In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

%matplotlib inline
```

In [2]:

```
data = pd.read_csv("C:/Users/sainath/Desktop/ds/pima-indians-diabetes-database/diabetes.csv
```

In [3]:

```
data.shape
```

Out[3]:

```
(768, 9)
```

In [4]:

```
data.head(5)
```

Out[4]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28

In [5]:

```
# check if any null value is present
data.isnull().values.any()
```

Out[5]:

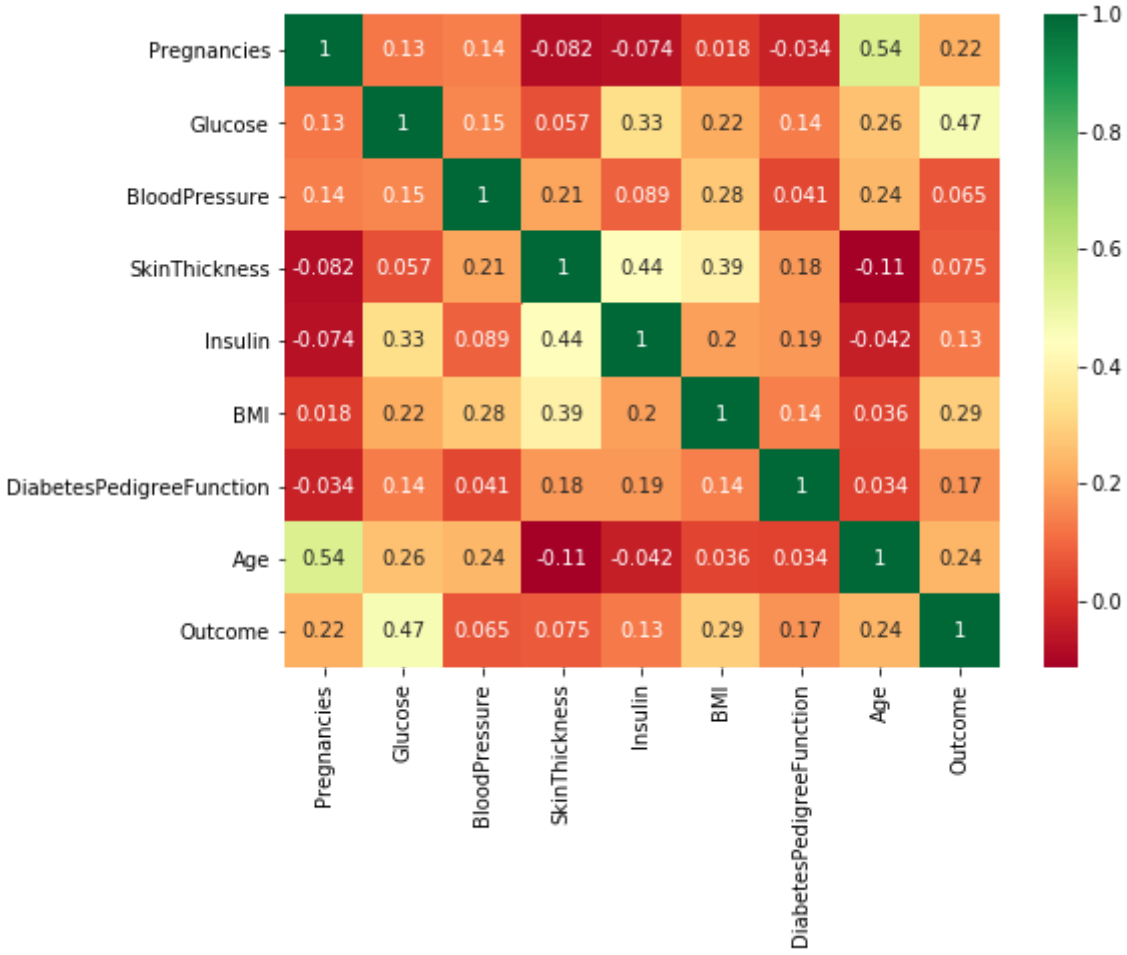
```
False
```

In [6]:

```
## Correlation
import seaborn as sns
import matplotlib.pyplot as plt
#get correlations of each features in dataset
corrmat = data.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(8,6))
#plot heat map
ax=sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="RdYlGn")
bottom, top = ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)
```

Out[6]:

```
(9.0, 0.0)
```



In [7]:

```
data.corr()
```

Out[7]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	
Pregnancies	1.000000	0.129459	0.141282	-0.081672	-0.073535	0.017
Glucose	0.129459	1.000000	0.152590	0.057328	0.331357	0.221
BloodPressure	0.141282	0.152590	1.000000	0.207371	0.088933	0.281
SkinThickness	-0.081672	0.057328	0.207371	1.000000	0.436783	0.392
Insulin	-0.073535	0.331357	0.088933	0.436783	1.000000	0.197
BMI	0.017683	0.221071	0.281805	0.392573	0.197859	1.000
DiabetesPedigreeFunction	-0.033523	0.137337	0.041265	0.183928	0.185071	0.140
Age	0.544341	0.263514	0.239528	-0.113970	-0.042163	0.036
Outcome	0.221898	0.466581	0.065068	0.074752	0.130548	0.292

In [8]:

```
data.index
```

Out[8]:

```
RangeIndex(start=0, stop=768, step=1)
```

In [9]:

```
corrmat.index
```

Out[9]:

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insuli
n',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [10]:

```
from sklearn.model_selection import train_test_split
feature_columns = ['Pregnancies',  'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
predicted_class = ['Outcome']
```

In [11]:

```
X = data[feature_columns].values
y = data[predicted_class].values


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30, random_state=10
```

In [12]:

```
from sklearn.ensemble import RandomForestClassifier
random_forest_model = RandomForestClassifier(random_state=10)

random_forest_model.fit(X_train, y_train.ravel())
```

```
C:\Users\sainath\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:245:
FutureWarning: The default value of n_estimators will change from 10 in vers
ion 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

Out[12]:

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                       max_depth=None, max_features='auto', max_leaf_nodes=N
one,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=10, verbos
e=0,
                       warm_start=False)
```

In [13]:

```
predict_train_data = random_forest_model.predict(X_test)
```

In [14]:

```
from sklearn import metrics

print("Accuracy = {0:.3f}".format(metrics.accuracy_score(y_test, predict_train_data)))
```

```
Accuracy = 0.745
```

In [15]:

```
params={
 "learning_rate"    : [0.05, 0.10, 0.15, 0.20, 0.25, 0.30 ] ,
 "max_depth"        : [ 3, 4, 5, 6, 8, 10, 12, 15],
 "min_child_weight" : [ 1, 3, 5, 7 ],
 "gamma"            : [ 0.0, 0.1, 0.2 , 0.3, 0.4 ],
 "colsample_bytree" : [ 0.3, 0.4, 0.5 , 0.7 ]

}
```

In [16]:

```
from sklearn.model_selection import RandomizedSearchCV
import xgboost
```

In [32]:

```
pip install xgboost
```

```
Collecting xgboost
  Downloading https://files.pythonhosted.org/packages/5e/49/b95c037b717b4cea
dc76b6e164603471225c27052d1611d5a2e832757945/xgboost-0.90-py2.py3-none-win_a
md64.whl (https://files.pythonhosted.org/packages/5e/49/b95c037b717b4ceadc76
b6e164603471225c27052d1611d5a2e832757945/xgboost-0.90-py2.py3-none-win_amd6
4.whl) (18.3MB)
Requirement already satisfied: scipy in c:\users\sainath\anaconda3\lib\site-
packages (from xgboost) (1.3.1)
Requirement already satisfied: numpy in c:\users\sainath\anaconda3\lib\site-
packages (from xgboost) (1.16.5)
Installing collected packages: xgboost
Successfully installed xgboost-0.90
Note: you may need to restart the kernel to use updated packages.

  WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None,
status=None)) after connection broken by 'ReadTimeoutError("HTTPSConnectionP
ool(host='files.pythonhosted.org', port=443): Read timed out. (read timeout=
15)")': /packages/5e/49/b95c037b717b4ceadc76b6e164603471225c27052d1611d5a2e8
32757945/xgboost-0.90-py2.py3-none-win_amd64.whl
  WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None,
status=None)) after connection broken by 'ReadTimeoutError("HTTPSConnectionP
ool(host='files.pythonhosted.org', port=443): Read timed out. (read timeout=
15)")': /packages/5e/49/b95c037b717b4ceadc76b6e164603471225c27052d1611d5a2e8
32757945/xgboost-0.90-py2.py3-none-win_amd64.whl
  WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None,
status=None)) after connection broken by 'ReadTimeoutError("HTTPSConnectionP
ool(host='files.pythonhosted.org', port=443): Read timed out. (read timeout=
15)")': /packages/5e/49/b95c037b717b4ceadc76b6e164603471225c27052d1611d5a2e8
32757945/xgboost-0.90-py2.py3-none-win_amd64.whl
  WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None,
status=None)) after connection broken by 'ProtocolError('Connection aborte
d.', OSError(0, 'Error'))': /packages/5e/49/b95c037b717b4ceadc76b6e164603471
225c27052d1611d5a2e832757945/xgboost-0.90-py2.py3-none-win_amd64.whl
  WARNING: Retrying (Retry(total=0, connect=None, read=None, redirect=None,
status=None)) after connection broken by 'ReadTimeoutError("HTTPSConnectionP
ool(host='files.pythonhosted.org', port=443): Read timed out. (read timeout=
15)")': /packages/5e/49/b95c037b717b4ceadc76b6e164603471225c27052d1611d5a2e8
32757945/xgboost-0.90-py2.py3-none-win_amd64.whl
```

In [17]:

```
classifier=xgboost.XGBClassifier()
```

In [18]:

```
random_search=RandomizedSearchCV(classifier,param_distributions=params,n_iter=5,scoring='ro
```

In [19]:

```
def timer(start_time=None):
    if not start_time:
        start_time = datetime.now()
        return start_time
    elif start_time:
        thour, temp_sec = divmod((datetime.now() - start_time).total_seconds(), 3600)
        tmin, tsec = divmod(temp_sec, 60)
        print('\n Time taken: %i hours %i minutes and %s seconds.' % (thour, tmin, round(ts
```

In [ ]:

```
from datetime import datetime
# Here we go
start_time = timer(None) # timing starts from this point for "start_time" variable
random_search.fit(X,y.ravel())
timer(start_time)
```

In [20]:

```
from datetime import datetime
# Here we go
start_time = timer(None) # timing starts from this point for "start_time" variable
random_search.fit(X_train, y_train.ravel())
timer(start_time)
```

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 8 concurrent workers.
[Parallel(n_jobs=-1)]: Done  19 out of  25 | elapsed:    6.3s remaining:
1.9s
[Parallel(n_jobs=-1)]: Done  25 out of  25 | elapsed:    6.3s finished


 Time taken: 0 hours 0 minutes and 6.75 seconds.
```

In [21]:

```
random_search.best_estimator_
```

Out[21]:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.7, gamma=0.0,
              learning_rate=0.1, max_delta_step=0, max_depth=3,
              min_child_weight=7, missing=None, n_estimators=100, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=1, verbosity=1)
```

In [22]:

```
classifier=xgboost.XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
       colsample_bytree=0.3, gamma=0.0, learning_rate=0.25,
       max_delta_step=0, max_depth=3, min_child_weight=7, missing=None,
       n_estimators=100, n_jobs=1, nthread=None,
       objective='binary:logistic', random_state=0, reg_alpha=0,
       reg_lambda=1, scale_pos_weight=1, seed=None, silent=True,
       subsample=1)
```

In [23]:

```
from sklearn.model_selection import cross_val_score
score=cross_val_score(classifier,X_train,y_train.ravel(),cv=10)
```

In [24]:

```
score
```

Out[24]:

```
array([0.67272727, 0.74074074, 0.7962963 , 0.77777778, 0.68518519,
       0.7037037 , 0.79245283, 0.71698113, 0.73584906, 0.77358491])
```

In [25]:

```
score.mean()
```

Out[25]:

```
0.7395298900959277
```

In [27]:

```
classifier.fit(X_train,y_train)
```

```
C:\Users\sainath\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:
219: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using r
avel().
  y = column_or_1d(y, warn=True)
C:\Users\sainath\Anaconda3\lib\site-packages\sklearn\preprocessing\label.py:
252: DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using r
avel().
  y = column_or_1d(y, warn=True)
```

Out[27]:

```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=0.3, gamma=0.0,
              learning_rate=0.25, max_delta_step=0, max_depth=3,
              min_child_weight=7, missing=None, n_estimators=100, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=True, subsample=1, verbosity=1)
```

In [28]:

```
y_pred =classifier.predict(X_test)
```

In [29]:

```
y_pred
```

Out[29]:

```
array([1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1,
       1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1,
       1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1,
       0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0], dtype=int64)
```

In [71]:

```
from sklearn.metrics import confusion_matrix,accuracy_score

cm=confusion_matrix(y_test,y_pred)
score=accuracy_score(y_test,y_pred)

print(cm)
print(score)
```

```
[[120  24]
 [ 40  47]]
0.7229437229437229
```

In [ ]:

```

```