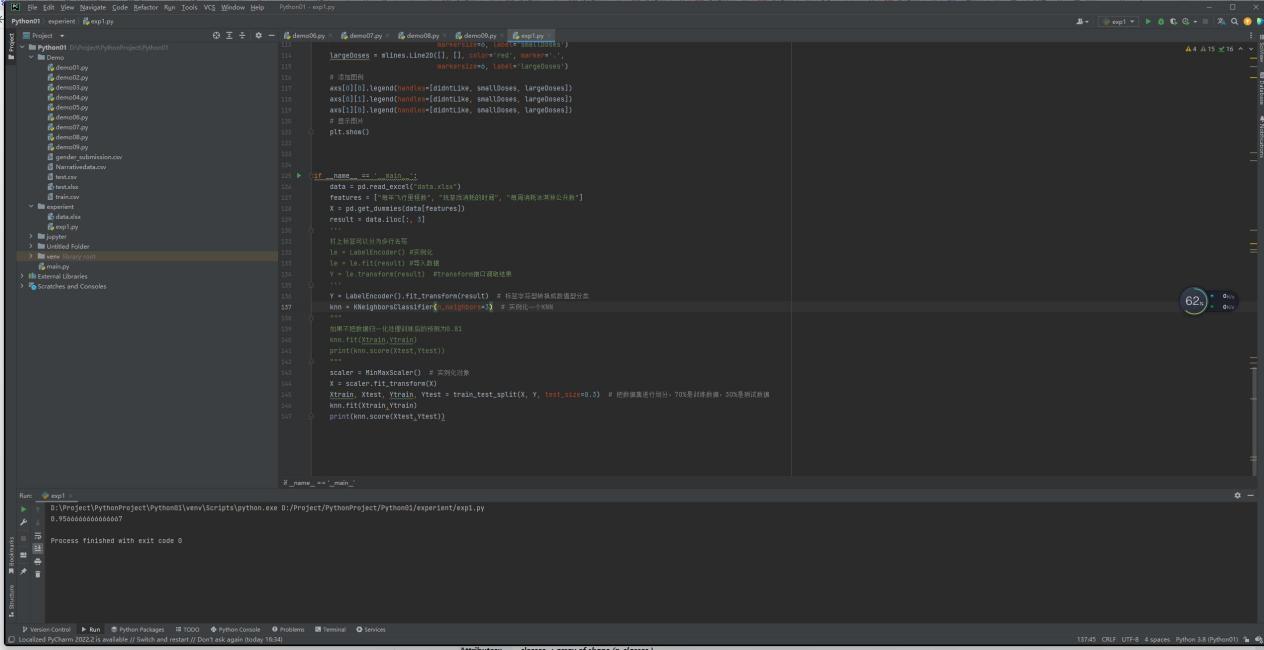
**问题一：约会问题有三个，k=3每一个近邻如何分类？**

当k=3的时候，KNN算法会选择类别中距离（欧式距离）最小的三个点，以此判断这个哪一类。

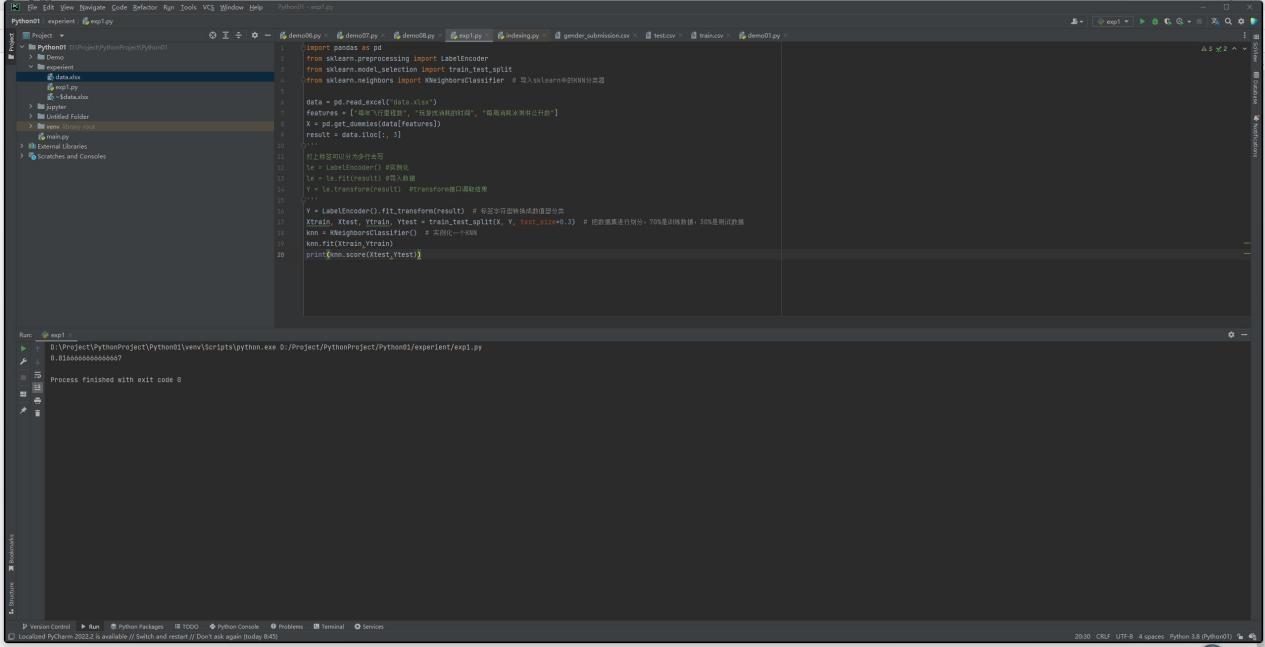
****

**源代码：**

data = pd.read\_excel("data.xlsx")  
features = ["每年飞行里程数", "玩游戏消耗的时间", "每周消耗冰淇淋公升数"]  
X = pd.get\_dummies(data[features])  
result = data.iloc[:, 3]  
'''  
打上标签可以分为多行去写  
le = LabelEncoder() #实例化  
le = le.fit(result) #导入数据  
Y = le.transform(result) #transform接口调取结果  
'''  
Y = LabelEncoder().fit\_transform(result) # 标签字符型转换成数值型分类  
knn = KNeighborsClassifier(n\_neighbors=3) # 实例化一个KNN  
"""  
如果不把数据归一化处理训练后的预测为0.81  
knn.fit(Xtrain,Ytrain)  
print(knn.score(Xtest,Ytest))  
"""  
scaler = MinMaxScaler() # 实例化对象  
X = scaler.fit\_transform(X)  
Xtrain, Xtest, Ytrain, Ytest = train\_test\_split(X, Y, test\_size=0.3) # 把数据集进行划分，70%是训练数据，30%是测试数据  
knn.fit(Xtrain, Ytrain)  
showdatas(Xtrain,Ytrain)

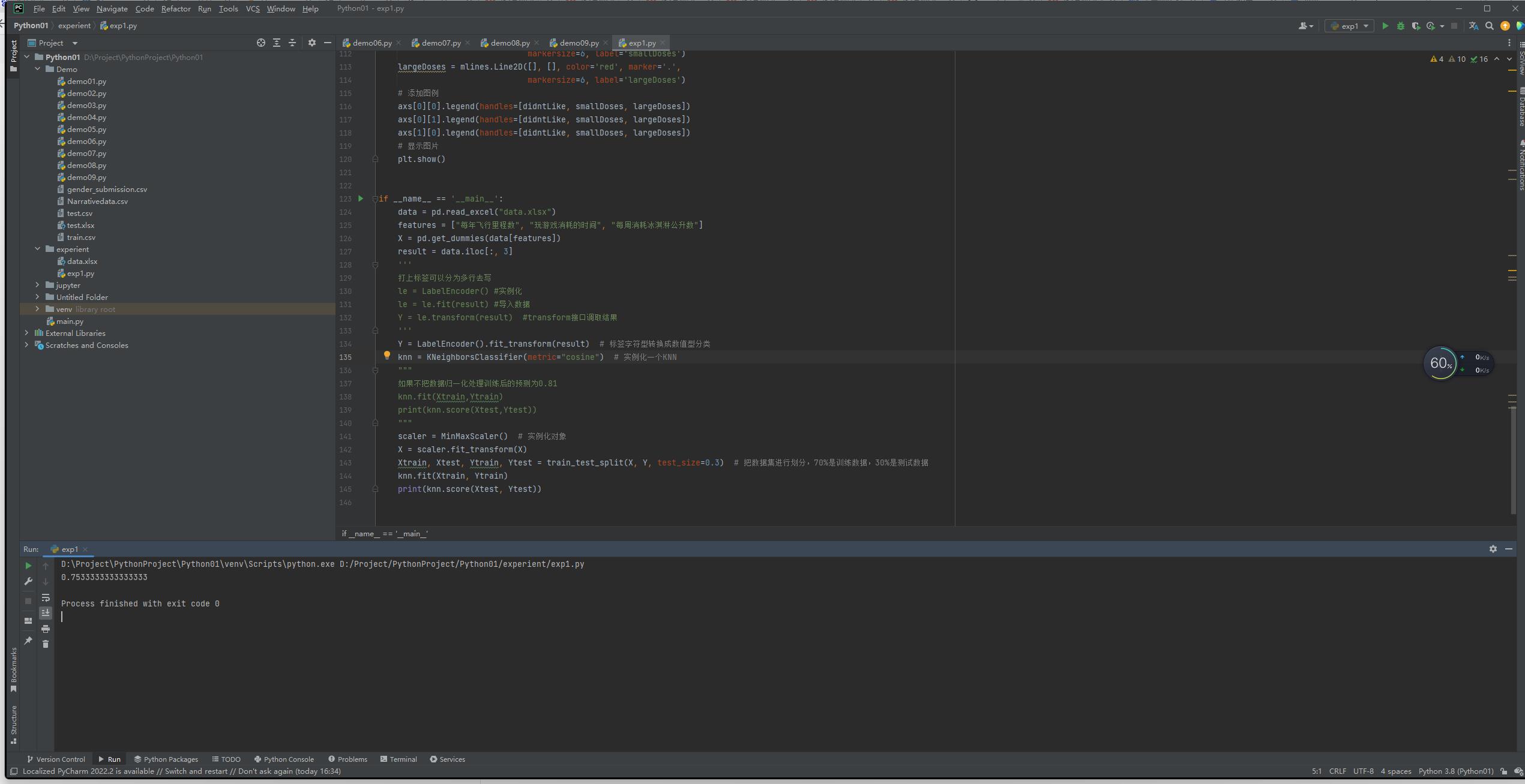
**问题二：约会问题如果不进行数据归一化处理结果会如何？**

数据预测结果很差只有0.81，可以看出数值大的特征对数值小的特征在计算欧式距离的时候有很大的影响因此需要进行归一化处理。

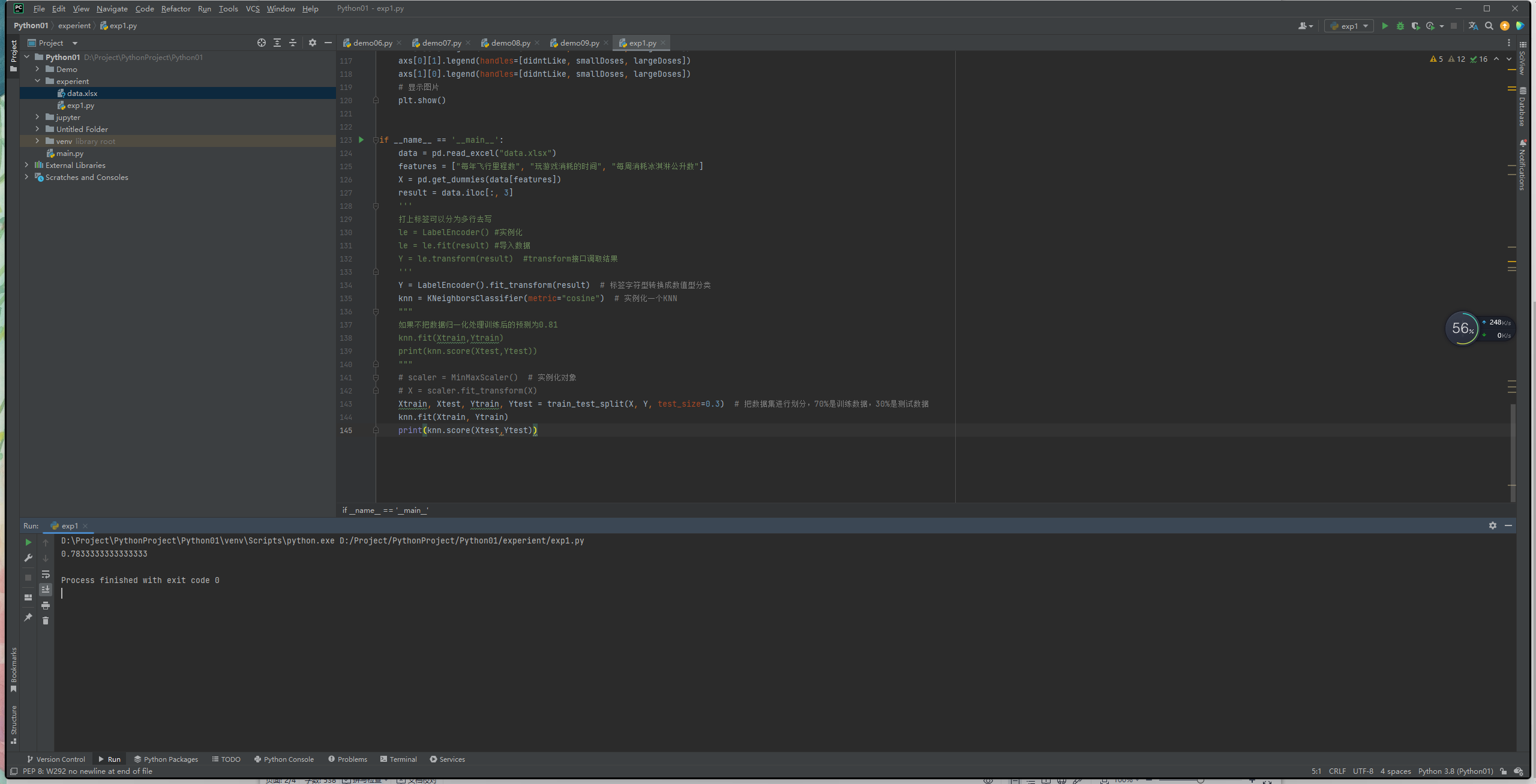
****

**问题三：如果把欧式距离替换成余弦距离结果如何？是否还需要进行归一化处理？**

如果把欧式距离替换成余弦距离（数据归一化）可以看出结果只有0.75，预测结果显然没有达到理想状态，因为余弦距离只能分析两个向量相似度，在knn中我们在分类的时候是根据欧式距离去判断的，距离越近代表特别越相似。余弦距离只在方向上是否相似，并不表现在特征上。

****

不进行归一化之后发现只有0,78，比归一化更高一点，说明在计算余弦距离的时候跟数据大小不相关，只和方向有关。所以不需要归一化处理了。



**源代码：**

data = pd.read\_excel("data.xlsx")  
features = ["每年飞行里程数", "玩游戏消耗的时间", "每周消耗冰淇淋公升数"]  
X = pd.get\_dummies(data[features])  
result = data.iloc[:, 3]  
'''  
打上标签可以分为多行去写  
le = LabelEncoder() #实例化  
le = le.fit(result) #导入数据  
Y = le.transform(result) #transform接口调取结果  
'''  
Y = LabelEncoder().fit\_transform(result) # 标签字符型转换成数值型分类  
knn = KNeighborsClassifier(metric="cosine") # 实例化一个KNN  
"""  
如果不把数据归一化处理训练后的预测为0.81  
knn.fit(Xtrain,Ytrain)  
print(knn.score(Xtest,Ytest))  
"""  
# scaler = MinMaxScaler() # 实例化对象  
# X = scaler.fit\_transform(X)  
Xtrain, Xtest, Ytrain, Ytest = train\_test\_split(X, Y, test\_size=0.3) # 把数据集进行划分，70%是训练数据，30%是测试数据  
knn.fit(Xtrain, Ytrain)  
print(knn.score(Xtest, Ytest))