HASHTAG

Code:

```
"""We have done few changes to the features of(some dropped) trainset and xtest files according to the one hot
   encoding and correlation heatmap. Therefore this code creates the submission file according to exsiting
   trainset and xtest file under random forest model.
"""

from sklearn.ensemble import RandomForestClassifier

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

#Trainset = pd.read_csv("/kaggle/input/hackstat2k19/Trainset.csv", header = 0)
Trainset = pd.read_csv("E:\\jup\\Trainset.csv", header = 0)
Trainset = Trainset.dropna()

sampleset = pd.read_csv("E:\\jup\\sample_submisison.csv", header = 0)
sampleset = sampleset.dropna()

xset = pd.read_csv("E:\\jup\\xtest.csv")
xset = xset.dropna()

#Trainset['is_train'] = np.random.uniform(0,1,len(Trainset)) <= 0.75
print(Trainset)

Train, Test = Trainset,xset

print('Number of Training examples', len(Train))
print('Number of Testing examples', len(Test))
```

```
features = Trainset.columns[:30]
print(features)
y = Train['Revenue']
y

id_ = sampleset['ID']
#new_ = id_.DataFarame(id_)
clf = RandomForestClassifier(n_estimators = 30, n_jobs = 2, random_state = 0)
clf.fit(Train[features],y)

preds = clf.predict(Test[features])

#print(tpe(new_))
#print(type(preds))
YArray= id_.as_matrix(columns=None)
print (YArray)

df = pd.DataFrame({"ID" : YArray, "Revenue" : preds})
df.to_csv("E:\\jup\\submission.csv", index=False)
#dfObj = pd.DataFrame(YArray,preds,columns = ["ID","Revenue"])

#dfObj.to_csv('submit.csv', index = False)

#pd.crosstab(Test['Revenue'], preds, rownames = ['Actual Revenue'], colnames = ['Predicted Revenue'])
#Test['Revenue']
#X, y = Trainset.iloc[:, :16], Trainset.iloc[:, 17]

#Xset = xset.iloc[:, :16]
#print(X)
```

```
#print(y)

#clf = RandomForestClassifier(n_estimators=10)
#clf = clf.fit(X, y)
#clf

#clf = RandomForestClassifier(n_estimators=17, max_depth=None,min_samples_split=2, random_state=0)
#scores = cross_val_score(clf, X, y, cv=5)
#scores.mean()
```

One hot encoding:

```
####This code does the one hot encoding for five existing features.
import pandas as pd
df = pd.read_csv("E:\\jup\\Trainset.csv")

dummies = pd.get_dummies(df.Month)
merged1 = pd.concat([df,dummies],axis='columns')
final1 = merged1.drop(['Month'],axis='columns')


dummies_province = pd.get_dummies(final1.Province)
#dummies_province_dropped = dummies_province.drop([9],axis='columns')
merged2 = pd.concat([final1,dummies_province_dropped],axis='columns')
final2 = merged2.drop(['Province'],axis='columns')


dummies_browser = pd.get_dummies(final2.Browser)
#dummies_browser_dropped = dummies_browser.drop([1],axis='columns')
merged3 = pd.concat([final2,dummies_browser_dropped],axis='columns')
final3 = merged3.drop(['Browser'],axis='columns')


dummies_operatingSystems = pd.get_dummies(final3.OperatingSystems)
#dummies_operatingSystems_dropped = dummies_operatingSystems.drop([1],axis='columns')
merged4 = pd.concat([final3,dummies_operatingSystems_dropped],axis='columns')
final4 = merged4.drop(['OperatingSystems'],axis='columns')

dummies_VisitorType = pd.get_dummies(final4.VisitorType)
#dummies_VisitorType_dropped = dummies_VisitorType.drop(['Other'],axis='columns')
merged5 = pd.concat([final4,dummies_VisitorType_dropped],axis='columns')
```

```
final5 = merged5.drop(['VisitorType'],axis='columns')


print(final5.shape)



f=pd.DataFrame(final5)
f.to_csv('E:\\jup\\sample_Traintest.csv',index=False,)
```

Correlation map:

```
# calculate the correlation matrix
corr = Trainset.corr()


plt.figure(figsize=(40,40))
# plot the heatmap
sns.heatmap(corr,
        xticklabels=corr.columns,
        yticklabels=corr.columns,vmin = -1,vmax =1 ,center = 0,cmap = sns.diverging_palette(-400,220,n=200),square = True)
```