In [32]:
```
import numpy as np
import pandas as pd
import random
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import make_pipeline
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn import svm
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
filename = 'train.csv'
n = sum(1 for line in open(filename)) - 1
s = 100000
skip = sorted(random.sample(range(1, n+1), n-s))
df = pd.read_csv(filename, skiprows=skip)
```

In [3]:
```
dest = pd.read_csv('destinations.csv')
df.shape
```

Out[3]:
```
(100000, 24)
```

In [4]:
```
df.to_csv(r'C:\Users\Gabe\Documents\Bellevue University\Predictive Analytics\Week 5\subset_train.csv', index = False)
```

CSV was exported to perform EDA and Data Preparation in R

The data prepared CSV will be use for the modeling and algorithms

In [5]:
```
df = pd.read_csv('Merged_Prepared_Train.csv')
```

In [7]:
```
df = df.drop(columns='Unnamed: 0')
df.head()
```

Out[7]:

	date_year	date_month	site_name	posa_continent	user_location_country	user_location_region	user_locatic
0	2014	7	11	3	205	354	
1	2014	11	2	3	66	462	
2	2014	6	2	3	66	314	
3	2013	1	2	3	66	174	
4	2013	10	2	3	66	149	

5 rows × 176 columns

In [30]:
```
relevant_hotel_info = [df.groupby(['srch_destination_id','hotel_country','hotel_market','hotel_cluster'])['is_booking'].agg(['sum','count'])]
agg = pd.concat(relevant_hotel_info).groupby(level=[0,1,2,3]).sum()
#agg.dropna(inplace=True)
agg.head()
```

Out[30]:

				sum	count
srch_destination_id	hotel_country	hotel_market	hotel_cluster		
148	50	953	42	1	1
245	50	365	25	1	1
259	50	444	15	1	1
263	50	455	16	1	1
305	50	453	77	1	1

In [21]:
```
agg['sum_and_cnt'] = 0.85*agg['sum'] + 0.15*agg['count']
agg = agg.groupby(level=[0,1,2]).apply(lambda x: x.astype(float)/x.sum())
agg.reset_index(inplace=True)
agg.head()
```

Out[21]:

	srch_destination_id	hotel_country	hotel_market	hotel_cluster	sum	count	sum_and_cnt
0	148	50	953	42	1.0	1.0	1.0
1	245	50	365	25	1.0	1.0	1.0
2	259	50	444	15	1.0	1.0	1.0
3	263	50	455	16	1.0	1.0	1.0
4	305	50	453	77	1.0	1.0	1.0

In [22]:
```
agg_pivot = agg.pivot_table(index=['srch_destination_id','hotel_country','hotel_market'], columns='hotel_cluster', values='sum_and_cnt').reset_index()
agg_pivot.head()
```

Out[22]:

hotel_cluster	srch_destination_id	hotel_country	hotel_market	0	1	2	3	4	5	6	...	9
0	148	50	953	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	Na
1	245	50	365	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	Na
2	259	50	444	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	Na
3	263	50	455	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	Na
4	305	50	453	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	Na

5 rows × 98 columns

In [23]:
```
df = pd.merge(df, agg_pivot, how='left', on=['srch_destination_id','hotel_country','hotel_market'])
df.fillna(0, inplace=True)
df.shape
```

Out[23]: (542, 271)

In [24]:
```
df = df.loc[df['is_booking'] == 1]

# Determing features
X = df.drop(['user_id', 'hotel_cluster', 'is_booking'], axis=1)
y = df.hotel_cluster
```

In [33]:
```
# K-Nearest Neighbor Classifier
from sklearn.neighbors import KNeighborsClassifier
clf = make_pipeline(preprocessing.StandardScaler(), KNeighborsClassifier(n_neighbor
s=5))
np.mean(cross_val_score(clf, X, y, cv=10, scoring='accuracy'))
```

Out[33]: 0.611901065651586

In [34]:
```
# Random Forest Classifier
clf = make_pipeline(preprocessing.StandardScaler(), RandomForestClassifier(n_estima
tors=273,max_depth=10,random_state=0))
np.mean(cross_val_score(clf, X, y, cv=10))
```

Out[34]: 0.5691095486128402

In [35]:
```
# Multi-Class Logistic Regression
from sklearn.linear_model import LogisticRegression
clf = make_pipeline(preprocessing.StandardScaler(), LogisticRegression(multi_class=
'ovr'))
np.mean(cross_val_score(clf, X, y, cv=10))
```

Out[35]: 0.6501366019042722

In [36]:
```
# SVM Classifier
from sklearn import svm
clf = make_pipeline(preprocessing.StandardScaler(), svm.SVC(decision_function_shape
='ovo'))
np.mean(cross_val_score(clf, X, y, cv=10))
```

Out[36]: 0.5875114694032719