B站RFM用户分析 + KMeans聚类模型

分析报告内容如下：

- 实现目标
- 分析思路
    - IFL模型
    - 维度打分标准
- 数据处理与分析
    - 数据探索
    - IFL模型构建
    - K-Means建模
- 运营策略优化

一、实现目标

此次分析将采用视频社区产品B站的视频播放行为数据作为数据集，参考经典用户价值分层模型RMF模型，变种出一套适用于社区调性产品的模型IFL模型来评估视频的质量，依托K-Means聚类完成对up主的分群，并分析每类的up主的特征，从而制定不同的运营策略。

二、分析思路

IFL模型

IFL模型将从以下三个维度来评估视频的质量。

I(Interaction_rate)：
I值反映的是平均每个视频的互动率，互动率越高，表明其视频更能产生用户的共鸣，使其有话题感。
I=(总弹幕数+总评论数)/总播放量/发布视频数量

F(Frequence)：
F值表示的是每个视频的平均发布周期，每个视频之间的发布周期越短，说明内容生产者创作视频的时间也就越短，创作时间太长，不是忠实粉丝的用户可能将其遗忘。
F=(统计范围内最晚发布视频时间-最早发布视频时间)/发布视频的数量

L(Like_rate)：
L值表示的是统计时间内发布视频的平均点赞率，越大表示视频质量越稳定，用户对up主的认可度也就越高。
L=(点赞数X1+投币数X2+收藏数X3+分享数X4)/播放量/发布视频数

维度打分标准

维度确认的核心是分值确定，按照设定的标准，我们给每个消费者的I/F/L值打分，分值的大小取决于我们的偏好， 即我们越喜欢，打的分数就越高：

- I值，I代表了up主视频的平均评论率，这个值越大，就说明其视频越能使用户有话题，当I值越大时，分值越大。
- F值表示视频的平均发布周期，我们当然想要经常看到，所以这个值越大时，分值越小。
- L值表示发布视频的平均点赞率，L值越大时，质量越稳定，分值也就越大。

三、数据处理与分析

1 数据探索

本次数据集基于 bilibili 网站上的公开信息，主要采集了以下数据维度：

2019年1月~2020年3月，科技区播放量过5w视频的分区名称、作者名称、作者id、发布时间、播放数、硬币数、弹幕数、收藏数、点赞数、分享数、评论数，一共50130行。

1.1读取数据

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
from mpl_toolkits.mplot3d import Axes3D
"""设置中文"""
plt.rcParams["font.family"] = ['Arial Unicode MS']  # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus']=False  #解决保存图像是负号'-'显示为方块的问题

from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans

%matplotlib inline
sns.set()


data = pd.read_excel(u'./b站分析数据.xlsx')
# data.head(1)
data.sample(5)
```

1.2 数据探索

初步数据探索结果：

- 存在少量重复值、缺失值和异常值，数量很少做删除处理
- 需要进行日期类型转化

```
data.info()

data["分区"].value_counts()

data.describe().T

#查看缺失值比例
data.isnull().mean()
```

1.3 数据预处理

数据预处理包括空值处理、重复值处理、异常值处理，数据变换等

```
#删除含有缺失值的行
data.dropna(inplace=True)
#查看缺失值
data.isnull().sum()
data.drop_duplicates(keep="first",inplace=True)

#恢复索引
data.index = range(data.shape[0])

# 变换日期类型
data['date'] = pd.to_datetime(data['date'], format='%Y-%m-%d %H:%M')
```

2 IFL模型构建

按照以下计算规则构建特征值，经过特征归一化后采用K-Means聚类方法对up主进行聚类。

I=(总弹幕数+总评论数)/总播放量/统计范围内视频数量

F=(统计范围内最晚发布视频时间-最早发布视频时间)/发布视频的数量

L=(点赞数X1+投币数X2+收藏数X3+分享数X4)/播放量/发布视频数

2.1 构建特征值

进行特征选择完成特征数据准备

```
#进行特征选择完成特征数据准备
IFL_I = data.groupby(['author'],as_index = False).agg(
 {'bv':'count','danmu':'sum','comment':'sum','share':'sum','view':'sum'}).renam
e(columns={"bv": "bv_count","danmu": "danmus","comment":
"comments","share":"shares","view":"views"})


IFL_F = data.groupby('author')
['date'].agg([np.min,np.max]).reset_index().rename(columns={"amin": "date_min",
"amax": "date_max"})


data['L'] = (data['like'] + data['coin']*2 + data['favorite']*3 +
data['share']*4) / data['view']
IFL_L = data.groupby('author')['L'].agg(np.sum).reset_index()

IFL = pd.merge(I_F,IFL_L, on='author', how='inner')
IFL.head()

IFL['I'] = (IFL['danmus']+IFL['comments'])/IFL['views']/IFL['bv_count']
IFL['F'] = (IFL['date_max']-IFL['date_min']).dt.days/IFL['bv_count']
IFL['L'] = IFL['L']/IFL['bv_count']

IFL_final = IFL[['author', 'I','F','L']]
```

2.2 数据标准化

```
df_ifl=IFL_final.drop(columns=['author'])
X = MinMaxScaler().fit_transform(df_ifl)
df_ifl_scaled = pd.DataFrame(data=X, columns=list(df_ifl.columns))

df_ifl_scaled.head()
```

可视化查看数据分布

```
# visualize the distribution of "Interaction_rate", "Frequency", and
"Like_rate"
fig,(ax1,ax2,ax3)=plt.subplots(1,3,figsize=(10,4))
```

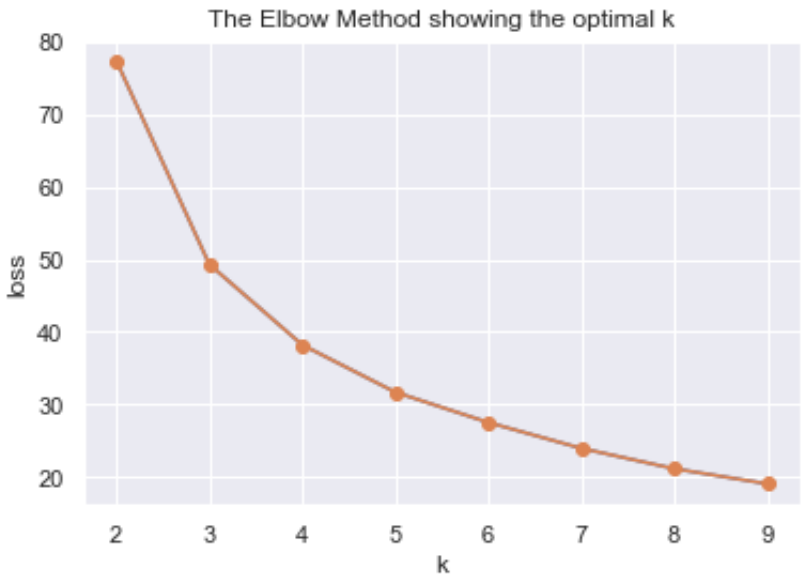
```
# plot "Interaction_rate"
ax1.hist(df_ifl_scaled['I'],bins=10,color='lightsteelblue')
ax1.set_xticks(np.arange(0,5,1))
ax1.set_xlabel('Interaction_rate')
ax1.set_ylabel('number of author')

# plot "Frequency"
ax2.hist(df_ifl_scaled['F'],bins=10,color='lightsteelblue')
ax2.set_xlabel('frequency')


# plot "Like_rate"
ax3.hist(df_ifl_scaled['L'],bins=10,color='lightsteelblue')
ax3.set_xlabel('Like_rate')


plt.tight_layout()

df_ifl_scaled.describe().T
```

3 K-Means建模

3.1 建立模型

采用肘部法则求解最佳聚类个数K（轮廓系数法辅助验证）

```
# 肘部法则--求解合适的K
loss = []
K = range(2,10)
for i in K:
    model = KMeans(n_clusters=i,random_state=99).fit(df_ifl_scaled)
    loss.append(model.inertia_)

plt.plot(K,loss)
plt.xlabel('k')
plt.ylabel('loss')
plt.title('The Elbow Method showing the optimal k')
plt.plot(K,loss,'o-')
plt.show()
```



结论：

从图上可以看出K值为3的时候下降率变缓出现"肘点"，即认为是最佳K值。

以下用轮廓系数法再次验证

```
# 使用轮廓系数法
from sklearn.metrics import silhouette_score

score_list = list()
silhouette_int = -1
for n_clusters in range(2,10):
    model_kmeans = KMeans(n_clusters=n_clusters)  #建立聚类模型对象
    labels_tmp = model_kmeans.fit_predict(df_ifl_scaled)  #训练聚类模型
    silhouette_tmp = silhouette_score(df_ifl_scaled, labels_tmp)  #得到每个K下的
平均轮廓系数
    if silhouette_tmp > silhouette_int:  #如果平均轮廓系数更高
        best_k = n_clusters  #保存最佳的K值
        silhouette_int = silhouette_tmp
        best_kmeans = model_kmeans  #保存模型实例对象
        cluster_labels_k = labels_tmp  #保存聚类标签
    score_list.append([n_clusters, silhouette_tmp])  #将每次K值及其平均轮廓系数记录

print(score_list)
print('最优的K值是：{0} \n对应的轮廓系数是{1}'.format(best_k, silhouette_int))
```

当K值为3时，对应的轮廓系数最接近于1。再次验证最佳K值为3。

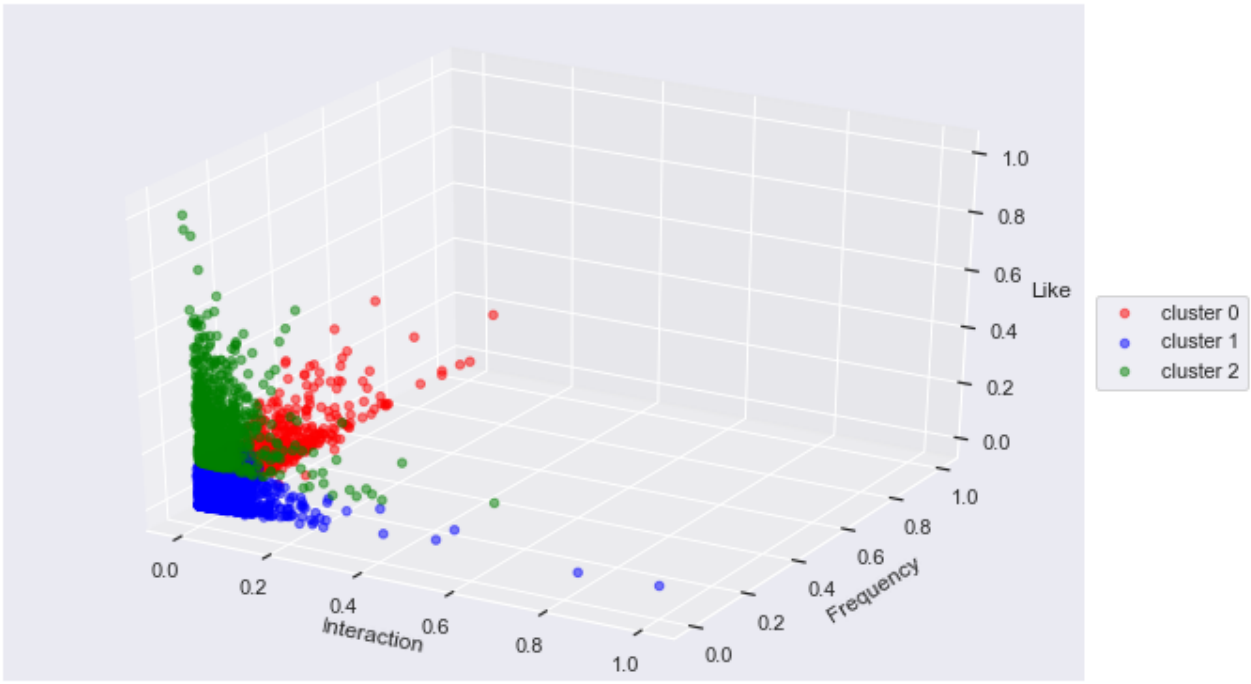
开始建模

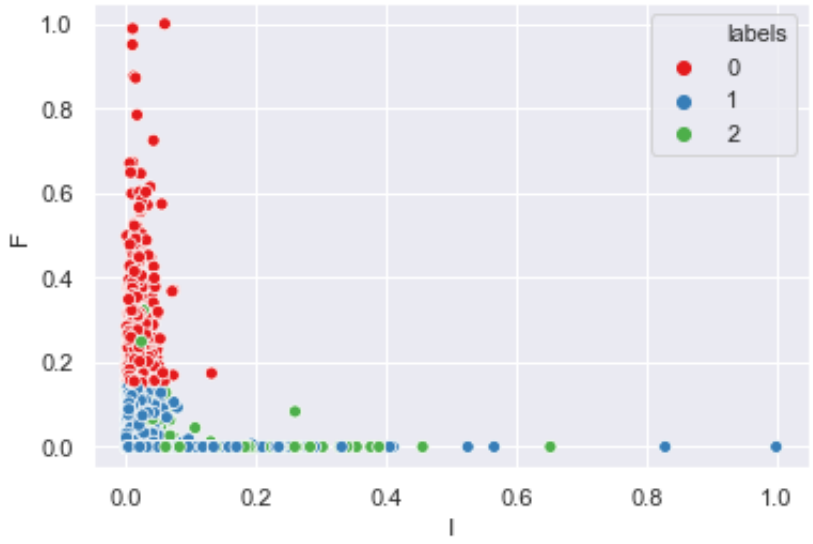
```
#采用k-measn聚类算法对客户数据进行客户分群，聚成3类
kmodel = KMeans(n_clusters=3,random_state=99)
kmodel.fit(df_ifl_standardize)

kmodel.cluster_centers_   #查看聚类中心
kmodel.labels_ #查看样本对应类别
df_ifl_scaled['labels'] = kmodel.labels_
df_ifl_scaled.head()

#查看各类别客户群人数
df_ifl_scaled.groupby('labels')['labels'].count()
```

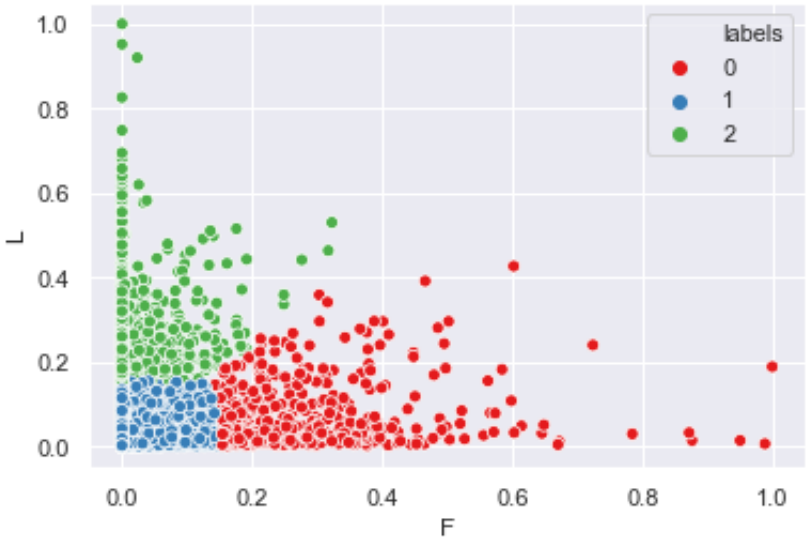
可视化分类效果

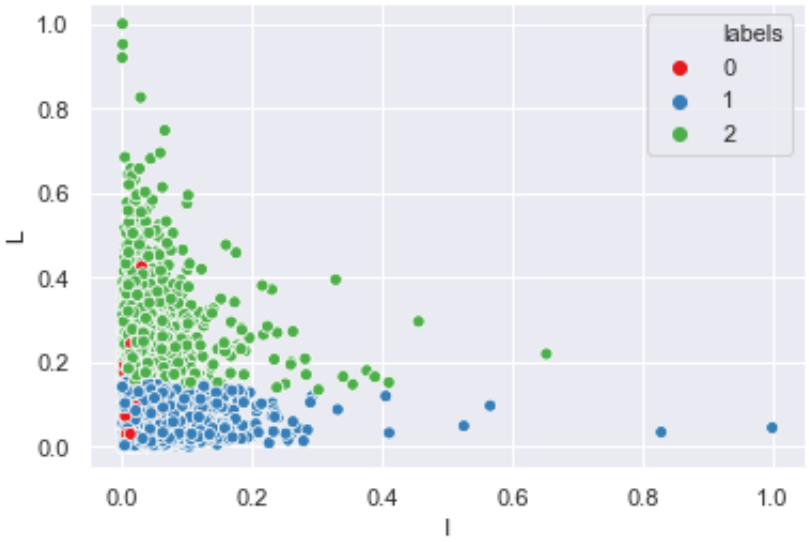
```
plt.rcParams["font.family"] = ['Arial Unicode MS']  # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus']=False #解决保存图像是负号'-'显示为方块的问题

#3D散点图展示
from mpl_toolkits.mplot3d import Axes3D
# step 1: data preparation
I=[]
F=[]
L=[]
mycolors=['red','blue','green']
for i in range(3):

 I.append(df_ifl_scaled.loc[df_ifl_standardize.labels==i,'I'].values.tolist())

 F.append(df_ifl_scaled.loc[df_ifl_standardize.labels==i,'F'].values.tolist())

 L.append(df_ifl_scaled.loc[df_ifl_standardize.labels==i,'L'].values.tolist())

# step 2: 3D scatter plot
fig=plt.figure(figsize=(8,5))
ax=Axes3D(fig)
for i in range(3):
    ax.scatter(I[i], F[i],
L[i],c=mycolors[i],marker='o',alpha=0.5,label='cluster '+str(i))
ax.set_xlabel('Interaction')
ax.set_ylabel('Frequency')
ax.set_zlabel('Like')
#ax.set_xlim(0,4)
#ax.set_xticks(list(range(4)))
ax.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.show()

#二维平面散点图进一步观看分类效果
```

```
sns.scatterplot(x="I", y="F", hue="labels", data =df_ifl_scaled,
palette='Set1')
sns.scatterplot(x="F", y="L", hue="labels", data =df_ifl_scaled,
palette='Set1')
sns.scatterplot(x="I", y="L", hue="labels", data =df_ifl_scaled,
palette='Set1')
```







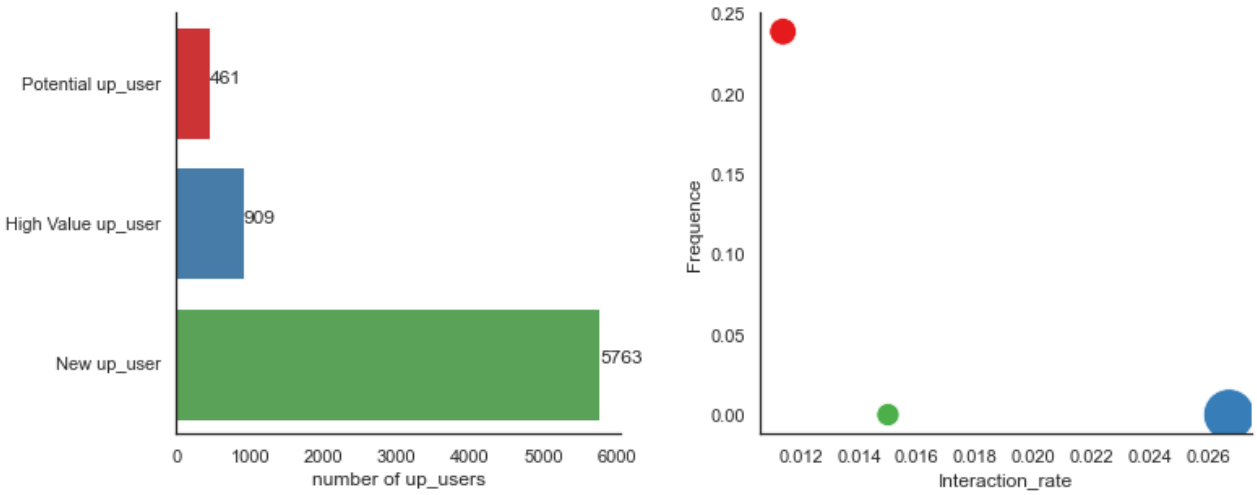


为分析做进一步可视化

```
# replace k-means cluster names with more meaningful names
d1={0:"Potential up_user", 1:"New up_user", 2: "High Value up_user"}
df_ifl_scaled.loc[:,"segments"]=df_ifl_scaled.loc[:,"labels"].map(d1)

df_ifl_scaled.head()

df_ifl_scaled['labels'].value_counts()


# up主分层可视化
plt.figure(figsize=(10,4))
seg_names=['Potential up_user','High Value up_user','New up_user']

# plot the number of up_authors in each segment
sns.set_style("white")
plt.axes([0, 0, 0.38, 0.9])
seg=df_ifl_scaled.groupby('segments').size().to_frame().rename(columns=
{0:'number of up_users'}).reset_index()
```

```
sns.barplot(x='number of
up_users',y='segments',data=seg,order=seg_names,palette='Set1')
for i in range(3):
    number=int(seg.loc[seg.segments==seg_names[i],'number of up_users'])
    x_pos=round(number,2)
    plt.text(x_pos,i,number)
plt.ylabel("")
sns.despine()

#显示Interaction_rate,Frequence,Like_rate的中值
df_ifl2=df_ifl_scaled.groupby('segments').agg(I=('I',np.median),F=
('F',np.median),L=('L',np.median),size=("labels","size")).reset_index()
# df_ifl2=df_ifl_scaled.groupby('segments').agg(I=('I',np.mean),F=
('F',np.mean),L=('L',np.mean),size=("labels","size")).reset_index()
plt.axes([0.5,0,0.42,0.9])
sns.scatterplot(x='I',y='F',hue='segments',hue_order=seg_names,palette='Set1',s
ize='L',sizes=(200,1000),legend=False,data=df_ifl2)
plt.xlabel('Interaction_rate')
plt.ylabel('Frequence')
sns.despine()
```

下面右图横纵坐标分别代表互动率中值（Interaction_rate）和发布周期中值（Frequence），散点大小代表喜爱度中值（Like_rate），散点颜色代表分类标签。



3.2 聚类结果特征分析

模型聚类成的三个簇按其特征被划分为三个群体。每个群体的特征如下：

labels0(红色)：成长期up主

该群体的I值略低，且F值明显高于其他两类，L值处于中间状态。

此特征的群体与观众的互动率不高，创作周期较长，但视频点赞率还不错，总体评估还有很大提升空间。

该类别大部分为业余up主，处于成长期。

labels1(蓝色)：高价值up主

该群体的I值和L值都是最高，且F值也很低。

此特征的群体拥有10万以上的粉丝；视频更新稳定，在保证不拖更的同时，能收获到观众大量的三连（点赞、投币、收藏）。为该分区的高价值up主群体。

labels2(绿色)：新手up主

该群体占比最大；I值稍高于中值，F值为三者中最低，但L值平均点赞率是短板，最低。

此特征的群体与观众的互动良好，更新频率高，活跃程度高，但暂未能获得大量观众的点赞、投币、收藏；需要提高视频稿件的质量，创作符合观众口味的视频，收获更多的三连。

四、运营策略优化

针对新手Up主，创作频率高，热情高，但是质量欠佳，大多是跟随热点事件，标题党一类。可以推出了一系列学习激励政策，提升新人创作力。开展一些竞赛活动以激励底层UP主生产数量更多质量更高的内容，以此来提升UP主的粉丝数量，最终使UP主获得成长，向上流动。

针对成长期Up主，作者往往是花费了大量的时间成本与金钱成本，视频质量不错，但是产出不多，可以开展一系列激励计划激发up主创作热情。比如只针对10万以下的粉丝的up主才可参与的“全勤挑战”的有奖活动，激励UP主保持活跃，并培养UP主养成投稿的习惯。

针对高价值up主，是高质量视频的积极贡献者，能够带动社区氛围和流量。需要做好深度情感维系和利益共享，建议与这些优质UP主深度绑定。比如通过付费签约绑定，防止核心UP主被竞争对手挖走或流失。