***PEKING UNIVERSITY***

*MLAI 23*

机器学习与人工智能 小组作业

*Author:*

## MLAI 13组

*Supervisor:*

北京大学 张颖婕

A Lecture note submitted for the MLAI 23

2023 年 1 月 20 日

目录

[MLAI 13组 1](#_Toc156658858)

[1 项目总览 3](#_Toc156658859)

[1.1 选题背景 3](#_Toc156658860)

[1.2 小组成员分工情况（按姓名首字母排序） 4](#_Toc156658861)

[2 创业者角度 5](#_Toc156658862)

[2.1 创业性格检测 5](#_Toc156658863)

[**2.1.1** 背景和数据介绍 5](#_Toc156658864)

[**2.1.2** 模型训练及预测 6](#_Toc156658865)

[**2.1.3** 创业性格匹配度检测 8](#_Toc156658866)

[2.2 员工满意度预测 12](#_Toc156658867)

[**2.2.1** 背景和数据介绍 12](#_Toc156658868)

[**2.2.2** 员工满意度预测 12](#_Toc156658869)

[2.3 融资预测 14](#_Toc156658870)

[**2.3.1** 背景和数据介绍 14](#_Toc156658871)

[**2.3.2** 数据清洗及特征工程 15](#_Toc156658872)

[**2.3.3** 融资预测模型 17](#_Toc156658873)

[**2.3.4** 融资预测结果分析 19](#_Toc156658874)

[2.4 破产预测 19](#_Toc156658875)

[**2.4.1** 背景及数据介绍 19](#_Toc156658876)

[**2.4.2** 破产预测模型 20](#_Toc156658877)

[3 投资者角度 22](#_Toc156658878)

[3.1 公司状态预测 22](#_Toc156658879)

[**3.1.1** 特征选择 22](#_Toc156658882)

[**3.1.2** 随机森林分类 24](#_Toc156658883)

[**3.1.3** PCA降维 26](#_Toc156658884)

[3.2 机器学习结果可视化 27](#_Toc156658885)

[4 总结 29](#_Toc156658886)

[5 参考文献 29](#_Toc156658887)

摘要

本报告利用一共5组、总量12w+的数据集进行了各种机器学习算法，包括监督学习回归模型、监督学习分类模型和非监督学习算法，并进行了机器学习结果的可视化，最终对创业者和投资者提供多角度切实有价值的建议。创业者角度，包括适合创业的性格、预测员工满意度的方法、预测是否可以获得融资以及破产预测，投资者角度，包括企业状态预测和美国企业可视化。

# 项目总览

## 选题背景

当今时代，创业之潮迅速涌起，各行各业充满了新的机遇与挑战，许多人胸怀崇高的理想与抱负，加入了创新创业的队伍中，全球初创公司数不胜数，不断涌现。您是否曾有过创业的想法？您是否适合成为创业者？您了解企业创建与运营的关键之处吗？您会选择投资哪些初创公司？我们的小组项目将从机器学习的视角，带您一步步了解创业的各方面情况！我们的思路为：

- 2.1 您的性格是否适合创业？

- 2.2 如何招纳并留住员工？

- 2.3 能否获得一笔好的融资

- 2.4 如何保持企业不破产

- 3.1 从投资者角度考虑，他们会看好什么企业？

- 3.2 以美国为例做企业成功预测的可视化

我们使用的数据共包括5组（2.3和3.1共用一组数据），数据总量12w+（包括文字和图片数据集）。数据来源包括Kaggle、Chalearn、Github等。

我们使用的模型包括：

- 有监督回归模型：CNN、线性回归（拉索回归、岭回归）、K近邻等；

- 用到的有监督分类模型：决策树、随机森林、梯度提升、XGBoost、支持向量机等；

- 用到的无监督模型：自编码器、PCA

希望我们的工作可以切实帮助投资者和创业者做出决策，助力创业成功！

## 小组成员分工情况（按姓名首字母排序）

蔡仪纯：2.4、3.1

蒋丽雯：1、3.1、4

李柯毅：2.1、2.3、3.2、小组项目代码网站创建及代码整合

辛婷毅：2.2、2.3

以上各部分代码和报告文档由相应成员完成，最终全体成员共同进行报告整合。

# 

# 创业者角度

成功企业的例子不胜枚举，创业者们的故事耳熟能详。他们或乐观勇敢，永远能想到新奇的点子，是天生的创业者；他们或对事业充满热情，善于沟通与合作，收获投资者的青睐；他们或勤勉刻苦，步步为营，将小企业逐步打造为“商业帝国”。然而，客观来看，创业之旅往往伴随着一系列独特的风险和挑战。不是所有人都适合创业，也不是所有初创公司都能抵挡竞争的残酷。创业失败的代价往往是较为惨痛的，所以对于创业者来说，需要综合考虑各类因素，谨慎做出创业的决定，并尽量运营好自己的公司。

在本部分，我们希望通过各类数据的分析和预测，为创业者提供相应建议，我们的模型共包括以下几个内容：从创业者的性格角度，检测创业者是否适合创业；企业应该怎样提高员工的满意度；创业后能否获得较好的融资；怎么样的企业可能破产。我们希望可以通过这些模型，帮助创业者合理评估创业的决定，降低创业失败的可能。

## 创业性格检测

### 背景和数据介绍

创业不可盲目，我们首先要了解自身的性格特点，做出是否创业的决定。2023年10月发表在Nature的文章表明[1]，创业者性格与创业成功之间具有较强的相关性。该文章基于BIG FIVE人格[[1]](#footnote-1)和Crunchbase数据库展开分析，发现成功的创业者与失败的创业者或者员工的性格具有显著不同。同时，创业者性格还可进一步细分为六类，分别具有不同的创业偏好和路径。在本部分，我们将利用深度学习模型（自编码器），辅助创业者检测其BIG FIVE人格，并根据检测结果给出最匹配的创业性格以及匹配度。

传统的BIG FIVE人格检测往往通过问卷进行，过程繁琐，且具有较大的主观性。为此我们设计了一种更简单快捷的BIG FIVE检测方式，即通过自拍照（或视频中的自拍截图）进行BIG FIVE人格预测，并根据公式计算出创业性格分数，以及最适合的创业类型。我们希望实现，创业者只需上传自己的一张自拍照或者一段视频，即可得到自身的BIG FIVE人格特征以及创业匹配度信息。

我们的训练数据来自chalearn竞赛中的视频人脸的图片数据集，数据包括了BIG FIVE人格的标注。数据集总共包括了30935张208\*208的自拍图（X）（图2-1-1）以及相应数量的五维BIG FIVE人格指数作为标注（Y），标注数据均经过标准化，位于0-1范围内。模型部分我们参考了Github上的开源模型Personality的框架，主要使用了CNN及Autoencoder。这里我们完善了模型参数配置，并进一步改进并加入了创业匹配度预测的功能。完整数据集和代码可见最后的附件。

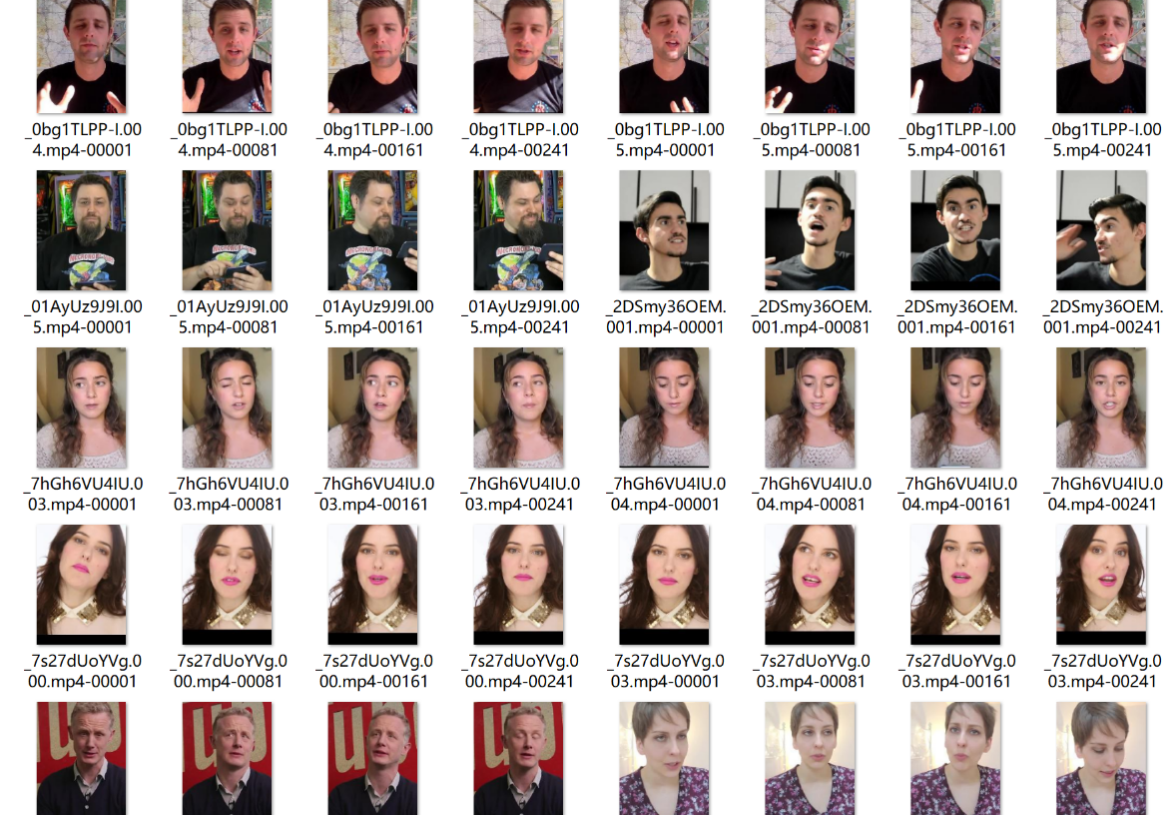


图2-1- 1 Chalearn自拍图片数据集（部分）

### 模型训练及预测

首先，我们将数据按照9:1划分为训练集和测试集（训练集数据量为28152，测试集数据量为2783），并分别保存为pickle文件。之后我们利用CNN建立了一个自编码器模型（autoencoder），无监督下进行图像的特征提取和压缩重构，其可分为编码器和解码器两部分。编码器的输入层是形状为（208,208,1）的图像，通过三个卷积层和三个池化层进行特征提取和降采样，逐渐压缩图像表示，最终输出形状（26,26,16）的低维表示。解码器是三个卷积层和三个上采样层的交替，最终通过上采样层将特征图还原为（208,208,1）的形状，实现了图片重构。自编码器的模型框架和参数设定如下表（部分），后续主要利用编码器部分压缩提取。

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

input\_7 (InputLayer) (None, 208, 208, 1) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_37 (Conv2D) (None, 208, 208, 4) 40

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_19 (MaxPooling (None, 104, 104, 4) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_38 (Conv2D) (None, 104, 104, 8) 296

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_20 (MaxPooling (None, 52, 52, 8) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_39 (Conv2D) (None, 52, 52, 16) 1168

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_21 (MaxPooling (None, 26, 26, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_40 (Conv2D) (None, 26, 26, 8) 1160

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_19 (UpSampling (None, 52, 52, 8) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_41 (Conv2D) (None, 52, 52, 4) 292

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

up\_sampling2d\_20 (UpSampling (None, 104, 104, 4) 0

表2-1- 1 自编码器框架设定

之后我们进一步建立了卷积神经网络模型（Classifier），用于有监督条件下的回归预测。该模型主要包括了编码器和全连接层两部分，编码器包括两个卷积层和两个池化层，输出经过reshape后转换为一维向量（784,），之后经过三个全连接层，分别包括500、500、100个神经元，最后到达输出层，共包括5个神经元，分别代表BIG FIVE的五类人格，参数总数为698245个[[2]](#footnote-2)。模型具体框架如下表所示：

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

input\_6 (InputLayer) (None, 26, 26, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_11 (Conv2D) (None, 26, 26, 16) 2320

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_11 (MaxPooling (None, 13, 13, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

conv2d\_12 (Conv2D) (None, 13, 13, 16) 2320

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

max\_pooling2d\_12 (MaxPooling (None, 7, 7, 16) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

reshape\_6 (Reshape) (None, 784) 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_21 (Dense) (None, 500) 392500

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_22 (Dense) (None, 500) 250500

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_23 (Dense) (None, 100) 50100

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

dense\_24 (Dense) (None, 5) 505

表2-1- 2 神经网络模型

之后我们利用迭代循环训练模型，输入为编码器压缩后的embeding\_train，模型为之前训练的Classifier。这里我们迭代了400次[[3]](#footnote-3)，并记录迭代过程中的loss（训练集的均方误差）、val\_loss（测试集的均方误差）以及mean\_acc（平均准确度）。这里准确度的定义为五个人格预测相似度的均值。模型结果如图2、3所示。

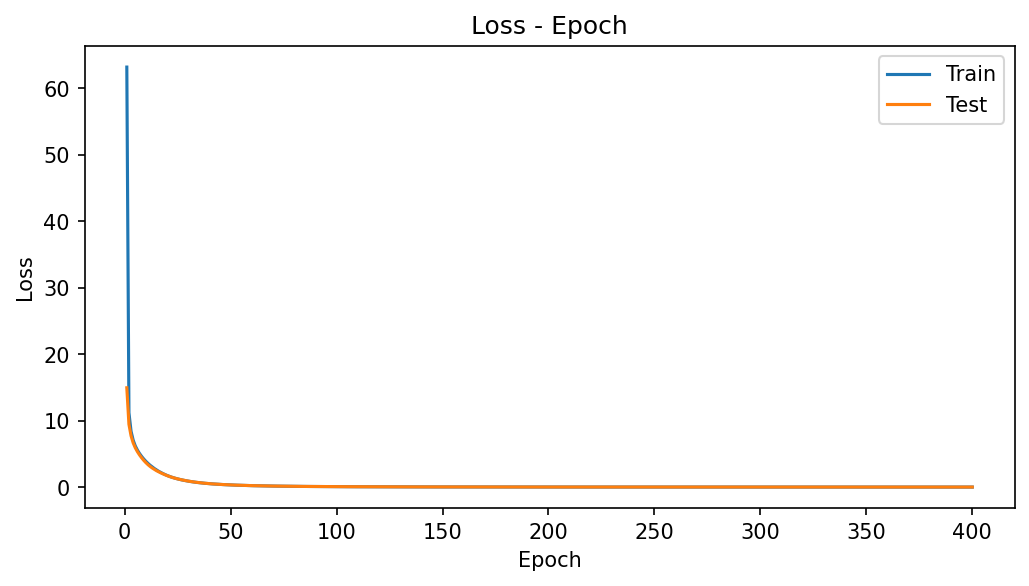


图2-1- 2 模型Loss随迭代次数变化

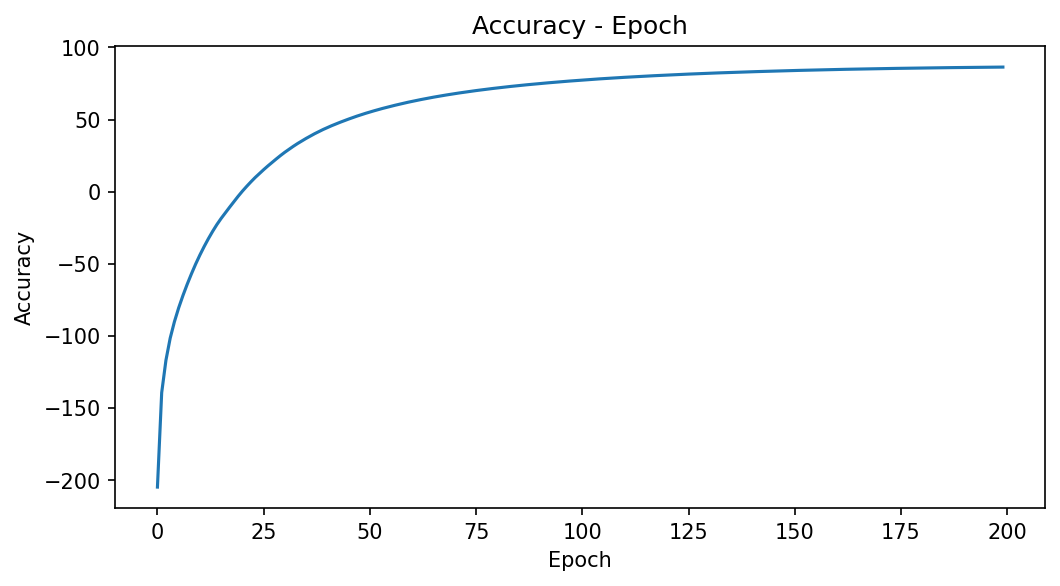


图2-1- 3 模型Accuracy（mean\_acc）随迭代次数变化

经过迭代我们找到并保存了最佳模型，其精确度（mean\_acc）为88.04%。以下我们将利用模型预测创业性格匹配度。

### 创业性格匹配度检测

我们首先回到论文，论文中展示了成功的创业者和失败的创业者及员工之间BIG FIVE人格特征的对比（图2-1-4&图2-1-5）。由此我们可以看出，成功的创业者的BIG FIVE人格数值均更高（这里按照神经质为指标替代论文中的情绪稳定性），体现出正相关的关系。同时图5还表明，openness（开放性）对于创业成功影响最大，其次是Extraversion，故在大五人格和创业性格的匹配中赋予它们更高的权重。[[4]](#footnote-4)这里我们通过加权平均得到创业性格匹配度的计算公式为：P= (O \* 1.5 + E \* 1.25 + A + C + N)/(1.5 + 1.25 + 1 + 1 + 1)\*100%。其中P为你的性格与创业性格匹配度，O、C、E、A、N分别代表大五人格的英文首字母。

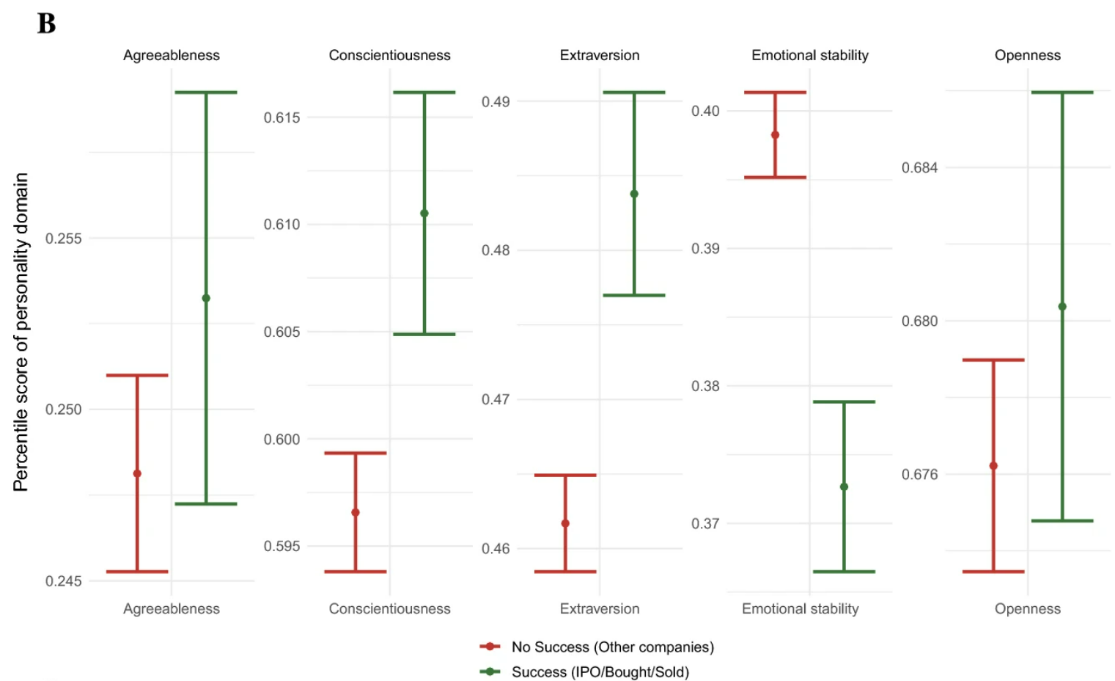


图2-1- 4 成功与失败的创始人大五人格对比

（注：论文中使用的情绪稳定性可以看做我们的研究中神经质的反义指标）[1]

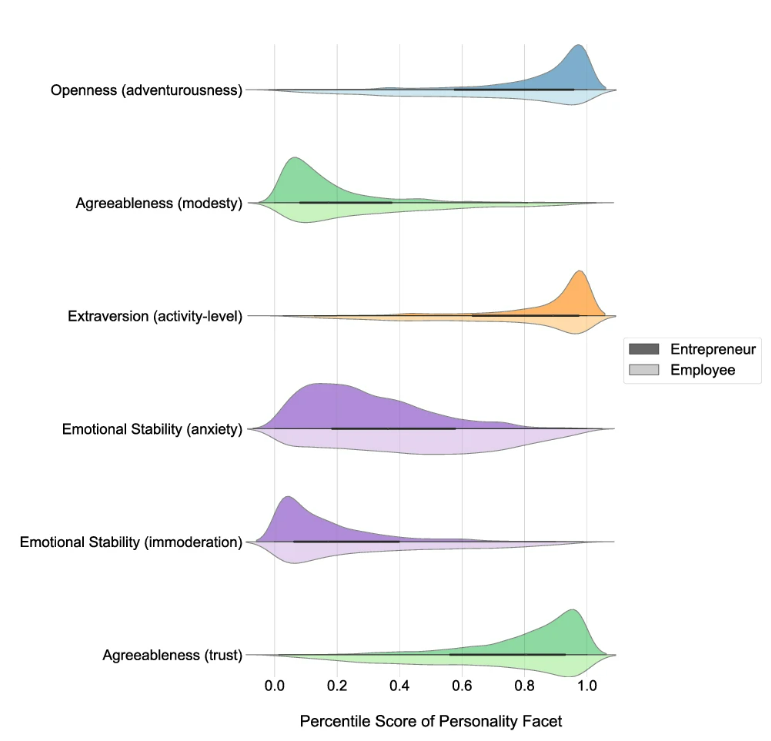


图2-1- 5 创业者和员工之间的大五人格对比和权重分配[1]

将公式应用到我们的模型中，对训练集数据进行预测，并绘制直方图如图2-1-6，可见近似高斯分布，说明公式较为合理。创业性格匹配度最高为93%，最低为45%。

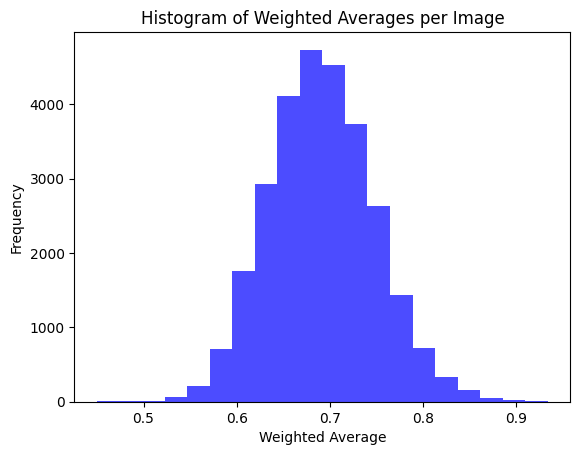


图2-1- 6 训练集的创业性格匹配度直方图

此外，论文中还根据人格特征将成功的创业者进一步分为六类，分别对应于不同的行业（图2-1-7），每种类型的创业者均呈现出不同的人格特点：

* Fighters在neuroticism一项中具有极高的评分；
* Operators则在conscientiousness和agreeableness两项得分较高
* Accomplishers注重extraversion和conscientiousness；
* Leaders在openness和agreeableness较高；
* Engineers则有极高的openness；
* Developers则在各项里更为均匀。

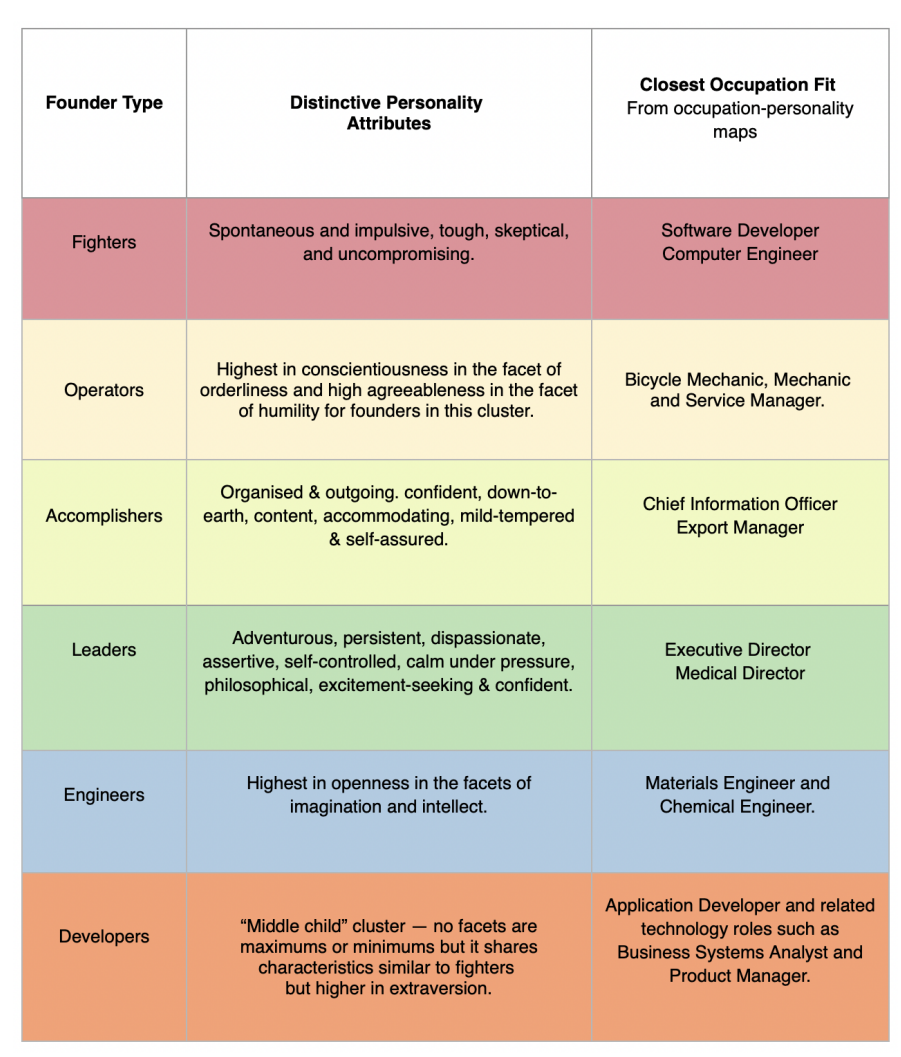


图2-1- 7 创业者类型、人格特点及行业

根据以上特点，我们可以基于我们的模型对创业者做进一步的类型划分。这里一个简单的策略是比较以上各项的平均值并选取最高的作为类型划分结果。最终结果如图2-1-8所示，Fighters和Accomplishers人数最多，而Developers在数据集中没有发现，数量为0。这说明大多数人还是在某几方面更为突出，各方面人格均衡且数值都很高的人寥寥。对应图2-1-7可知，fighters更适合从事互联网的创业，最多的fighters人数从侧面印证了当今成功的互联网创业公司数量最多这一事实。

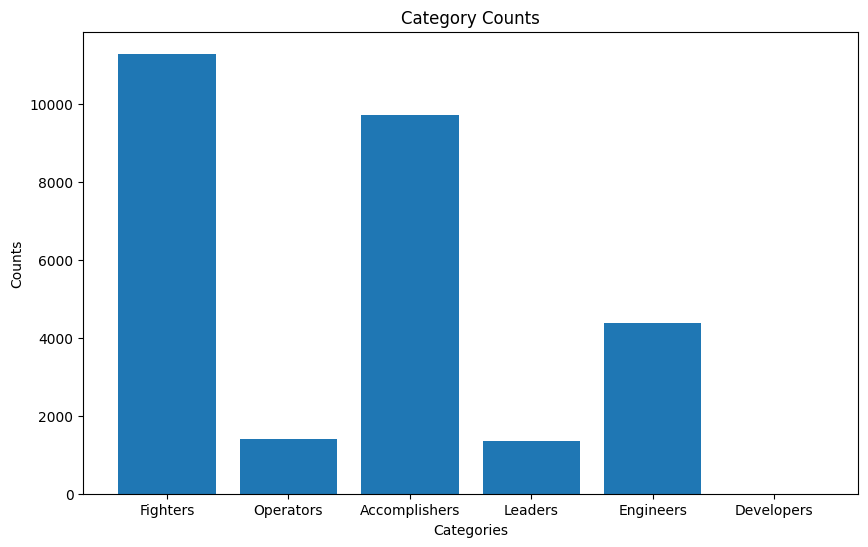


图2-1- 8 训练集中的创业者类型分布图

## 员工满意度预测

### 背景和数据介绍

初创在企业管理的早期阶段，人力资源面临着许多挑战，其中最为紧迫的问题之一是员工满意度。员工的满意度直接影响着他们在工作中的投入和绩效。由于创业公司往往处于高度不确定性的环境中，保持员工的满意度对于提高生产力和团队稳定性至关重要。

本节中，我们将使用Zakari提供的员工满意度调查数据集，对员工满意度的各个方面进行和预测。该数据集共有15787行数据，特征包括员工的身份识别号、自我报告的满意度水平、绩效评估、项目参与情况、工作时数、在公司的任职年限、工作事故、过去5年内的晋升情况、部门隶属关系和薪资水平。我们希望通过这个数据集进行员工满意度预测，建立预测模型，以更好地理解员工满意度的动态变化，为公司提供更具实效性的人力资源管理策略。

### 员工满意度预测

在读取数据后，我们首先对数据集中的缺失值和重复行进行了处理，以确保每条数据的完整性。接下来，我们对部门（dept）和薪资水平（salary）这两列特征通过“LabelEncoder”包进行 编码，将结果存储在了新创建的部门编码（dept\_encoded）和薪水编码（salary\_encoded）列中，为后续的分析和建模做好了数据准备工作。

接下来，我们将数据集划分为训练集和测试集，测试集大小为20%，并设置随机种子为42。我们使用了不同的回归算法，包括线性回归、随机森林回归、支持向量回归、决策树回归、K近邻回归、岭回归、Lasso回归这七种回归算法，以比较它们在预测员工满意度方面的表现，并通过均方误差（MSE）进行量化评估。我们发现随机森林算法的均方误差最小，因此使用它作为用于分析的预测模型。

对于使用随机森林算法预测的模型，我们获取了变量的重要性并将其可视化。可以从可视化结果看到，员工同时间正在工作的项目数量、每个月的平均工作时长、所属部门、在公司待的时间和薪水是对员工满意度影响最大的五个因素。值得注意的是，前两项因素的重要性是第三个因素的重要性的两倍多，可以看出工作量和工作时间是员工满意度的决定性因素。

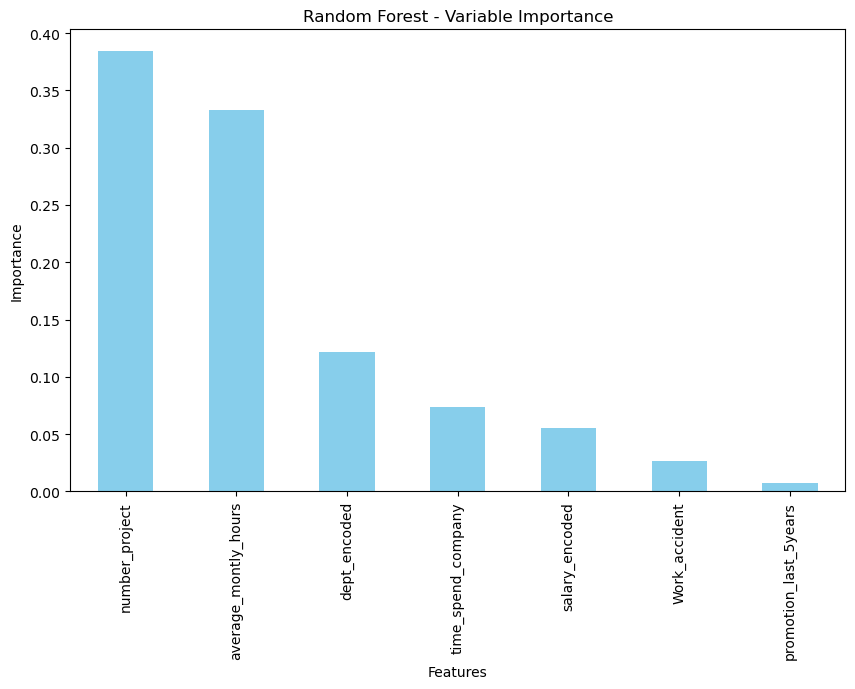


图2-2- 1 员工满意度的决定性因素

接下来，我们使用“pdpbox”库，对员工同时间正在工作的项目数量和每个月的平均工作时长这两个变量进行了部分依赖图（Partial Dependence Plots）的绘制。 总体而言，员工同时间工作的项目数量越多、每个月的平均工作时长越长，员工满意度就会越高。然而，当项目数量多于5、工作时长超过300个小时，员工的满意度便会下跌。这提示初创公司需要审视工作分配、项目管理和员工福利等方面的因素，以避免员工感到过度压力和疲劳，进而提高整体的员工满意度。

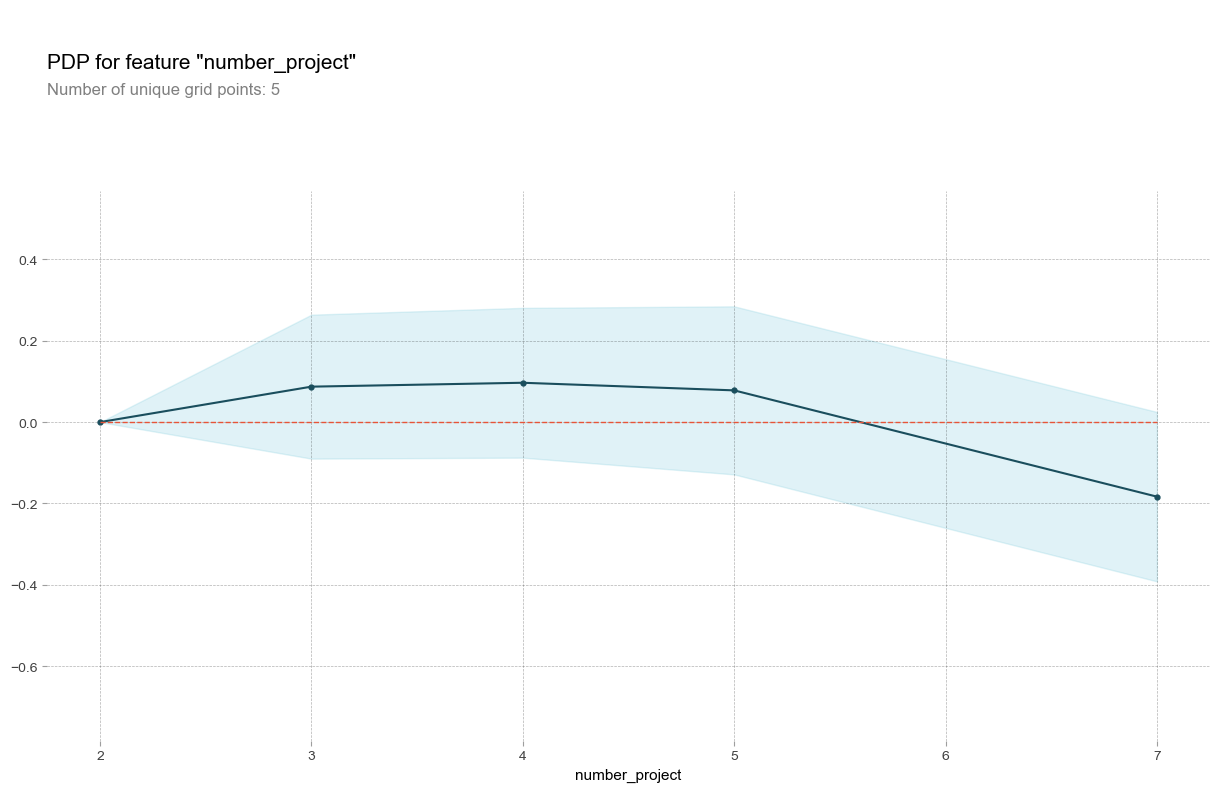


图2-2- 2 员工项目数量的部分依赖图

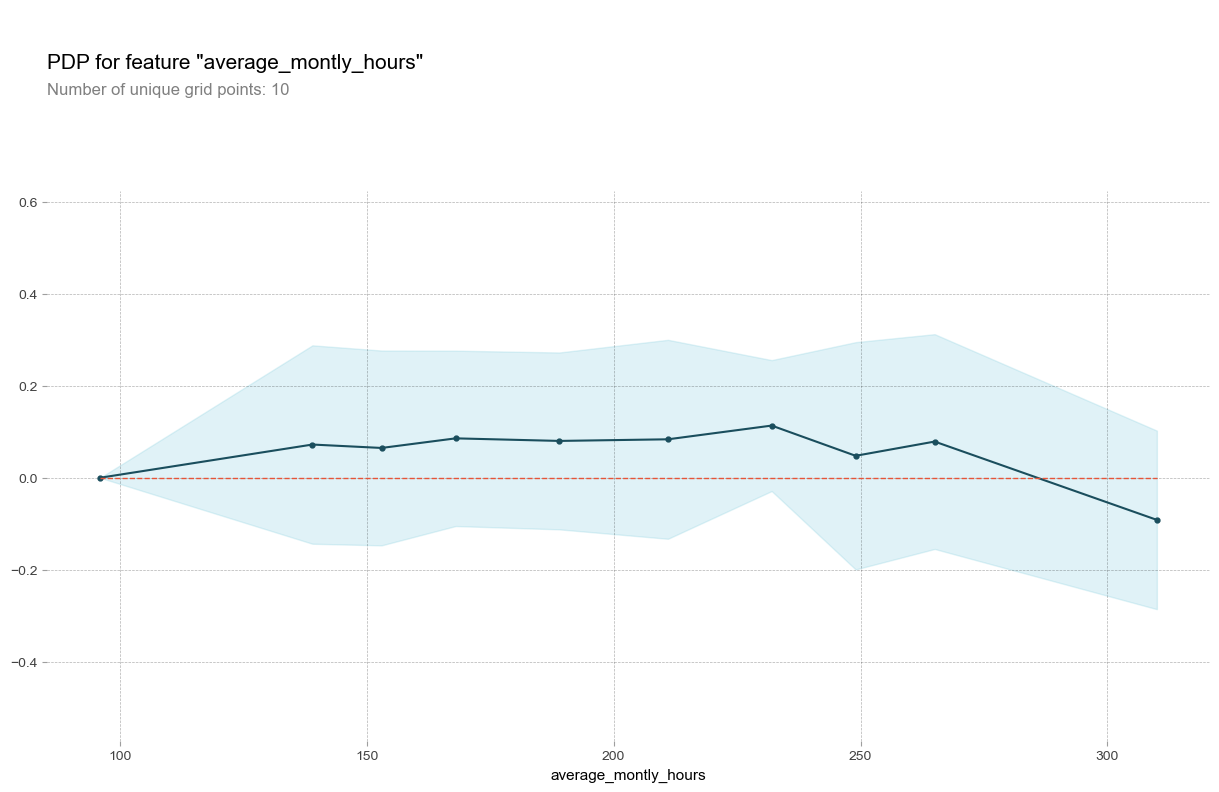


图2-2- 3 员工月平均工作时长的部分依赖图

## 融资预测

### 背景和数据介绍

初创企业面临很多挑战，其中最严峻的是资金的匮乏。大多数企业在创办开始是不盈利的，它们可能需要几个月甚至几年的时间才能看到真正的利润，因此通过融资获得足够的资金并维持良好的财务状况，对于初创企业是非常重要的。

本节中我们将利用Kaggle的初创企业数据集对企业的融资情况进行分析和预测。数据主要由Liston等人于Crunchbase网页爬取并整理而成，共包括49438家初创企业，最后更新于2020年。对于每家初创公司，数据包括了公司的成立时间、地点、行业、网页链接等基本信息，公司经营状态（运营/被收购/倒闭），以及具体的投资情况（各轮次投资金额和时间等），总计39列。具体数据可参见附件。

本部分我们将重点研究公司的投资情况，故在此对数据所包含的各投资轮次进行简要介绍，以便后续预测等工作的理解：

1. **Seed（种子轮）：**
   * 初创公司最早的融资阶段，通常用于验证产品概念、进行初步市场调研和组建初始团队。
   * 投资者通常是天使投资人或初创孵化器。
2. **Venture（风险投资）：**
   * 初创和成长公司提供资金支持的投资方式。这可以包括多个融资轮次
3. **Equity Crowdfunding（股权众筹）：**
   * 通过互联网平台从大量个人投资者那里筹集小额资金，以换取公司股权。
4. **Undisclosed（未公开融资）：**
   * 公司未公开具体融资细节的情况，可能是出于保密或策略考虑。
5. **Convertible Note（可转换债券）：**
   * 一种债务工具，持有者有权在将来某个时间点将其转换为公司的股权
6. **Debt Financing（债务融资）：**
   * 公司通过借款来筹集资金，而不是出售股权。这通常需要按期偿还本金和利息。
7. **Angel（天使投资）：**
   * 个人或天使投资团体对初创公司进行的早期投资，用于支持产品开发和市场推广。
8. **Grant（赠款/补助）：**
   * 通常由政府、基金会或非营利组织提供，用于支持特定项目或研究，不需要偿还。
9. **Private Equity（私募股权）：**
   * 对非上市公司进行的股权投资，通常由私募股权基金进行。
10. **Post-IPO Equity（上市后股权）：**
    * 公司在首次公开募股（IPO）后发行的股权。
11. **Post-IPO Debt（上市后债务）：**
    * 公司在IPO后发行的债务证券，如债券。
12. **Secondary Market（二级市场）：**
    * 已发行证券（如股票、债券）的交易市场，与发行市场相对。
13. **Product Crowdfunding（产品众筹）：**
    * 通过互联网平台筹集资金，以支持特定产品的开发和生产。与股权众筹不同，投资者通常获得产品而非股权。
14. **Round A—H：**
    * 风险投资轮次，每轮通常涉及更高金额的投资，用于支持公司的不同成长阶段。随着轮次的增加，公司的估值通常也会上升。

### 数据清洗及特征工程

观察发现，数据中部分变量的格式较为混乱，需要进行清洗，主要包括以下几项：

①公司投资总量数据，数据格式较为混乱，故我们这里将数据格式进行了统一，替换了冗余的符号，并将“-”表示的空值替换为0。

②日期数据表示方法并不唯一，这里用to\_datetime函数统一转变为标准日期表示法。

③投资数据方面，由于大部分公司只经历了所列融资的几种方式，因此数据中存在大量空值，这里我们直接用fillna函数将空值填充为0。

接下来我们对变量特征做了进一步挖掘整理：

1. 国家地区：我们利用“country”数据集与表格中的国家地区代码相匹配，建立了sub-region类别特征以表示创业地区，并对其进行编码。
2. # counrtry level comparison
3. country = pd.read\_csv('country.csv') # uploading dataset to map countyr code to continent
4. country = country[['alpha-3','region','sub-region','region-code','sub-region-code']]
5. country = country.dropna(how='any',axis=0)
6. country.isnull().sum() # checking if there is null values
7. df1 = df.merge(country, left\_on='country\_code', right\_on='alpha-3')
8. **print**(df1.head())
9. df1['sub\_region'] = df1['sub-region'].astype('category')
10. label\_encoder = LabelEncoder()
11. df1['sub\_region\_cat'] = label\_encoder.fit\_transform(df1['sub\_region'])
13. category\_mapping\_region = dict(zip(label\_encoder.classes\_, label\_encoder.transform(label\_encoder.classes\_)))
14. **print**(category\_mapping\_region)
15. 所属行业：原数据集行业分类过于零散，总共有753个行业。为方便后续预测，我们将其统一归纳为43类，并进行编码。分类依据网站<https://support.crunchbase.com/hc/en-us/articles/360043146954-What-Industries-are-included-in-Crunchbase->
16. 时间：时间特征选取了公司开始创业的时间及首次得到募资的时间，均以月为单位。
17. 融资情况：我们将各投资轮次按照相应的标准划分为类别变量（高、中、中下、低、无投资），并分别进行编码。选取变量包括天使轮、种子轮、风险投资轮、债务融资、A轮、B轮、C轮、D-H轮及其他融资。不同投资因为金额的分布区间不同，划分标准也不一，但均使得各分类数据量较为均衡。以A轮为例，我们分别按200万、500万、1000万为标准划分评级区间，并对区间进行编码。
18. # round\_A degree
19. df1.loc[(df1["round\_A"]>= 10000000),"round\_A\_degree"] = "High\_round\_A"
20. df1.loc[(df1["round\_A"]< 10000000)& (df1["round\_A"]>= 5000000),"round\_A\_degree"] = "Middle\_round\_A"
21. df1.loc[(df1["round\_A"]< 5000000)& (df1["round\_A"]>= 2000000),"round\_A\_degree"] = "MiddleLow\_round\_A"
22. df1.loc[(df1["round\_A"]< 2000000)& (df1["round\_A"]>0),"round\_A\_degree"] = "Low\_round\_A"
23. df1.loc[(df1["round\_A"]==0),"round\_A\_degree"] = "No\_round\_A"
24. df1['round\_A\_degree'].value\_counts()

### 融资预测模型

接下来，我们针对A、B、C、D-H轮分别进行了融资预测。由于四个预测模型采取的预测策略都是相似的，下面以对轮次A的预测为例，具体解释我们的预测过程。

首先，我们将经过特征工程处理的特征列提取出来，包括地区类别（sub\_region\_cat）,行业组别类别（cat\_Industry\_Group）,成立月数（months\_since\_founding），距首次融资月数（months\_to\_first\_funding），种子轮融资程度编码（seed\_degree\_encoded）,天使轮融资程度编码（angel\_degree\_encoded）,债务融资程度编码（debt\_financing\_degree\_encoded）,其他投资程度编码（Other\_Investment\_degree\_encoded），A轮融资程度编码（round\_A\_degree\_encoded）这九列。

将最后一列（A轮融资程度编码）单独剔除，作为目标变量后，我们将数据划分为训练集和测试集，其中测试集的大小为总数据集的20%，并设置了随机种子为42。接下来，我们在训练集上训练了一系列机器学习分类器，以确定在测试集上表现最佳的模型。选取的模型包括 K近邻（KNN），决策树（CART），随机森林（RF），支持向量机（SVC），梯度提升（GBM），以及XGBoost。每个模型都通过 fit 方法在训练集上进行训练，然后使用测试集计算 F1-score 进行评估。通过在测试集上计算 F1-score，并比较各模型的表现，代码最终输出了在测试集上表现最佳的模型的名称和对应的最佳 F1-score。

1. **from** sklearn.metrics **import** f1\_score, classification\_report
3. best\_model\_name = None
4. best\_model = None
5. best\_f1\_score = 0
7. # 训练并评估每个模型
8. **for** name, model **in** models:
9. model.fit(X\_train, y\_train)
10. y\_pred = model.predict(X\_test)
11. f1 = f1\_score(y\_test, y\_pred, average='weighted')
13. **if** f1 > best\_f1\_score:
14. best\_f1\_score = f1
15. best\_model\_name = name
16. best\_model = model
18. **print**(f'{name} - F1-score: {f1}')
19. report = classification\_report(y\_test, y\_pred)
20. **print**(f'{name} - Classification Report:\n{report}')
22. **print**(f'Best Model: {best\_model\_name} - Best F1-score: {best\_f1\_score}')

在这一轮比较后，我们确定了对于轮次A融资的最佳预测模型为梯度提升分类器（GBM）。接下来，我们通过GridsearchCV进行了超参数搜索，使用F1-score为评价指标，采用5折交叉验证，对学习率、最大深度和估计器数量这三个超参数进行了计算。最优超参数组合中，学习率为0.1，估计器数量为100，最大深度为3，在训练集上的F1-score为0.43，在测试集上的F1-score为0.34。最后，我们从最佳模型中提取了特征重要性，并将特征重要性按照从高到低的顺序进行排序。

1. # 定义要搜索的超参数范围
2. param\_grid = {
3. 'learning\_rate': [0.1, 0.05, 0.01],
4. 'n\_estimators': [100, 200, 300],
5. 'max\_depth': [3, 4, 5]
6. }
8. # 创建 GradientBoostingClassifier 模型
9. base\_model = GradientBoostingClassifier()
11. # 创建 GridSearchCV 实例，并设置 refit=True
12. grid\_search = GridSearchCV(estimator=base\_model, param\_grid=param\_grid, cv=5, scoring='f1\_weighted', verbose=2, n\_jobs=-1, refit=True)
14. # 在训练数据上进行搜索
15. grid\_search.fit(X\_train, y\_train)
17. # 输出最佳参数
18. **print**("Best parameters found: ", grid\_search.best\_params\_)
20. # 输出在整个训练集上的最佳模型的F1-score
21. y\_pred\_train = grid\_search.predict(X\_train)
22. f1\_train = f1\_score(y\_train, y\_pred\_train, average='weighted')
23. **print**("F1-score on train set with best model: {:.2f}".format(f1\_train))
25. # 输出在测试集上的F1-score
26. y\_pred\_test = grid\_search.predict(X\_test)
27. f1\_test = f1\_score(y\_test, y\_pred\_test, average='weighted')
28. **print**("F1-score on test set with best model: {:.2f}".format(f1\_test))
30. # 获取最佳模型
31. best\_model = grid\_search.best\_estimator\_
33. # 按照重要性从高到低排序并打印
34. sorted\_feature\_importances = sorted(zip(X\_train.columns, feature\_importances), key=**lambda** x: x[1], reverse=True)
35. **print**("Feature Importances (High to Low):")
36. **for** feature, importance **in** sorted\_feature\_importances:
37. **print**(f"{feature}: {importance}")

此外，我们还试图使用PCA对数据降维处理。但由于降维后训练的模型效果不如降维前，因此我们最终放弃使用了PCA方法。

### 融资预测结果分析

我们对于轮次A、B、C、D-H的融资预测模型的F1-score分别达到了0.43，0.59，0.72和0.80，这展示了我们的模型在不同轮次的融资性能随着融资轮次的递增而提升。

具体而言，在轮次A的融资预测中，模型关注的最重要的三个特征和其重要性为距首轮融资月数（months\_to\_first\_funding）：0.201953，行业组别类别（cat\_Industry\_Group）：0.196701，以及成立月数（months\_since\_founding）：0.196622。这说明对于A轮融资而言，首次融资的时间间隔以及公司所属的行业类型等因素对模型的影响最为显著。

对于轮次B的融资预测而言，最关键的三个特征分别是成立月数（months\_since\_founding）：0.095353，地区类别（sub\_region\_cat）：0.092633，以及行业组别类别（cat\_Industry\_Group）：0.088243。这显示了在轮次B的融资决策中，公司的成立时间和地理位置等因素起到了一定作用。

对于轮次C和D-H的融资预测而言，虽然我们将其分为两个任务进行预测，但最具影响力的三个特征却是相同的，即B轮融资程度编码（round\_B\_degree\_encoded）、成立月数（months\_since\_founding）以及地区类别（sub\_region\_cat），其中B轮融资程度编码的特征重要性最高（0.261695）。这表明C轮及之后的融资情况与B轮融资情况高度相关，反映出了B轮融资的重要性。

## 破产预测

### 背景及数据介绍

研究企业破产的风险涉及到企业生命周期中可能面临的财务、法律和战略挑战，对于初创企业至关重要。通过机器学习方法分析破产案例各项指标的特征，可以帮助初创企业运营中尽早识别破产风险并做出针对性决策，调整和改进战略以适应不断变化的市场环境。

本节中我们将利用Kaggle的企业数据集对公司的运营和破产情况进行分析和预测。数据收集自 1999 年至 2009 年的《台湾经济日报》。公司破产的定义基于台湾证券交易所的业务规定。对于每家公司，数据包括了公司资产回报率、业务毛利率、现金流动率等基本信息和公司的经营状态（倒闭与否）。总计6819条数据，96个变量。具体数据可参见附件。分析发现，后续评价机器学习模型时需要考虑该数据集数据偏斜问题：6819个公司数据中，96.77%正常运行，3.23%面临破产风险或已经破产。



图2-4- 1破产数据集描述

### 破产预测模型

我们在训练集上训练了一系列机器学习分类器，以确定在测试集上表现最佳的模型。逻辑斯蒂回归(Logistic Regression)预测准确率为96.43%，线性支持向量机预测准确率为96.33%，rbf核技巧的支持向量机预测准确率为96.58%。

1. model1 = LogisticRegression()
2. model1.fit(X\_train, y\_train)
4. # Make predictions
5. predictions = model1.predict(X\_test)
7. # Generate the confusion matrix
8. cm = confusion\_matrix(y\_test, predictions)
10. # Print accuracy
11. accuracy = accuracy\_score(y\_test, predictions)
12. **print**(f'Accuracy: {accuracy \* 100:.2f}%')
13. # Visualizing the confusion matrix
14. plt.figure(figsize=(8, 6))
15. sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
16. plt.title('Confusion Matrix LogisticRegression')
17. plt.xlabel('Predicted Label')
18. plt.ylabel('True Label')
19. plt.show()
21. # Predict probabilities for the test set
22. test\_probabilities = model1.predict\_proba(X\_test)[:, 1]
24. # Predict probabilities for the training set
25. train\_probabilities = model1.predict\_proba(X\_train)[:, 1]
27. # Calculate ROC Curve for the test set
28. test\_fpr, test\_tpr, \_ = roc\_curve(y\_test, test\_probabilities)
29. test\_auc = roc\_auc\_score(y\_test, test\_probabilities)
31. # Calculate ROC Curve for the training set
32. train\_fpr, train\_tpr, \_ = roc\_curve(y\_train, train\_probabilities)
33. train\_auc = roc\_auc\_score(y\_train, train\_probabilities)
35. # Plotting both ROC curves
36. plt.figure(figsize=(8, 6))
37. plt.plot(test\_fpr, test\_tpr, color='blue', label=f'Test ROC curve (area = {test\_auc:.8f})')
38. plt.plot(train\_fpr, train\_tpr, color='green', label=f'Train ROC curve (area = {train\_auc:.8f})')
39. plt.plot([0, 1], [0, 1], color='darkgrey', linestyle='--')
40. plt.xlabel('False Positive Rate')
41. plt.ylabel('True Positive Rate')
42. plt.title('ROC Curve LogisticRegression')
43. plt.legend(loc='lower right')
44. plt.show()

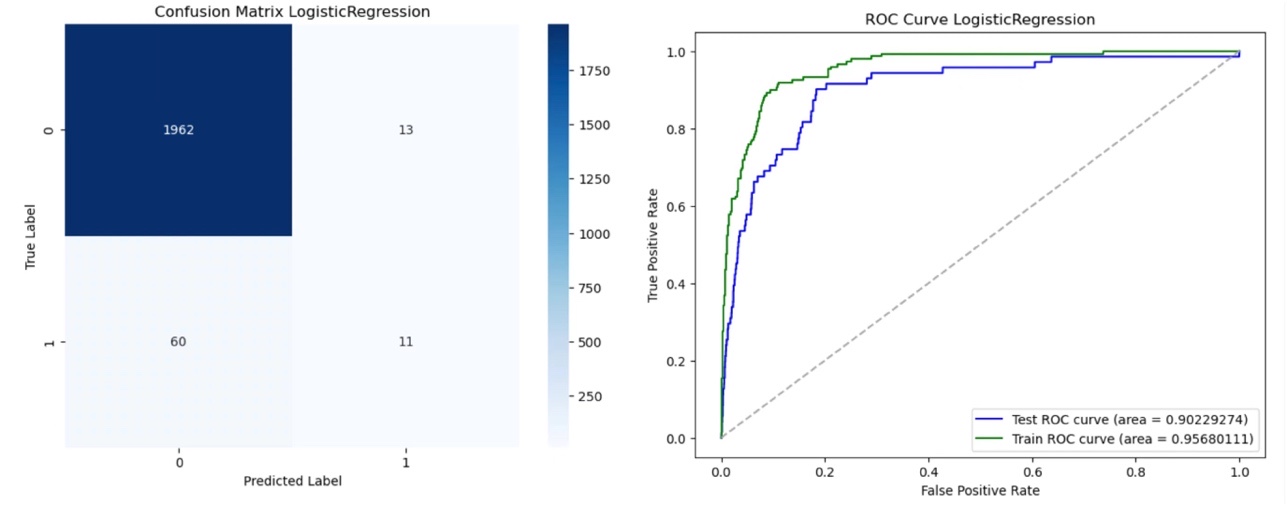


图2-4- 2 Logistic Regression 结果

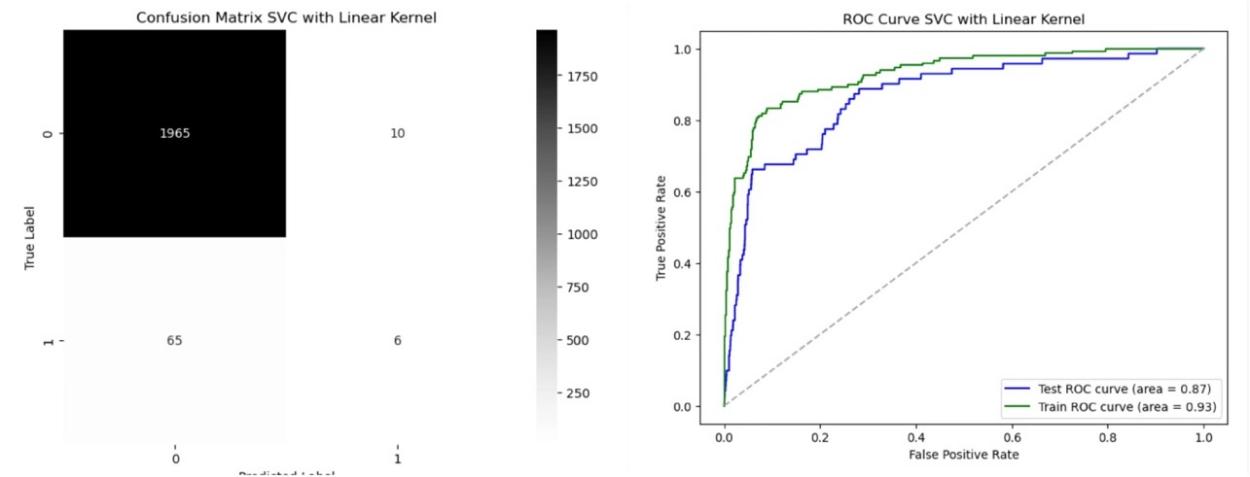


图2-4- 3 Linear SVC 结果

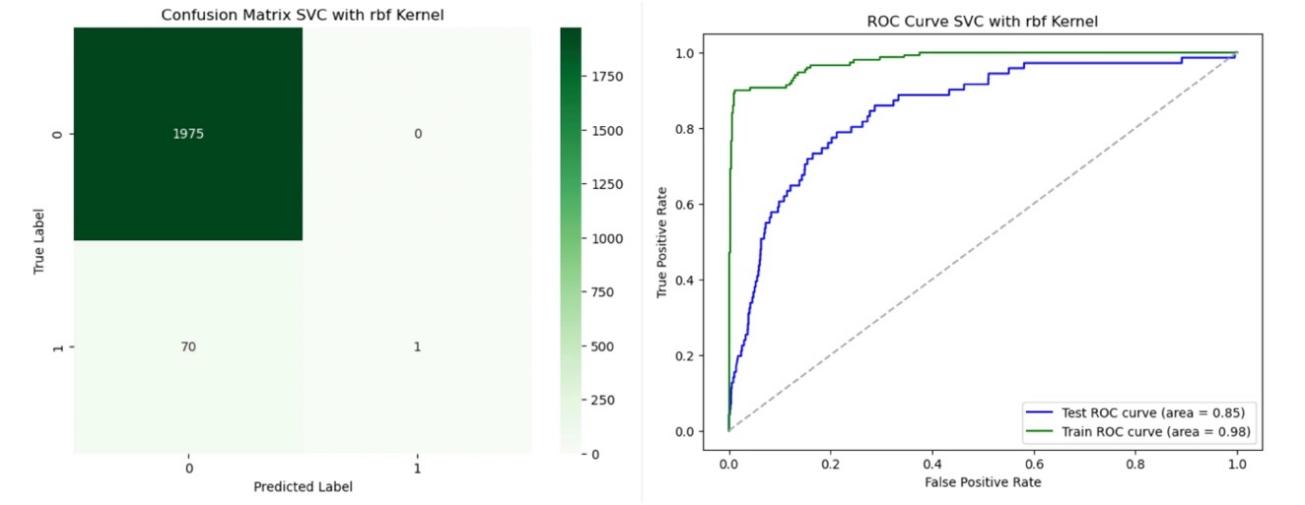


图2-4- 4 rbf Kernel SVC 结果

# 投资者角度

成功的初创企业是投资者们追逐的宝藏，而决定投资哪家公司往往是一项具有挑战性的任务。投资者们常常需要综合考虑诸多因素，以确定他们的投资是否能够在未来取得可观的回报。本部分通过分析和预测初创公司的数据集，旨在为投资者提供有针对性的参考和建议，使其能够更好地评估潜在投资。我们还提供了网页可视化，投资者可以根据个人需求选择不同的模型和参数，获得预测结果的可视化图像和ROC曲线。通过这些模型和可视化工具，我们希望能够帮助投资者做出明智的投资决策，降低投资风险，提高投资回报。

## 公司状态预测

作为初创公司的投资者，无论从公司的哪一轮融资开始对公司投资，都希望公司可以长久运营，以期获得公司股权收益。那么，投资者最关心的问题就是公司的未来和前景，以及公司保持长久运营的可能性。如果公司运营状态良好，投资者就可以获得公司股权，享受红利和股价上涨收益；如果公司被收购，原投资者就会面临失去公司控制权或股权被折价收购的风险；而如果公司破产倒闭，投资者就会面临巨额亏损，血本无归，甚至连投资本金都无法收回。所以对于投资者来说，在决定是否对一家初创公司投资时，需要十分审慎，综合考虑各种因素，尽量提高投资成功的可能性，降低投资风险。

在本部分，我们希望通过初创公司数据集的分析和预测，为投资者提供相应参考和建议，即某初创公司是否会稳定运营，而不是被收购或破产倒闭。希望可以帮助投资者合理评估投资的决定，并降低投资失败的可能。



### 特征选择

由经验得出，企业的行业和地理位置对企业能否成功运营影响很大。绘制堆叠柱状图。

1. status\_dict={'closed': 1, 'operating': 2, 'acquired':3}
2. # 使用 Seaborn 的 barplot 函数创建分层柱状图
3. plt.figure(figsize=(10, 6))  # 设置图形尺寸
5. pivot\_df = df\_plot.pivot\_table(index='cat\_Industry\_Group', columns='cat\_status', aggfunc=len, fill\_value=0)
7. # Create a stacked bar plot using Matplotlib
8. fig, ax = plt.subplots(figsize=(10, 6))
10. pivot\_df.plot(kind='bar', stacked=True, ax=ax)
11. ax.set\_yscale('log')  # Set y-axis to logarithmic scale
13. # Add labels and title
14. plt.xlabel('Industry')
15. plt.ylabel('Count')
16. plt.title('Stacked Column Chart of Industry by Status')
18. # 显示图例
19. # Create legend labels using the mapping dictionary
20. legend\_labels = [f'{value}: {key}' **for** key, value **in** status\_dict.items()]
22. # Add the legend to the right side of the plot
23. plt.legend(legend\_labels, title='Status', loc='upper right', bbox\_to\_anchor=(1, 1))
25. # Show the plot
26. plt.tight\_layout()
27. plt.show()
29. **for** key,value **in** category\_mapping\_industry.items():
30. **print**(value,":",key)

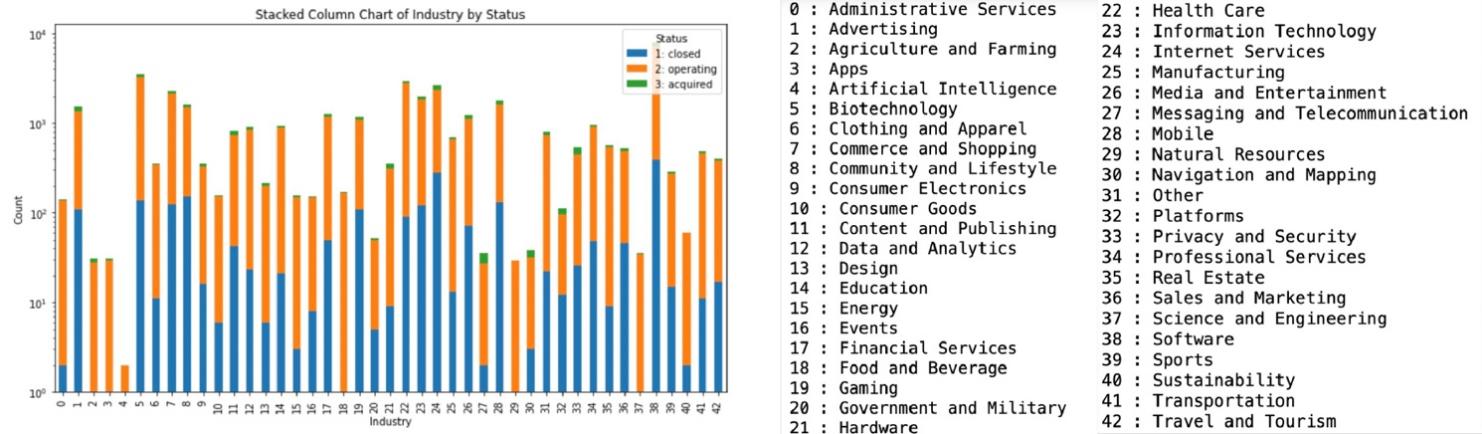


图3-1- 1 按行业划分的公司运营状态

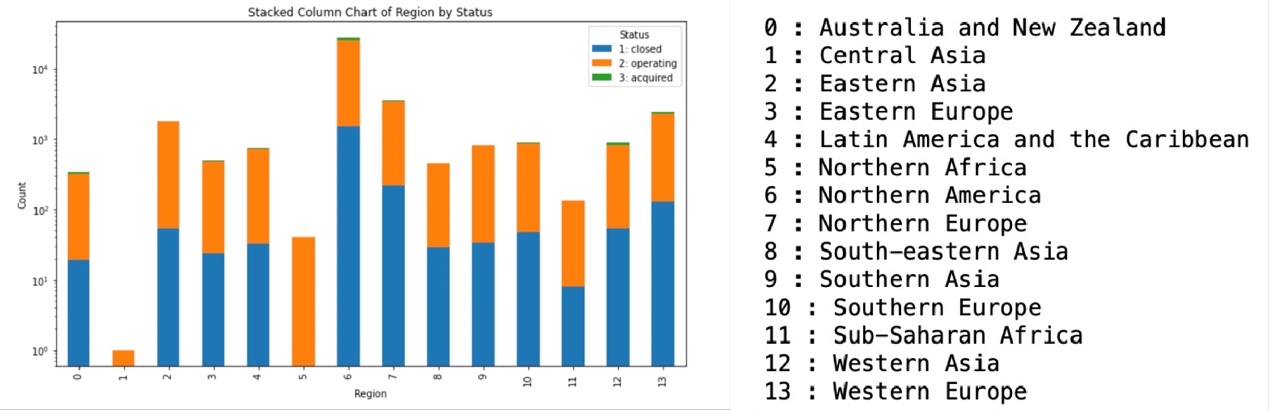


图3-1- 2 按地区划分的公司运营状态

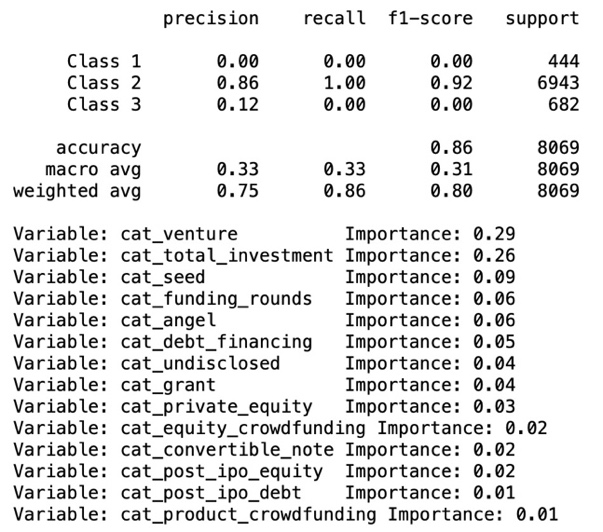
从上面两个堆叠条形图中，我们可以得出地区和行业组是两个重要的分类特征。农畜牧业、软件业、人工智能、食品饮料、自然资源、科学技术类初创企业成功率很高。位于北美的初创企业最多、位于中亚和北非的初创企业成功率最高。

然而，我们的数据集包含了各种描述各轮融资金额的特征，其中哪些是与最终状态分类相关的相对重要的特征呢？我们借助随机森林算法进行特征选择！

1. X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 42)# test and train dataset

4. rfc = RandomForestClassifier(n\_estimators = 1000, random\_state = 42)
5. rfc.fit(X\_train, Y\_train)#training the model
6. rfc\_pred\_test = rfc.predict(X\_test)
7. **print**(classification\_report(Y\_test, rfc\_pred\_test, target\_names=['Class 1', 'Class 2', 'Class 3'])) # model is overfitting for class2 and bad at fitting 1 and 3

10. # Get numerical feature importances
11. importances = list(rfc.feature\_importances\_)
12. feature\_list = list(X.columns)
13. # List of tuples with variable and importance
14. feature\_importances = [(feature, round(importance, 2)) **for** feature, importance **in** zip(feature\_list, importances)]
15. # Sort the feature importances by most important first
16. feature\_importances = sorted(feature\_importances, key = **lambda** x: x[1], reverse = True)
17. # Print out the feature and importances
18. [**print**('Variable: {:20} Importance: {}'.format(\*pair)) **for** pair **in** feature\_importances];



根据重要性排序结果，我们决定选择行业类型、地区、风险投资、投资总额、天使投资和融资轮数作为执行分类任务的特征。

### 随机森林分类

我们提取出前面选择出来的变量，即公司所属行业类别，公司所在地区，风险投资，公司投资总额，天使轮投资，融资轮数并创建一个新数据框，来进行随机森林分类。

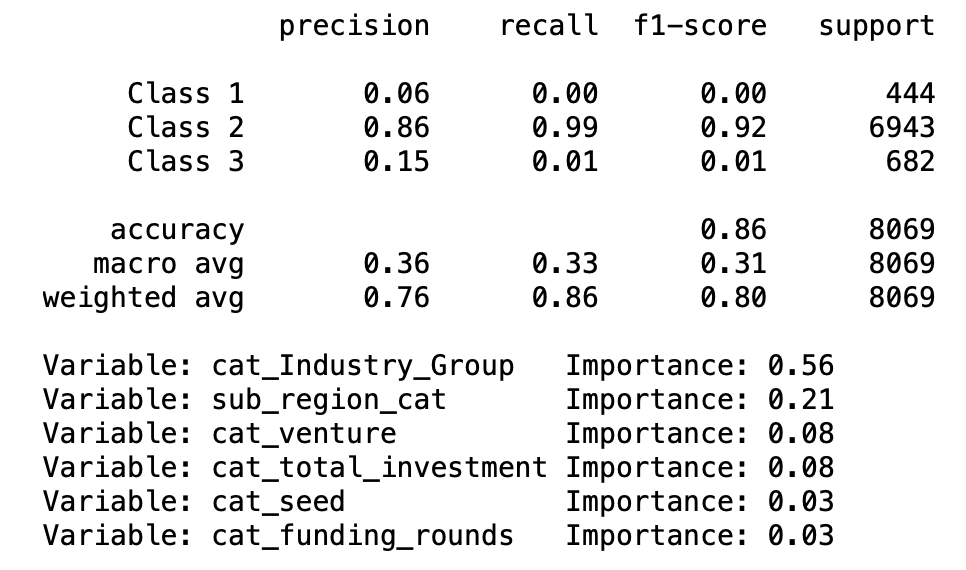
1. df3 = df1[['cat\_Industry\_Group','sub\_region\_cat','cat\_venture', 'cat\_seed',
2. 'cat\_funding\_rounds', 'cat\_total\_investment',
3. 'cat\_status']]
4. df3.dropna(inplace=True)
5. df3.head()
6. **print**(df3.dtypes)

我们训练随机森林模型，并报告其在测试集上的表现，发现该模型对于class2即公司运营中的标签预测表现较好，precision、recall和f1-score指标都较高，对于另外两个公司状态标签即公司倒闭（class1）和被收购（class3）的预测表现较差，原因可能是数据集中标签为这两类的数据点较少，而标签为运营中的数据量足够大。

同时，我们打印出变量重要性，发现最重要的四个变量分别为公司所属行业，公司所在地区，风险投资和总投资。

1. Y = df3.cat\_status
2. Y = label\_encoder.fit\_transform(Y)
3. X = df3.drop('cat\_status', axis = 1) #setting features
5. X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 42)# test and train dataset
7. rfc = RandomForestClassifier(n\_estimators = 1000, random\_state = 42)
8. rfc.fit(X\_train, Y\_train)#training the model
9. rfc\_pred\_test = rfc.predict(X\_test)
10. **print**(classification\_report(Y\_test, rfc\_pred\_test, target\_names=['Class 1', 'Class 2', 'Class 3'])) # model is overfitting for class2 and bad at fitting 1 and 3

13. # Get numerical feature importances
14. importances = list(rfc.feature\_importances\_)
15. feature\_list = list(X.columns)
16. # List of tuples with variable and importance
17. feature\_importances = [(feature, round(importance, 2)) **for** feature, importance **in** zip(feature\_list, importances)]
18. # Sort the feature importances by most important first
19. feature\_importances = sorted(feature\_importances, key = **lambda** x: x[1], reverse = True)
20. # Print out the feature and importances
21. [**print**('Variable: {:20} Importance: {}'.format(\*pair)) **for** pair **in** feature\_importances];



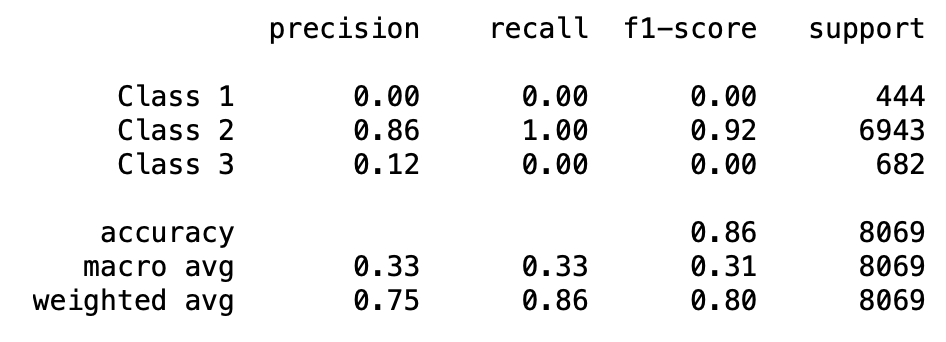
### PCA降维

我们在随机森林变量选择前（一共14个自变量）进行PCA，并查看PCA的结果如各个特征所占百分比、降维后方差损失等，再进行随机森林分类，查看对变量进行降维是否提升了随机森林分类模型的表现。

我们发现，PCA降维后一共产生了五个主成分，所占百分比分别为74.84%，9.62%，5.02%，3.83%，2.19，降维后方差损失为4.5%。

随机森林模型对class2标签（公司运营中）的预测表现没变，precision、recall和f1-score都没有改变，但对class1（公司倒闭）和class3（公司被收购）的预测表现变差了，其中class1标签precision变小，class3标签precision、recall和f1-score都有所下降。因此，在对这个数据集进行随机森林分类时，在变量选择前进行PCA不是一个很好的选择。

1. **from** sklearn.decomposition **import** PCA
3. Y = df2.cat\_status
4. Y = label\_encoder.fit\_transform(Y)
5. #print(Y.dtypes)
6. X = df2.drop('cat\_status', axis = 1) #setting features
7. #print(X.dtypes)
8. X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 42)# test and train dataset
10. pca = PCA(n\_components=0.95, random\_state=0)
11. X\_train\_pca = pca.fit\_transform(X\_train)
12. X\_test\_pca = pca.transform(X\_test)
14. rfc = RandomForestClassifier(n\_estimators = 1000, random\_state = 42)
15. rfc.fit(X\_train\_pca, Y\_train)#training the model
16. rfc\_pred\_test = rfc.predict(X\_test\_pca)
17. **print**(classification\_report(Y\_test, rfc\_pred\_test, target\_names=['Class 1', 'Class 2', 'Class 3'])) # model is overfitting for class2 and bad at fitting 1 and 3

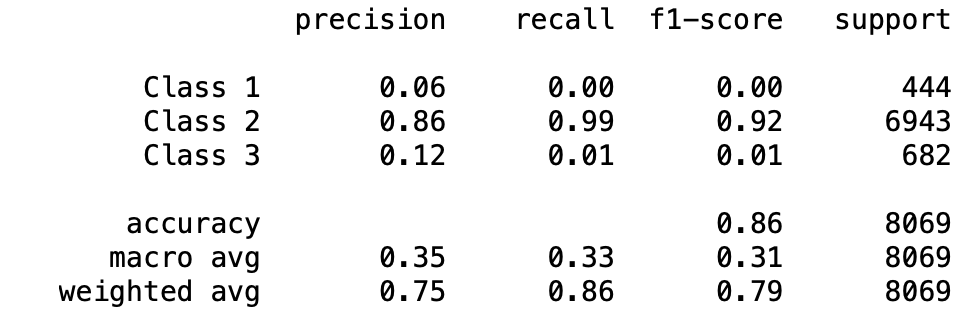


接下来，我们用随机森林变量选择后的变量（一共六个变量）进行PCA，并查看PCA的结果如各个特征所占百分比、降维后方差损失等，再进行随机森林分类，查看对变量进行降维是否提升了随机森林分类模型的表现。

我们发现，PCA降维后一共产生了两个主成分，所占百分比分别为94.78%，3.32%，降维后方差损失为1.9%。

随机森林模型对class1（公司倒闭）和class2标签（公司运营中）的预测表现没变，precision、recall和f1-score都没有改变，但对class3（公司被收购）的预测表现变差了，其中precision变小，recall和f1-score没变。因此，在对这个数据集进行随机森林分类时，在变量选择后进行PCA也不是一个很好的选择。

1. Y = df3.cat\_status
2. Y = label\_encoder.fit\_transform(Y)
3. X = df3.drop('cat\_status', axis = 1) #setting features
4. X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.2, random\_state = 42)# test and train dataset
6. pca = PCA(n\_components=0.95, random\_state=0)
7. X\_train\_pca = pca.fit\_transform(X\_train)
8. X\_test\_pca = pca.transform(X\_test)
10. rfc = RandomForestClassifier(n\_estimators = 1000, random\_state = 42)
11. rfc.fit(X\_train\_pca, Y\_train)#training the model
12. rfc\_pred\_test = rfc.predict(X\_test\_pca)
13. **print**(classification\_report(Y\_test, rfc\_pred\_test, target\_names=['Class 1', 'Class 2', 'Class 3'])) # model is overfitting for class2 and bad at fitting 1 and 3



## 机器学习结果可视化

此外，我们建立了网页https://laokaoya.github.io/MLAI-13-final-work/，以实现机器学习结果的可视化。在网页中可以进行模型选择和参数设定，之后点击按钮进行训练和预测，可以看到预测结果的可视化图像以及模型的ROC曲线。

本部分我们选取了Kaggle中美国地区初创企业的数据集，并对美国各州的企业成功率进行预测[[5]](#footnote-5)。我们目前的可选模型包括了Gradient Boosting、Decision Tree和Random Forest，对于每个模型可以设置深度、叶节点数、（模型数、学习率）。之后网页可以根据选择的模型和参数，给出美国地区各州企业成功率预测结果及其ROC曲线。

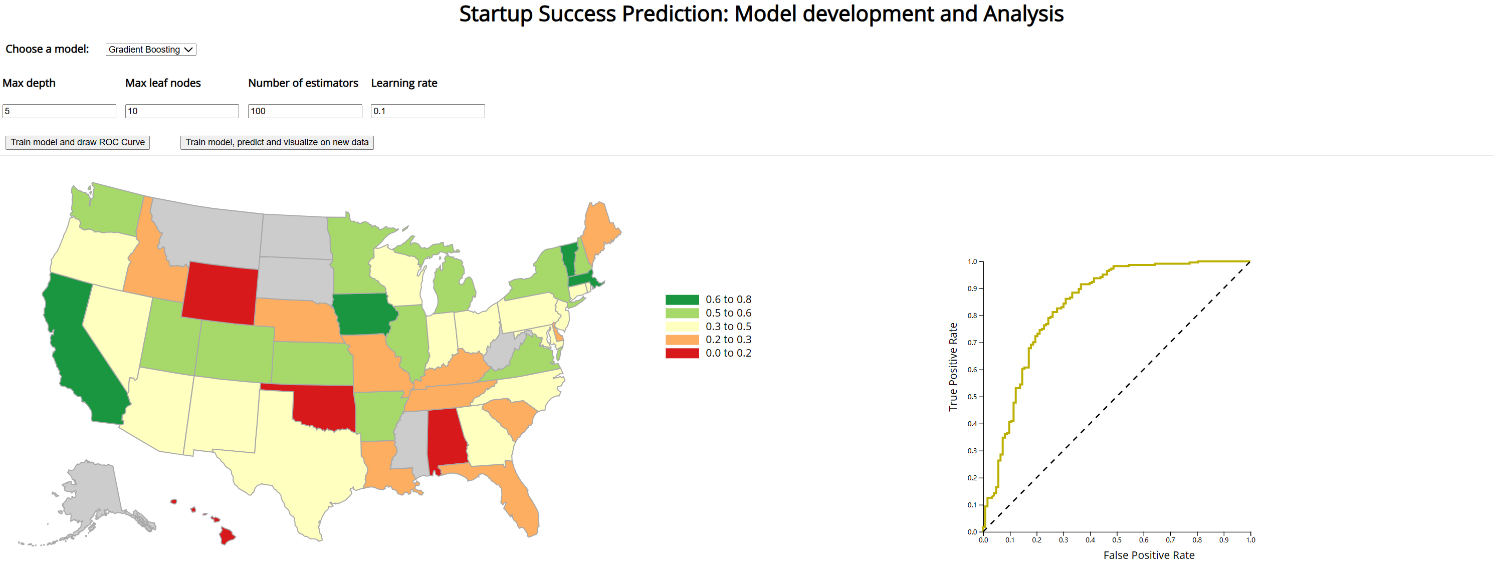


图3-2- 1 机器学习结果可视化

网页具体实现过程：我们首先搭建了flask服务器，并配置了相应的html和js文件。在js文件中我们设置了参数的默认值，并定义了两个按钮的事件函数，即地图预测结果的呈现和ROC曲线的绘制，具体可以参见代码附件。此外，在具体实现过程中，因为在生产模式中flask的公用端口需要SSL协议才能生效，否则会被浏览器拦截。这里我们通过内网穿透，将本地的127.0.0.1的flask端口穿透到公网，使得网页可以成功呈现。之后我们利用NSSM保持flask服务器在后台稳定运行，这样网页便搭建成功。

但需要承认的是我们的网页目前还不是很完善，首先因为内网穿透经常出现bug，导致网页稳定性较差。其次，网页可选模型数量较少，且呈现的结果只有地图和roc曲线，后续可以添加更多模型和可视化指标。

若网页出现问题，如按钮无法启动，可以与我们进行联

# 总结

本报告中，我们利用员工满意度、初创企业和企业破产情况等多个数据集，分别尝试了包括CNN、线性回归（拉索回归、岭回归）、K近邻等监督学习回归模型；决策树、随机森林、梯度提升、XGBoost、支持向量机等监督学习分类模型和自编码器、PCA等非监督学习模型，并对机器学习结果进行可视化，得到对创业者和投资者有价值的结论和建议。

github链接：

<https://github.com/laokaoya/MLAI-Group13/tree/main/MLAI%20Group13>

网页链接：

<https://laokaoya.github.io/MLAI-13-final-work/>

根据我们机器学习算法和模型的结果，我们得到以下结论：

1、适合创业的性格特征：本部分中，我们**实现了根据自拍照预测性格与创业的匹配度，以及具体的创业者类型**。并发现训练集中Fighters和Accomplishers人数最多，而Developers在数据集中没有发现，数量为0。

2、提高员工满意度：**工作量和工作时间是员工满意度的决定性因素，总体而言，员工同时间工作的项目数量越多、每个月的平均工作时长越长，员工满意度就会越高。然而，当项目数量多于5、工作时长超过300个小时，员工的满意度便会下跌**。这提示初创公司需要审视工作分配、项目管理和员工福利等方面的因素，以避免员工感到过度压力和疲劳，进而提高整体的员工满意度。

3、融资预测模型：我们的模型在不同轮次的融资性能随着融资轮次的递增而提升。具体而言，对于A轮融资而言，**首次融资的时间间隔以及公司所属的行业类型等因素对模型的影响最为显著**；在轮次B的融资决策中，**公司的成立时间和地理位置等因素起到了一定作用**；**C轮及之后的融资情况与B轮融资情况高度相关，反映出了B轮融资的重要性**。

4、投资者角度：对公司未来状态（运营、别收购或倒闭）预测的模型中，**最重要的变量为公司所属行业和公司所在地区**。

# 参考文献

1、创业性格部分：

1. 模型参考了Github上personality的开源框架，我们在此基础上做了改进，并添加创业性格预测等功能。
2. 模型参考论文的链接：

<https://www.nature.com/articles/s41598-023-41980-y>

2、数据集链接：

1. 员工满意度数据集来源为：<https://www.kaggle.com/datasets/redpen12/employees-satisfaction-analysis>
2. 初创企业数据集来源为：<https://www.kaggle.com/datasets/arindam235/startup-investments-crunchbase/data>
3. 企业破产情况数据集来源为：<https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction>

3、此外，报告的代码部分参考了Kaggle网站上的代码

1. BIG FIVE人格包括五方面，分别为openness（开放性）、conscientiousness（责任心）、extroversion（外倾性）、agreeableness（宜人性）、neuroticism（神经质）。这五种特质可以涵盖人格描述所有方面，是目前比较公认的人格分析模型。研究表明，即使经历了重大的生活事件，人格特质在成年后仍保持稳定，故而可作为重要的判定标准。 [↑](#footnote-ref-1)
2. 卷积层1：16个滤波器 \* (3 \* 3 + 1) = 2320个参数；卷积层2：16个滤波器 \* (3 \* 3 \* 16 + 1) = 2320个参数；全连接层1：784个输入 \* 500个神经元 + 500个偏置 = 392500个参数；全连接层2：500个输入 \* 500个神经元 + 500个偏置 = 250500个参数；全连接层3：500个输入 \* 100个神经元 + 100个偏置 = 50100个参数；输出层：100个输入 \* 5个神经元 + 5个偏置 = 505个参数。总体参数数量为 698,245。 [↑](#footnote-ref-2)
3. 这里迭代400次需要跑接近两小时，若做测试可降低迭代次数 [↑](#footnote-ref-3)
4. Agreeableness在论文中被细分为多块，权重差异大，取均值后属于中等影响，因而我们没有增加它的权重 [↑](#footnote-ref-4)
5. 因为做地图可视化需要用到地理信息的json文件，但我们只获取到了美国的，因此这里只选取美国数据进行研究 [↑](#footnote-ref-5)