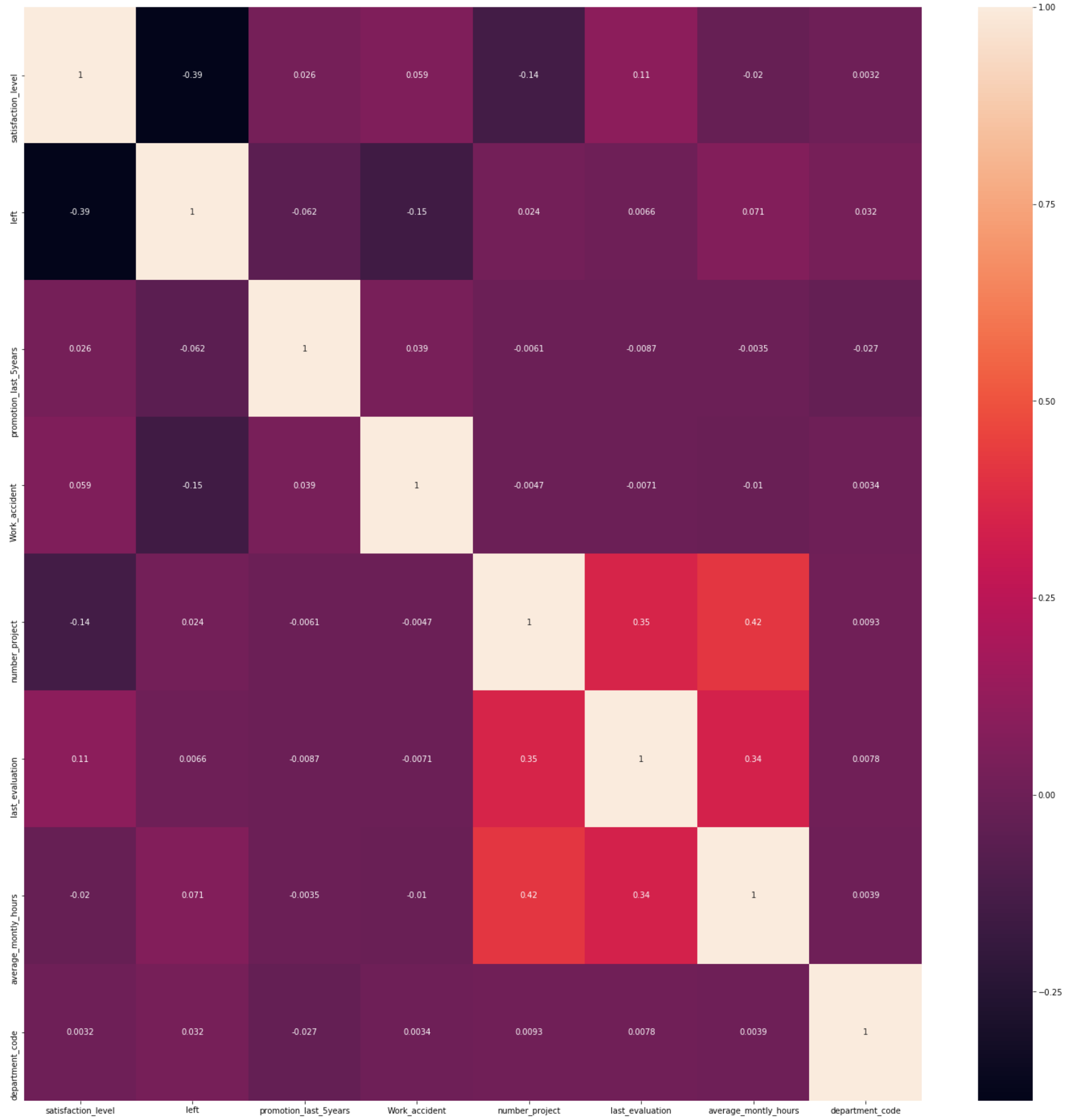
the box plot above tells that the dataset is not balanced that there's a variable with large values. the model will assign a higher value to that variable and yet the variable may not be that very importantthe that variable will skew our results

In [118]:

```
# histograms to see the distribution of variables in the dataset
dff.hist(bins=50,figsize=(30,30))
plt.show()
```



In [76]:
```python
#showing the correlation between variables
heat_map = dff.corr()
plt.figure(figsize=(25,25))
sns.heatmap(heat_map,annot=True,vmax=1)
plt.show()
```



In [119]:
```python
#dropping salary_scale and departments because weve already converted them to
 numerical values so we wont need them
#dff.drop(['department'],axis=1)
```

In [120]:
```
del dff['salary scale']
#dff.drop(['salary scale'],axis=1)
dff.columns
```

Out[120]:
```
Index(['satisfaction_level', 'department', 'left', 'promotion_last_5years',
       'Work_accident', 'number_project', 'last_evaluation',
       'average_montly_hours', 'salary_code', 'department_code'],
      dtype='object')
```

In [121]:
```
#del dff['department']
dff.columns
```

Out[121]:
```
Index(['satisfaction_level', 'department', 'left', 'promotion_last_5years',
       'Work_accident', 'number_project', 'last_evaluation',
       'average_montly_hours', 'salary_code', 'department_code'],
      dtype='object')
```

In [125]:
```
#The normalize function is running our dataset now
def normalize(dataset):
    dataNorm=((dataset-dataset.min())/(dataset.max()-dataset.min()))
    return dataNorm
```

In [ ]:
```
#dff=normalize(dff)
#dff.sample(5)
```

Trying out machine learning models

i am going to split the dataset into testing and training datasets and then tryout a few machine learning models

In [131]:
```
# Split-out validation dataset
array = dff.values
X = dff[['satisfaction_level', 'department_code','salary_code','promotion_last
_5years','Work_accident','number_project','last_evaluation','average_montly_ho
urs']]
Y = dff['left']
validation_size = 0.20
seed = 7
X_train, X_validation, Y_train, Y_validation = train_test_split(X, Y, test_siz
e=validation_size, random_state=seed)
```

In [132]:

```
# Spot-Check Algorithms
models = []
models.append(('LR', LogisticRegression()))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))
# evaluate each model in turn
results = []
names = []
for name, model in models:
    kfold = KFold(n_splits=10, random_state=seed)
    cv_results = cross_val_score(model, X_train, Y_train, cv=kfold, scoring='a
ccuracy')
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

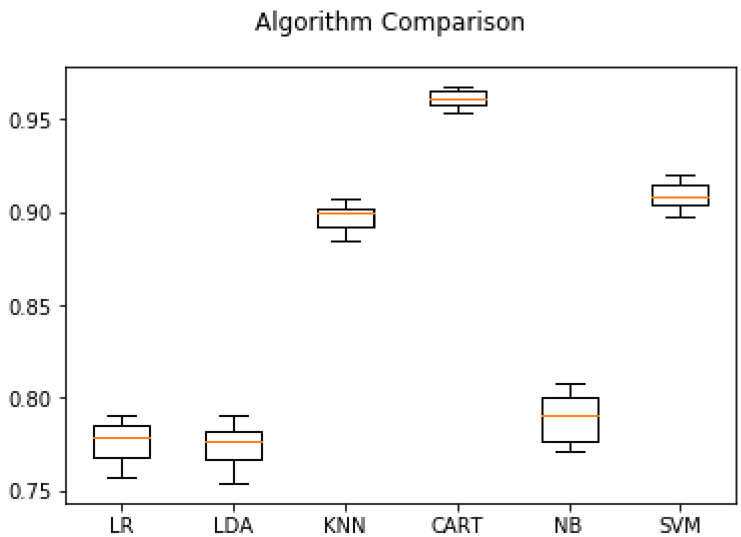
```
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\linear_model\logis
tic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22.
Specify a solver to silence this warning.
  FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\linear_model\logis
tic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22.
Specify a solver to silence this warning.
  FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\linear_model\logis
tic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22.
Specify a solver to silence this warning.
  FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\linear_model\logis
tic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22.
Specify a solver to silence this warning.
  FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\linear_model\logis
tic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22.
Specify a solver to silence this warning.
  FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\linear_model\logis
tic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22.
Specify a solver to silence this warning.
  FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\linear_model\logis
tic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22.
Specify a solver to silence this warning.
  FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\linear_model\logis
tic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22.
Specify a solver to silence this warning.
  FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\linear_model\logis
tic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22.
Specify a solver to silence this warning.
  FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\linear_model\logis
tic.py:433: FutureWarning: Default solver will be changed to 'lbfgs' in 0.22.
Specify a solver to silence this warning.
  FutureWarning)

LR: 0.776398 (0.012032)
LDA: 0.773231 (0.011692)
KNN: 0.897742 (0.006952)
CART: 0.961663 (0.004346)
NB: 0.789567 (0.013452)
```

```
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: F
utureWarning: The default value of gamma will change from 'auto' to 'scale' i
n version 0.22 to account better for unscaled features. Set gamma explicitly
to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: F
utureWarning: The default value of gamma will change from 'auto' to 'scale' i
n version 0.22 to account better for unscaled features. Set gamma explicitly
to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: F
utureWarning: The default value of gamma will change from 'auto' to 'scale' i
n version 0.22 to account better for unscaled features. Set gamma explicitly
to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: F
utureWarning: The default value of gamma will change from 'auto' to 'scale' i
n version 0.22 to account better for unscaled features. Set gamma explicitly
to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: F
utureWarning: The default value of gamma will change from 'auto' to 'scale' i
n version 0.22 to account better for unscaled features. Set gamma explicitly
to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: F
utureWarning: The default value of gamma will change from 'auto' to 'scale' i
n version 0.22 to account better for unscaled features. Set gamma explicitly
to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: F
utureWarning: The default value of gamma will change from 'auto' to 'scale' i
n version 0.22 to account better for unscaled features. Set gamma explicitly
to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: F
utureWarning: The default value of gamma will change from 'auto' to 'scale' i
n version 0.22 to account better for unscaled features. Set gamma explicitly
to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: F
utureWarning: The default value of gamma will change from 'auto' to 'scale' i
n version 0.22 to account better for unscaled features. Set gamma explicitly
to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
C:\Users\nakibedaAdmin\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: F
utureWarning: The default value of gamma will change from 'auto' to 'scale' i
n version 0.22 to account better for unscaled features. Set gamma explicitly
to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)

SVM: 0.909326 (0.007082)
```

In [133]:
```
# in the box plot below were going to compare the persormace our models. However the information above shows that CART has the highest accurate prediction
fig = pyplot.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
pyplot.boxplot(results)
ax.set_xticklabels(names)
pyplot.show()
```



In [134]:
```
# Make predictions on validation dataset
CART = DecisionTreeClassifier()
CART.fit(X_train, Y_train)
predictions = CART.predict(X_validation)
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))
```

```
0.9626666666666667
[[2248   69]
 [  43  640]]
              precision    recall  f1-score   support

           0       0.98      0.97      0.98      2317
           1       0.90      0.94      0.92       683

   micro avg       0.96      0.96      0.96      3000
   macro avg       0.94      0.95      0.95      3000
weighted avg       0.96      0.96      0.96      3000
```

In [135]:
```
# save the model to disk
filename = 'CART_model.sav'
joblib.dump(CART, filename)
```

Out[135]: ['CART_model.sav']