Problem Chosen	2023	Team Control Number
A	MCM/ICM	2314151
	Summary Sheet	

打破世界

摘要：随着《世界大战》在社交媒体上的流行，越来越多的用户开始玩这款拼字游戏。时间和文字属性是如何影响报道数量、尝试分布以及其他与报道相关的信息的?因此，我们使用 2022 年的游戏数据进行了建模分析。

在构建模型之前，我们对给定的数据进行了清理和规范化，并识别了单词属性，如重复字母的数量、元音字母的数量、辅音字母的数量、共性和频率。为模型的构建和求解做了初步的准备。

首先，为了预测未来报告的数量，我们建立了一个基于先知的时间序列预测模型，考虑了趋势、季节性和节假日的影响。这些预测产生了 2023 年 3 月 1 日的一系列报告数字:[10355,18742]。关于报告数量的变化，在一周内，报告数量往往在周三最高，周末最低。在探索单词属性对难度报告比例的影响时，我们计算了两者的高阶偏相关系数，控制了单词属性之间的相互作用，发现元音字母的数量、非重复的数量和单词的通用性呈负相关。辅音字母的数量与非重复单词的数量呈正相关。

其次，开发了优化的多目标回归预测框架，探索单词属性对报告结果分布的影响。该框架选择了最优 lasso 回归来预测测试集，RMSE 为 0.80。预测“EERIE”的尝试次数分布为(0、4、17、34、30、13、2)。对各属性的重要性排序进行计算，发现辅音字母数、元音字母数和频率对报告结果分布的影响更为显著，影响因子分别为 4.226、3.993 和 1.253。

接下来，使用上述模型来预测 5 个字母单词集中每个单词的报告结果分布。然后，根据平均尝试次数，采用 K-means 将单词分为高(≥4.37)、中(4.13 ~ 4.37)和低(<4.13)三个难度类别，发现不同类别之间的重复数、重复最大值、患病率和频率存在显著差异。此外，还对每个属性的区间进行了划分。根据已建立的模型，“EERIE”是困难的。通过对不同难度词的属性区间进行匹配，模型的准确率为 91.36%，可以推断所建立的模型和划分的属性区间是合理的。

最后，灵敏度分析结果表明，我们的模型具有鲁棒性和可靠性。此外，对数据集的研究还揭示了《世界大战》受欢迎程度的下降和难度模式挑战比例的上升，并为《纽约时报》提供了恢复游戏受欢迎程度的建议。

关键词:世界大战，先知，高阶偏相关，多目标回归预测，K-means

目录

1 介绍

1.1 问题背景

填字游戏似乎一直与媒体有着密不可分的联系。从 2022 年 1 月开始，《纽约时报》的数字填字游戏 Wor- dle 在许多国家越来越受欢迎[1]。玩家如何玩《世界大战》?他们可以从 26 个字母中选择 5 个字母来构建

一个由五个字母组成的单词，在不超过六次尝试的情况下才能成功完成世界大战。玩家提交单词后，贴纸的颜色会改变。绿色是正确的字母，黄色是单词中出现但出现在错误位置的字母。游戏有两种模式:正常模式和困难模式。困难模式是在之前的尝试中找到正确的字母(绿色或黄色)，并且必须在随后的尝试中使用。

《世界大战》每天更新一次谜题，许多玩家在社交媒体上报告他们的分数。因此，当天报告分数的人数、参与难度模式的玩家人数、不同尝试完成谜题的玩家比例等数据都会被收集和统计。通过明智地使用这些可用的数据，我们可以解决一些有趣的问题。

1.2 问题的重述

考虑到背景信息，问题陈述中概述的约束条件和额外的指导，我们需要解决以下问题:

•任务 1:建立一个可以解释和预测报告结果数量变化的模型，并提供 2023 年 3 月 1 日报告结果数量的预测区间。此外，有必要对单词属性对玩家在硬模式下提交报告的比例的影响进行检查，并给出这种现象的基本原理。

•任务 2:开发一个预测报告结果分布的模型，并探索该模型和预测所具有的不确定性。

•任务 3:建立一个根据难度对单词进行分类的模型，并确定与单词分类相关的因素。该模型用于确定 EERIE 的难度，并讨论分类模型的准确性。

•任务 4:列举和解释该数据集中固有的其他值得注意的特征。

•任务 5:在给《纽约时报》拼图编辑的一封信中，对研究结果进行简要总结。

1.3 我们的工作

基于对问题的分析，我们提出了如图 1 所示的模型框架，主要由以下几个部分组成:

数据分析:对报告的数据进行处理，识别单词的特征。

预测建模:选择 Prophet 算法构建时间序列回归预测模型，并使用高阶偏相关分析找到各个属性的影响程度。

多目标回归预测框架的开发:利用该框架帮助我们选择 Lasso 回归预测模型。

难度区间划分:使用 K-means 算法将单词难度分为三类，并通过 Lasso 回归预测对分类结果进行验证。

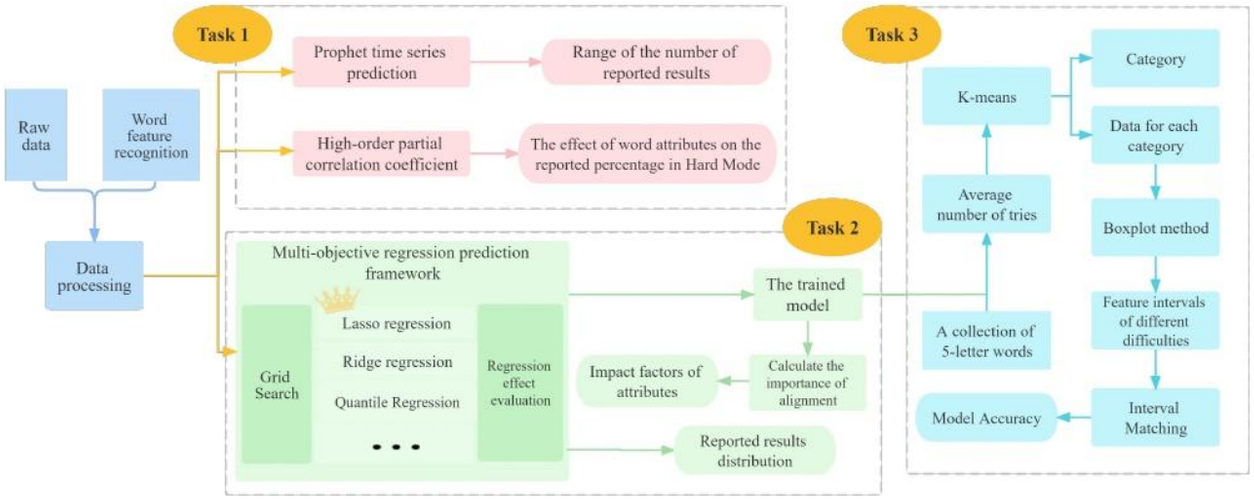


图 1:模型框架

2 模型的准备

2.1 假设

假设 1。假设问题中给出的用户数据是独立且相同分布的。

原因 1:这个假设保证了单个样本是相互独立的，避免了由于样本之间的关联而对建模过程的影响。

假设 2。假设预处理后的数据是可靠的。

原因 2:做这个假设是为了保证模型解的准确性。

假设 3。假设与游戏相关的外部环境不会突然改变

原因 3:外部因素保持稳定，保证预测模型稳定。

2.2 符号

表 1:注释

Symbol	Definition
s_j	Timestamp
k	Growth rate
δ_j	The amount of change in the growth rate on the timestamp
m	Offset amount
ϵ	Error term
N	Number of cycles in the seasonality model
D_i	Period before and after a holiday
κ_i	Range of holiday effects
P	Significance level

3 数据处理

3.1 数据清洗

Topic C 报告了过去一年里 world 的使用情况。但是，我们在这份报告中发现了很多无效数据。

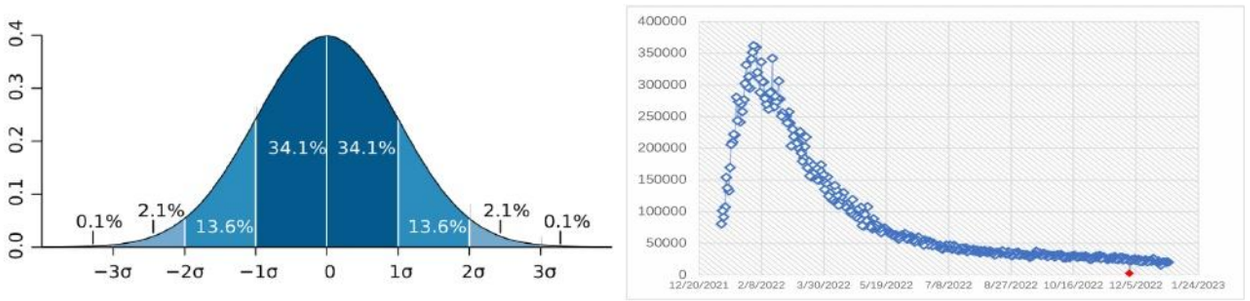
表 2:无效数据

Contest number	Word	Number of reported results	Number in hard mode	1 try	2 tries	3 tries	4 tries	5 tries	6 tries	7 or more tries (X)
525	clen	26381	2424	1	17	36	31	12	3	0
314	tash	106652	7001	2	19	34	27	13	4	1
540	naïve	21947	2075	1	7	24	32	24	11	1
473	marxh	30935	2885	0	9	30	35	19	6	1
207	favor	137586	3073	1	4	15	26	29	21	4

在上面显示的数据中，编号为 525 和 314 的两个单词不匹配游戏，因为它们的长度只有 4，所以我们推断数据集因为漏输入字母而出错。为了解决这样的问题，我们转而通过与人工智能算法进行比较，找到了与它们最相似的字母。编号为 540 的单词是由于字母拼写错误，应该是“幼稚”。我们搜索了单词数据库，发现编号为 473 的“marxh”这个单词是不存在的。然后，我们将这些单词的形状与数据库分析进行了比较，得出的结论是，正确的拼写应该是“marsh”。编号为 207 的单词在输入中有一个额外的空格，所以它也是一个离群值。我们可以删除多余的空格来得到正确的数据。

3.2 异常值剔除和标准化

我们使用 68-95-99.7 规则(3σ准则)筛选和拒绝异常值[2]。我们在 2022/11/30 的“study”一词的报告结果数据的数量中发现了一个异常，我们将其归零以带来它回到原来的数量级。



(a) 3σ criterion (b) Deviation point rejection

图 2:异常值拒绝

此外，我们还使用了 StandardScaler 数据归一化方法，该方法通过计算训练集的均值和标准差对训练集数据进行归一化[3]，见式

$$z = \frac{x - u}{s} \tag{1}$$

其中 X 为样本，u 为训练集特征列的均值，s 为训练集特征列的标准差。

3.3 单词属性的确定

在对报道结果进行预测的主题中，我们需要分析单词的属性。结合《世界大战》的玩法和回顾相关游戏的分析信息，我们将单词的属性分为以下几点。

1. 单词中出现字母的总频率:计算每个字母在候选单词列表中出现的频率。如果一个字母在 900 个单词中出现，则其出现频率为 900。然后根据总字母频率对候选单词进行排序，如果一个单词包含更多的高频字母，则排名第一。

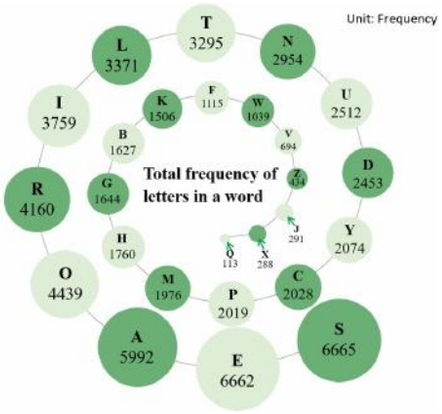


图 3:一个单词中字母的总出现频率

2. 单词中元音/辅音字母的数量:当玩家玩《世界大战》时，第一个单词通常被选择为 AUDIO 和 LEFTY，因为这包含了所有的元音字母:' AEIOU '。，在单词的构成中，元音字母很容易被知道，因此也很容易被猜出来。

3. 一个单词中不同元音/辅音字母的出现次数(无重复):元音/辅音字母的数量是我们推断的单词的属性之一，它会影响猜测单词的百分比。相应地，元音/辅音的出现次数在报告词的百分比中也是必不可少的。让我们以“there”为例。元音的个数是 2,The number of vowels(无重复)是 1，辅音的个数是 3，元音的个数(无重复)是 3。

表 3:字母属性

Word	Number of vowels	(no duplication)	Number of consonants	(no duplication)
there	2	1	3	3

4. 单词使用频率:我们在日常生活中使用很多单词，有些常见，有些不太熟悉。人们往往更容易猜测常用词。因此，我们将这个游戏中涉及到的所有五个字母的单词的使用频率进行了列出和排序。下表显示了部分排序后的数据。

表 4:字母共性排名

Word	Times	Rank	Word	Times	Rank	Word	Times	Rank
which	0.002044	1	their	0.001954	2	would	0.001711	3
about	0.001407	4	could	0.001296	5	there	0.001273	6

5. 重复字母的数量和最大重复次数:There may be

一个单词的构成中有几个重复的字母。这种情况并不常见，但字母重复的次数也会影响玩家猜单词的成功率。因此，我们将报告中单词中字母的这两个属性算作它们的特征。

表 5:字母属性

Word	Number of duplicate	Maximum number of repeats	Word	Number of duplicate	Maximum number of repeats
cross	1	2	exist	0	0
glass	1	2	apply	1	0

4 任务 1

4.1 先知算法

4.1.1 算法背景

虽然近年来神经网络模型越来越流行，但这种模型通常需要大量的数据进行训练。只有 400 个左右数据的数据集并不是考虑神经网络模型的好地方。

综合考虑，我们决定使用 Prophet 模型，这是一种基于加性模型的算法，用于预测具有季节性、趋势和假日等特征的时间序列数据。此外，Prophet[4]在处理非平稳时间序列和异常值等问题方面具有很强的鲁棒性。对于这个数据集，Prophet 是一个不错的选择。

4.1.2 基于 Prophet 算法的预测模型构建

1. 先知算法原理

Prophet 算法的原理如下:

$$y(t) = g(t) + s(t) + h(t) + \epsilon \tag{2}$$

G (t)为趋势项，表示非期间的时间序列趋势。S (t)表示期间项，一般以周或年为单位计量。H (t)表示假期期，它表示时间序列中那些潜在的非周期性假期对预测值的影响。ϵdenotes 为误差项或残差项，表示模型无法预测的波动，ϵfollows 为高斯分布。

Prophet 算法对模型的三个组成部分分别建模，然后将它们组合起来生成预测数据。

2. 趋势项模型

Prophet 对趋势部分的实现主要应用了两种模型，一种是饱和增长模型，另一种是分段线性模型。

•饱和增长模型

饱和增长模型，又称 logistic 增长模型，是用来描述增长率逐渐降低并最终趋于稳定的系统的模型。

$$g(t) = \frac{C(t)}{1 + \exp(-(k + a(t)^T\delta) \cdot (t - (m + a(t)^T\gamma)))} \tag{3}$$

C(t)表示承载能力，这是一个时间函数，限制了可以增长的最大值。K 表示生长速率。

•分段增长模型

$$g(t) = (k + a(t)^T\delta) \cdot t + (m + a(t)^T\gamma) \tag{4}$$

值得注意的是，分段线性函数和逻辑回归函数最显著的区别在于，在分段线性函数中 y 的设置是不同的。

$$\gamma_{j=-s_j\delta_j} \tag{5}$$

该模型定义了与增长率 k 的变化相对应的点，称为 n 个变化点。变化点先验尺度被定义为增长趋势模型的灵活性。

3. 季节性趋势模型

由于时间序列可能包含多天、多周、多月、多年和其他周期类型的季节性趋势，因此可以使用傅立叶尺度来近似这种周期性质。傅立叶级数如下所示。

$$s(t) = \sum_{n=1}^{N} (a_n cos(\frac{2\pi nt}{P}) + b_n sin(\frac{2\pi nt}{P})) \tag{6}$$

N 表示希望在模型中使用的周期数。较大的 N 值允许拟合更复杂的季节函数。然而，它们也会引入更多的过拟合问题。

4. 假日效应模型

在自然环境中，节假日可以显著影响时间序列。每个假期并不总是相同的，因此不同假期在不同时间点的影响被视为独立的模型。对于第 i 个假期，Di 表示假期前后的时间段。

为了表示假期效应，需要相应的指标函数，并需要参数κi 来表示假期效应的范围。

假设有 L 个节假日，则节假日效应模型为:

$$h(t) = Z(t)\kappa = \sum_{i=1}^{L} \kappa_i \cdot 1_{\{t \in D_i\}} \tag{7}$$

$$Z(t) = (1_{\{t \in D_1\}}, \cdots, 1_{\{t \in D_L\}}) \text{ and } \kappa = (\kappa_1, \cdots, \kappa_L)^T$$

4.1.3 参数设置

我们为趋势项模型选择一个基于分段线性函数的趋势项。我们将 n 个变化点设置为 25，并将变化点先验尺度设置为 0.05。对于季节性趋势，我们将季节性先验尺度设置为 10。对于假日效应，我们将假日先验尺度设置为 10。此外，我们将间隔宽度设置为 0.80,mcmc 样本设置为 0，不确定性样本设置为 1000。

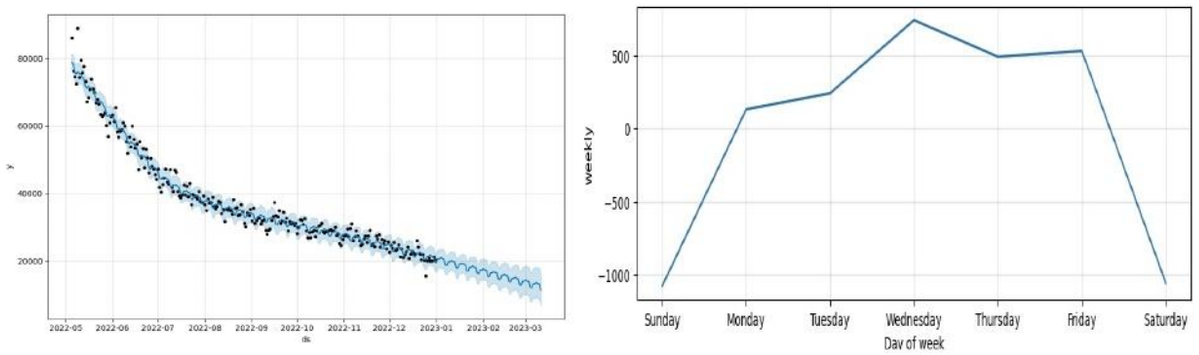
4.1.4 结果

我们使用图 a 所示的参数值建立模型。在选择数据范围时，通常需要考虑数据的数量级。之所以从相同的数量级开始取数据，是为了避免数据偏置和误差的影响，保证数据的准确性和可靠性。因此，我们对原始数据进行了筛选，选取了 2022 年 5 月 5 日之后的数据。我们最终得到的结果如下。

Table 6: Predicted results

ds	yhat	yhat_lower	yhat_upper
2023-03-01	14425.58926	10355.27753	18741.54302

该模型预测 2023 年 3 月 1 日的报告数量范围为 10355 至 18742。



(a) Prediction Chart (b) Weekly Trend Chart

图 4:预测图和趋势图

上图显示了时间序列趋势和每周季节性。图 a 显示了报告数量和未来预测的总体趋势。从 2022 年 5 月 5 日开始，报告数量减少，下降幅度逐渐变小。此外，预测 2023 年 1 月 1 日后 70 天的报告数量间隔。

图 b 显示了每周的周期格局，周三玩《世界》的人数明显增加。周末报告的数量往往更少。

4.2 高阶偏相关分析模型

4.2.1 使用 Pearson 相关系数进行相关分析

这个模型要解决的问题是分别对单词的属性和难度模式的百分比进行相关性分析。在数据处理中已经对单词的属性进行了分类，但是在对这个问题的数据进行分析之后，我们发现单词的属性之间存在很强的相关性。因此，我们首先进行了相关性测试在不控制其他变量的情况下，检查每个属性与难度模式百分比的 Pearson 相关系数;

表 7:Pearson 相关系数分布

Attributes of words	R	P
Number of vowels	0.083460383	0.093068989
Number of vowels(non-repetition)	0.05075013	0.307685109
Number of consonants	-0.083460383	0.093068989
Number of consonants(non-repetition)	-0.106284281	0.032271311
Word commonness	0.094056604	0.058285504
The sum of the frequencies of letters	0.008176597	0.869535404

从上表中可以看出，各属性相关检验的显著性检验 p 值与 percent hard 的显著性检验 p 值几乎都是弱相关的，只有单词中字母词频的总排名是强相关的。这个结果是不令人满意的。我们再次分析了单词的属性，发现这些属性之间存在很强的相关性，属性之间的影响不容忽视。综上所述，我们选择了高阶偏倚相关分析的算法，用难度模式对单词中各个属性所占的百分比做相关分析来解决这个问题[5]。

4.2.2 高阶偏相关分析模型的建立

(1)一阶偏相关系数:剔除剩余一个变量的影响后，计算出三个变量中任意两个变量的偏相关系数，称为一阶偏相关系数，公式如下:

$$r_{ij\cdot h}=\frac{r_{ij}-r_{ih}r_{jh}}{\sqrt{\left(1-r_{ih}^{2}\right)\left(1-r_{jh}^{2}\right)}} \tag{8}$$

在此式中，r_{ij} 为变量 x_i 和 x_j，之间的简单相关系数，r_{ih} 为变量 x_i 和 x_h，之间的简单相关系数，r_{ih} 为变量 x_j 和 x_h，之间的简单相关系数。

(2)高阶偏相关系数:一般情况下，如果有 k (k > 2)个变量 x1、x2、…， xk，则任意两个变量 x_i 和 x_j，的高阶样本(g≤k-2)偏相关系数公式为:

$$r_{ij\cdot l_1 l_2 \cdots l_g} = \frac{r_{ij\cdot l_1 l_2 \cdots l_{g-1}} - r_{il_g\cdot l_1 l_2 \cdots l_{g-1}} r_{jl_g\cdot l_1 l_2 \cdots l_{g-1}}}{\sqrt{(1 - r^2_{il_g\cdot l_1 l_2 \cdots l_{g-1}})(1 - r^2_{jl_g\cdot l_1 l_2 \cdots l_{g-1}})}} \tag{9}$$

式中，右侧均为 g-1 阶的偏相关系数。

4.2.3 结果分析

在高阶偏相关分析中，我们需要控制不相关变量，以消除其他变量对所研究变量的影响。得到的结果如下:

表 8:高阶偏倚相关结果分布

Attributes of words	High-order partial correlation coefficient	P
Number of vowels	-0.319578	4.37009E-11
Number of vowels(non-repetition)	-0.271561	2.72256E-08
Number of consonants	0.319578	4.37009E-11
Number of consonants(non-repetition)	0.306104	3.00056E-10
Word commonness	-0.080926	0.103481297
The sum of the frequencies of letters	-0.133207	0.007197084

如上表所示，上述 6 个词的属性对应的 p 值都远小于 0.05。因此，可以认为结果具有统计显著性。

一般来说，高阶偏相关系数取绝对值后，在 0-0.09 时为无相关性，在 0.1-0.3 时为弱相关性，在 0.3-0.5 时为中度相关性，在 0.5-1.0 时为强相关性。

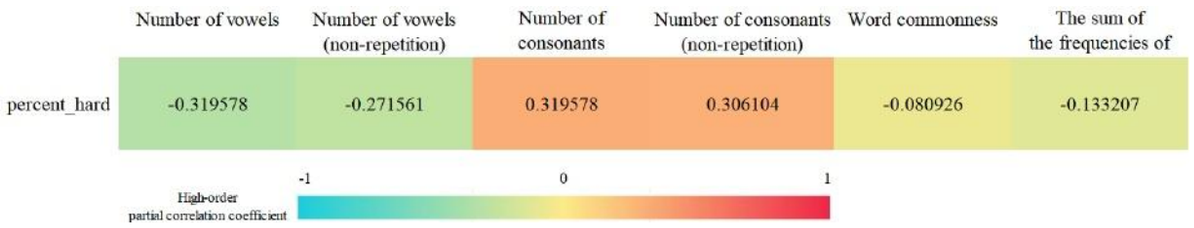


图 5:百分比硬的词属性高阶偏相关热图

通过分析数据以及热图的结果，我们得到了以下词属性与硬模式百分比的相关性:

表 9:词属性相关结果分布

Attributes of words	Degree of correlation
Number of vowels	Moderate negative correlation
Number of vowels(non-repetition)	Weak negative correlation
Number of consonants	Moderate positive correlation
Number of consonants(non-repetition)	Moderate positive correlation
Word commonness	No correlation
The sum of the frequencies of letters	Weak negative correlation

5 任务 2

5.1 多目标回归预测框架

在这一部分中，我们需要解决一个预测问题，即我们所构建的模型能够预测未来某个日期与《世界之战》谜题相关的用户粘性百分比。我们目前拥有《世界之战》过去一年的用户报告数据，

我们将每天的单词划分为几个相关属性，并考虑到时间的影响，因为用户在周末和工作日完成游戏的方式不同。

由于问题中给出的数据量非常小，我们放弃使用神经网络进行预测，而是使用优化的多目标回归预测框架[6]对其进行处理。

使用多目标回归预测框架来解决预测问题的具体过程如下:

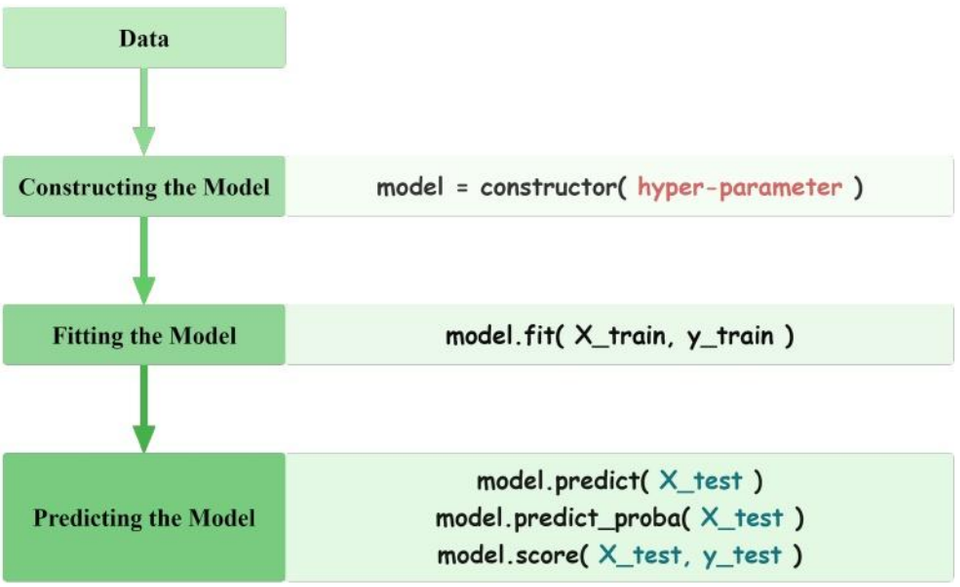


图 6:多目标回归预测框架的操作流程

我们的框架目前支持偏最小二乘回归算法、基于贝叶斯线性回归的超参数调优和特征选择算法、弹性网络回归模型、LASSO 回归算法模型、脊回归算法和高性能分位数回归算法模型。所有这些回归模型在能够处理多个分类任务并适合增量训练的同时，在小数据上都能表现良好。此外，它们良好的自适应能力、自学习能力、泛化能力也符合我们的要求。

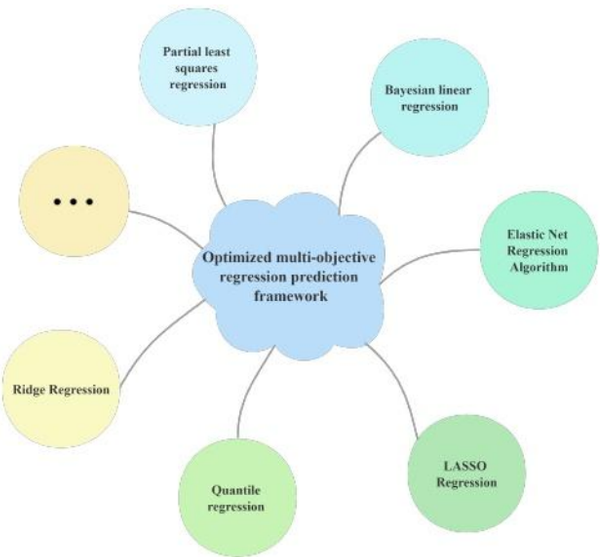


图 7:算法集成评价分析框架

5.2 预测模型的建立

(1)建立多目标回归预测框架

多目标回归预测框架将数据分为训练集和测试集，并对数据进行归一化。它通过网格搜索识别最优参数，并基于均方误差的均值来评估模型的准确性，进而确定最佳算法模型[7]。然后，多目标回归预测框架使用最优算法对数据进行预测，并以逆归一化的原始尺度返回数据。对于我们的数据集，基于评价函数的算法评价框架的输出如下图所示:

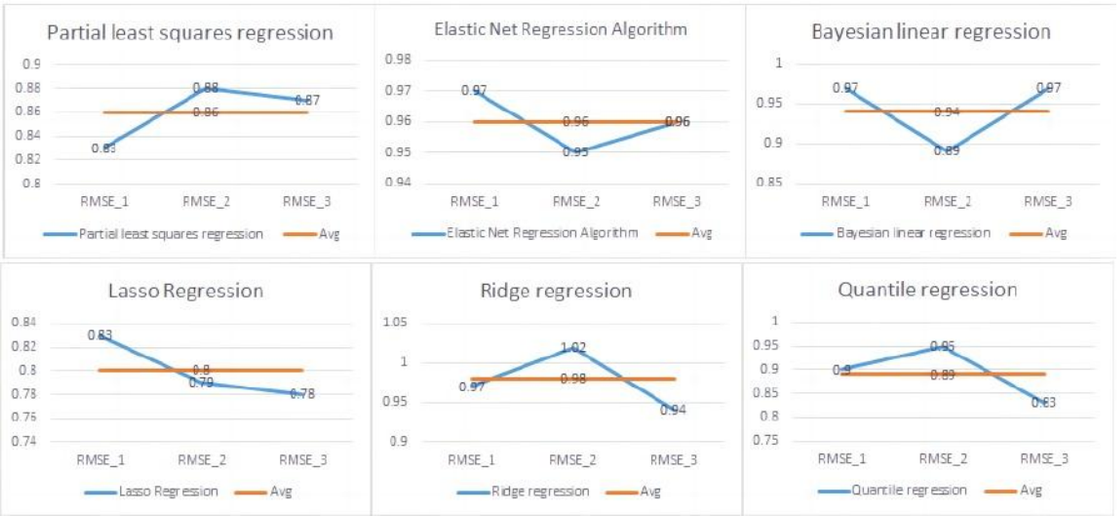


图 8:基于评价函数的多目标回归预测框架的输出

从上图的输出可以看出，LASSO 回归算法的均方根误差最小，其多次重复拟合的总平均均方根误差仅为 0.8，可以看出我们的回归模型能够很好地预测数据。

(2) Lasso 回归模型的建立

Lasso 回归(LASSO，最小绝对收缩和选择算子)是在残差平铺和最小化中添加 L1 奇偶性的惩罚项:

$$\min \sum e_i^2 + \lambda ||\hat{\beta}||_1 = \min \sum (y_i - \hat{y}_i)^2 + \lambda \sum_{u=1}^{k} \left| \hat{\beta}_u \right| \tag{10}$$

因为 L1 参数化是以绝对值的形式出现的，所以在零点处是不可导的。因此，它不再具有解析解，可以使用梯度下降(准确地说，是次梯度算法)来求解。岭回归不能消除变量。LASSO 回归的优点在于它可以产生稀疏性，它可以将一些不显著的回归系数降为零，从而达到消除变量的目的。它的损失函数为:

$$J(w,b) = \frac{1}{2m} \underset{w,b}{\operatorname{argmin}} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 + \alpha \sum_{i=1}^{n} \|w_i\| \tag{11}$$

5.3 词预测-EERIE

首先，EERIE 这个词可以使用上面建立的特征工程来归类。EERIE 这个词的相关属性如下。

表 10:单词 EERIE 的相关属性

Date	Number of vowels	Number of vowels (non-repetition)
01/03/23	4	2
Word	Number of consonant	Word commonness
EERIE	1	224
Contest number	Number of consonant (non-repetition)	The sum of the frequencies of letters
620	1	27903

在上述多目标回归预测框架之后，我们确定了一个改进的 Lasso 回归模型来实现对给定日期的词相关百分比的预测。我们对 EERIE 这个词进行了 Lasso 回归预测，以预测 2023 年 3 月 1 日报告的百分比。我们取多次拟合的平均值作为最终预测，使结果更加准确。

表 11:精确的结果——可怕

1 try	2 tries	3 tries	4 tries	5 tries	6 tries	7 or more tries (X)
0.023996469	3.60053273	16.9184627	33.8665962	30.5875392	12.9104259	2.09258948

最后对预测百分比结果进行四舍五入，得到最终预测结果:

表 12:最终结果- eerie

Date	Contest number	Word	1 try	2 tries	3 tries	4 tries	5 tries	6 tries	7 or more tries (X)
01/03/23	620	EERIE	0	4	17	34	30	13	2

5.4 特征影响程度分析

为了得到对我们的预测模型有更显著影响的词属性，我们提出了一种基于排列重要性的分析。

首先，我们得到一个训练好的套索模型。接下来，我们对某一列数据的值进行破坏，然后对得到的数据集进行预测。预测值与真实的目标值一起使用，来计算由于随机排序导致损失函数提升了多少。模型性能的衰减量代表了无序列的重要性。然后，我们恢复无序列，并在下一列数据上重复前面的操作，直到计算出每一列的重要性。

为了使结果更具通用性，我们运行三次分析，分析不同属性的得分，并将最终的拟合结果总结为下图，这有助于我们分析单词中不同属性对模型的影响程度。

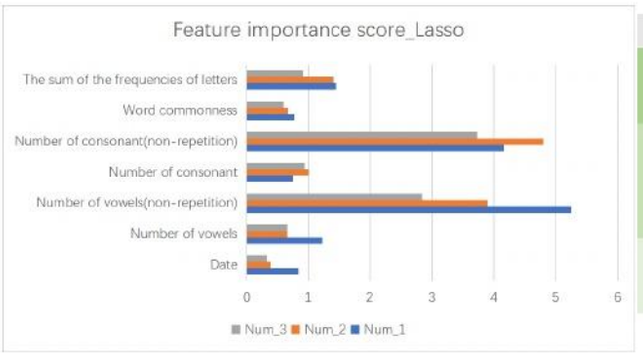


图 9:特性重要性评分套索

Weight	Feature
4.226±0.502	Number of consonant(non-repetition)
3.993±1.211	Number of vowels(non-repetition)
1.253±0.558	The sum of the frequencies of letters
0.896±0.187	Number of consonant
0.844±0.385	Number of vowels
0.678±0.118	Word commonness
0.517±0.314	Date

图 10:单词的 7 个属性的特征重要性的热图

在我们的模型中，我们观察到单词中辅音数量(不包括重复辅音)和元音数量(不包括重复元音)这两个属性之间存在很强的相关性。此外，我们还发现单词中所有字母出现的频率、辅音数量和元音数量的总和之间存在适度的相关性。相比之下，单词常见度和日期之间的相关性较弱。

5.5 模型信度分析

为了评估我们的预测模型的可靠性，我们使用了两种方法。首先，我们计算模型预测结果的均方根误差来衡量预测的偏差

来自实际数据的结果。其次，我们使用预测模型对给定数据进行预测，然后将预测结果与实际数据进行比较，以评估预测结果的准确性。

均方根误差是通过计算预测值与实际值之间偏差的平方和除以观测值个数 n 的比率，然后取平方根得到的。均方根误差的计算公式如下。

$$RMSE = \sqrt{\frac{1}{n} \cdot \sum (y_i - y_{ihat})^2} \qquad (12)$$

其中 n 为样本个数，yi is 为第 i 个样本的实际值，yihat is 为第 i 个样本的预测值。RMSE 的单位与目标变量的单位相同。我们拟合并图形化地分析拟合的 lasso 回归模型结果的均方根误差。

根据图 7 lasso 回归的输出图，我们发现三个解的 RMSE 都不超过 0.85，平均 RMSE 为 0.8。一般来说，当 RMSE 不超过 2 时，预测精度被认为是高的。

下面是一个对比 5 个成功猜测单词的趋势程度的图表示例:

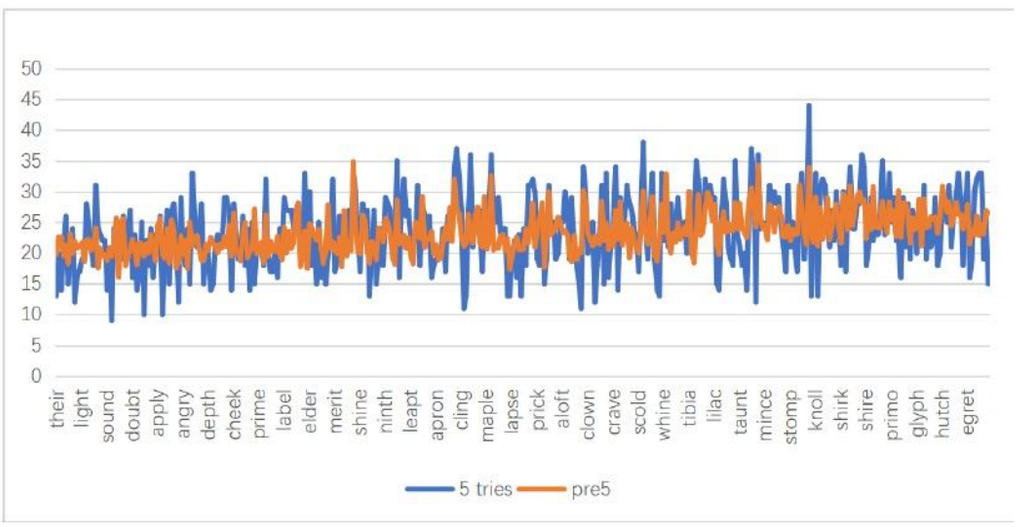


图 11:5 次猜对成功的单词的趋势程度对比

蓝色曲线为原始数据 5 次猜对的百分比。相比之下，橙色曲线是我们的预测模型预测的 5 次成功的百分比，可以看出其预测结果的偏差在可接受的范围内。

6 任务 3

我们认为“平均尝试次数”可以表示一个单词的难度等级，但由于问题中给出的数据集太小，会影响聚类分析的性能。因此，我们在去掉时间变量后，将 Task 2 的框架重新拟合到预测字典中 12974

个单词的尝试次数分布中，并利用这个分布来寻找平均尝试次数。在此之后，使用 K-means 聚类对平均尝试次数进行聚类[8]。

6.1 K-means 聚类算法

K-means 算法通过选择合适的距离公式来度量不同数据对象之间的相似度[9]。数据之间的距离与相似度成反比，即相似度越小，距离越大。K-means 算法首先需要从给定的数据对象随机指定初始簇数 k 和对应的初始簇中心 C，并计算初始簇中心到其余数据对象的距离[10]。在本文中，我们选择欧几里德距离。聚类中心到空间中其他数据对象的欧几里德距离公式为:

$$d(x, C_i) = \sqrt{\sum_{j=1}^{m} (x_j - C_{ij})^2} \tag{13}$$

其中 x 为数据对象，Ci 为第 i 个聚类中心，m 为数据对象的维数，xi, Cij 为数据对象 x 和聚类中心 C 的第 j 维的属性值 Ci。

根据相似度的欧几里德距离度量，将与聚类中心相似度最高的目标数据分配给 ci 的聚类，然后对 k 个聚类中的数据对象进行平均，形成新一轮的聚类中心。其计算公式为:

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} |d(x, C_i)|^2 \tag{14}$$

6.2 参数选择

•剪影系数法:该方法通过计算每个样本点的剪影系数来评估聚类质量。其中剪影系数综合了簇内距离和簇间距离的大小，取一个介于[-1,1]之间的值，越接近 1 意味着聚类效果越好。

•Davies-Bouldin 指数法(davis - bouldin index，简称 DBI):该方法通过计算每个聚类中所有点到聚类中心的平均距离与不同聚类中心之间的最短距离之比来评价聚类质量。

经过分析，我们根据难易度将单词分为 3-5 类，参数如下:

表 13:参数

Number of clusters	Silhouette Coefficient	DBI
3	0.545003549	0.562683379
4	0.536022115	0.561324728
5	0.523517926	0.558397471

如上表所示，随着集群数量的增加，剪影系数和 DBI 都略有下降。为了提高识别单词难度的准确性，我们选择聚类为 3 类。

6.3 聚类结果

基于以上分析，我们选择 K-means 算法将数据集聚为三类，得到如下聚类结果。

总体结果:

表 14:汇总结果

	Number of duplicate	Maximum of repeats	Number of vowels	Number of vowels(non-repetition)	Times
0	0.519623	0.63264	1.859318	1.546858	0.000002
1	0.019896	0.023617	1.77483	1.759786	0.000009
2	1.210897	1.182715	1.748239	1.354157	0.000001
	Rank time	Weighted sums	Number of consonants	Number of consonants(non-repetition)	
0	18842.21231	423.226591	3.140682	2.933519	
1	20294.22727	403.383609	3.22517	3.220317	
2	16888.876	451.443208	3.251761	2.434946	

难度区间划分:

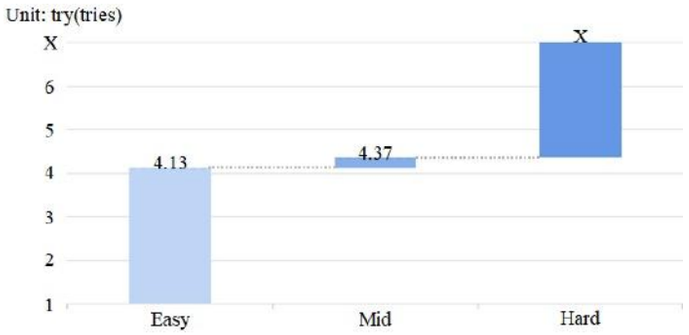


图 12:单词难度区间划分

对结果的分析表明，在不同的分类中，Number of duplicate、Maximum of repeats、患病率(患病率)和 Frequency 差异很大。根据四分位数法，正常值一般定义为大于 QL －kIQR 或小于 QU + kIQR。这里，我们 k 取 0.5，具体难度区间如下图所示。

表 15:具体难度区间

	Number of duplicate	Maximum of repeats	Normality	Frequency
easy	(0, 0)	(0, 0)	(0.000009, 0.002044)	(16767.75, 33314)
mid	(0, 1)	(0, 2)	(0.000003, 0.000012)	(14969.25, 29574.25)
hard	(1, 3)	(0, 2)	(0, 0.000004)	(6568, 28380.5)

6.4 词间隔识别——EERIE

“EERIE”属于难度区间。

我们将单词“EERIE”带入第 2 部分构建的 Lasso 回归预测模型，得到它的平均尝试次数为 4.384952125887，这意味着它属于难词。很容易发现，它的属性 Number of duplicate、Maximum of repeats、Normality 和 Frequency 都在我们已经分类的困难范围内。因此，我们认为“EERIE”这个词属于难音程。

6.5 模型信度分析

为了评估我们的分类模型的准确性，我们使用了从 2022 年 1 月 7 日到 2023 年 2 月 16 日的 405 个数据，这些数据已经被确定用于验证。验证过程如下。

首先，我们计算每个单词的平均猜测次数。我们根据前一节描述的聚类结果，将平均猜出的单词数划分为不同的间隔。然后，我们对每个字母的每个属性进行计数，比较每个属性击中了哪个难

度区间级别。命中次数最高的难度，就是特征所识别的难度。如果分类结果与单词命中间隔的结果相同，则认为我们的模型是准确的。算法实现如下:

Algorithm Classification verification algorithm based on interval matching

Input: Interval,$Word_p$,S_a

Output: Classification result,Forecast result

for $i = 1 \rightarrow len(data)$ **do**
  $Word_p1 \Leftarrow \emptyset, Word_p2 \Leftarrow \emptyset$ //Initialize the eigenvalue
  for $j = 1, 2, ..., a$ **do**
    if s_j = Interval **then**
      weight++ //Increasing weight
    else
      weight– //Decreasing weight
    end if
    Sum = Total(weight) //Weight summation
    Interval(S_j) //Partition by weight
  end for
end for
for $i = 1 \rightarrow a$ **do**
  Compare result and Interval(S_j)
  if result $\epsilon Interval(S_j)$ **then**
    Weight(S_j)
  else
    continue
  end if
end for
return RMSE

我们的模型的精度计算为 91。3649025%。对于小规模数据分类，我们的模型具有很高的准确率。

7 数据有趣的方面

报告结果的数量与困难模式中报告的分数百分比之间的关系

我们试图找到这个数据集的其他一些有趣的特征，但我们不知道从哪里开始。所以我们在社交媒体上搜索世界，以获得对这款游戏的具体看法。在这期间，我们突然发现大部分关于这款游戏的报道结果都来自 2022 年上半年，而在下半年，这款游戏的热度明显下降。因此，我们将玩家报告的数量、困难模式中报告的分数百分比和日期之间的关系进行关联和绘制。由于困难模式的百分比与报告结果的数量之间的差异太大，我们将困难模式中报告的分数百分比的值乘以 106 ，以便使关系图更加明显。

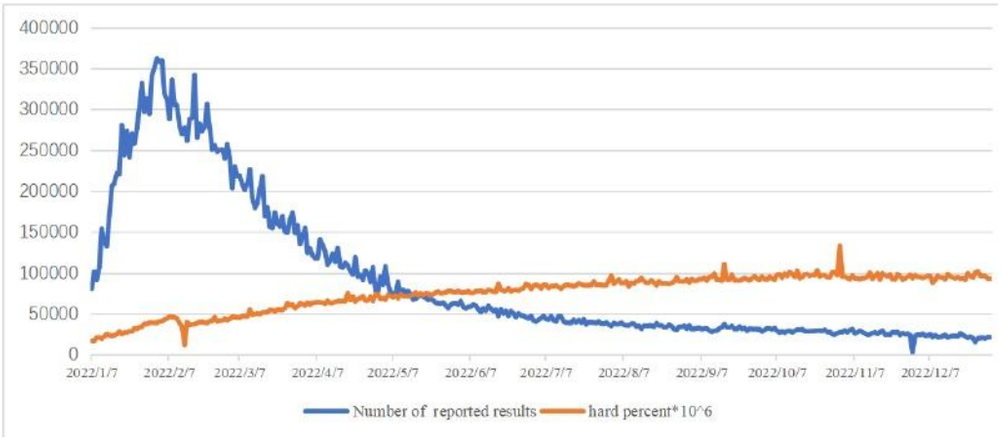


图 13:硬模式百分比 vs. Date vs. Num of reported

从上图中我们发现，随着报告结果的数量开始增长，硬模式的百分比稳步增长。然而，当报告的结果数量达到顶峰时，处于困难模式的百分比直线下降。我们假设有大量不熟悉《世界大战》的新玩家涌入，他们不太可能尝试难度模式，这导致了难度模式的比例大幅下降。随着时间的推移，世界大战每日报告结果的数量逐渐减少，但硬模式的百分比稳步增加。作为回应，我们认为《世界大战》拥有忠实的粉丝基础，他们经常玩《世界大战》，并且更喜欢复杂模式。这说明热心粉丝的游戏能力也在慢慢提升，同时也说明这款游戏有助于提高玩家的英语水平。

平均成功次数、报告结果次数和世界日期之间的关系

接下来，我们分析平均成功次数、报告总数和世界日期之间的关系。为了使散点图更加可见，我们将 Average number of success 的值乘以 10-5。

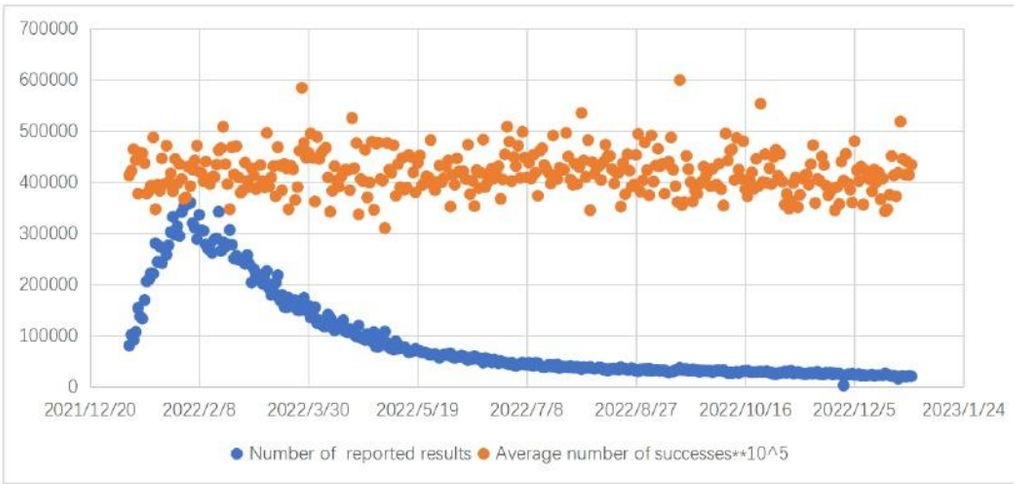


图 14:平均成功次数 vs.日期 vs.报告次数

如图所示，Average number of success 的值大多分布在 4 到 5 之间。这说明游戏的难度几乎保持不变，几乎每个单词都被猜出 4-5 次。这支持了我们之前的有趣发现，即游戏有助于提高玩家的英语水平，玩家玩游戏的时间越长，他们就越愿意尝试难度模式。

8 敏感度分析

在第二部分中，我们人为地指定测试集(POT)的比例为 20%，这个值的变化会影响模型的训练。敏感性分析的结果如表 7 所示。

表 16:敏感性分析

	POT =0.2	POT =0.1	POT =0.15	POT =0.25	POT =0.3
Partial least squares Regression	0.86	0.87(1.16% ↑)	0.93(8.14% ↑)	0.89(3.49% ↑)	0.97(12.79% ↑)
Elastic Net Regression	0.96	0.94(2.08% ↓)	0.95(1.04% ↓)	0.92(4.17% ↓)	0.91(5.21% ↓)
Bayesian linear regression	0.94	0.97(3.19% ↑)	1.01(7.45% ↑)	0.89(5.32% ↓)	0.96(2.13% ↑)
Lasso Regression	0.80	0.83(3.75% ↑)	0.84(5.00% ↑)	0.86(7.50% ↑)	0.92(15.00% ↑)
Ridge regression	0.98	0.91(7.14% ↓)	0.93(5.10% ↓)	0.93(5.10% ↓)	0.92(6.12% ↓)
Quantile regression	0.89	0.92(3.37% ↑)	0.82(7.87% ↓)	0.90(1.12% ↑)	1.05(17.98% ↑)

结果显示，通过改变测试集的比例，最大变化率为 10%，而各模型的 RMSE 均小于 1.0，证明我们的模型对该参数在自身 0.5 倍范围内的变化不敏感，我们的模型鲁棒可靠。

9 优势和劣势

9.1 强度

先知模型为小数据集提供了出色的预测，并可自由调整参数，从而有助于更准确地预测我们将在未来日期拥有的报告数量。

高阶偏相关分析可以控制不相关的变量对一个变量进行相关性分析，这有利于我们消除词之间的影响，更准确地分析变量的影响程度。

Lasso 回归模型在处理多个分类任务的同时，可以在小数据上表现良好，适合增量训练，在小数据集上做出更准确的回归预测。

排列重要性可以在拟合模型上执行数据中断和重排分析，以更准确地推导出每条数据的贡献。

9.2 缺点

多目标回归预测运行时间长，求解结果需要简单的人工筛选。

10 信

致:《纽约时报》纽约时报谜题编辑:2314151 团队

日期:2023 年 2 月 20 日

主题:我们团队的结果亲爱的纽约时报谜题编辑:

通过建立几个模型，我们已经完成了对 MCM 基于玩家参与你们网站每天提供的世界谜题的统计数据的分析。我们很荣幸地向您展示我们分析的结果。

在构建模型之前，我们对您提供给我们的数据集进行了澄清和规范化，并识别了单词属性，如重复字母的数量、元音字母的数量、辅音字母的数量、共性和频率。接下来，我将为大家描述一下建模求解过程:

首先，我们考虑了趋势、季节性和假期的影响，并构建了一个基于先知的时间序列预测模型，以解释报告数量的每日变化，并预测 2023 年 3 月 1 日报告结果数量的间隔:[10355,18742]。我们发现，从 2022 年 1 月到 2 月初，报告的结果数量急剧增加，就报告数量的变化而言，报告数量往往在一周内的周三最高，周末最低。

在探索单词属性对报告的影响时，我们首先使用了相关分析算法来求解每个属性，然而，我们发现它们的 Pearson 相关性系数都小于或接近 0.1，并且不相关。为了克服这种关联，我们使用了一种高阶偏相关分析算法，在控制单词属性之间相互作用的同时，对单词的每个属性进行相关性分析，最终得到了我们想要的结果:元音字母的数量、非重复的数量、单词的共性呈负相关，辅音字母的数量和非重复的数量呈正相关。

接下来，我们开发了一个优化的多目标回归预测框架，以探索单词属性对报告结果分布的影响。通过将数据输入到框架中，选择了最适合该数据集的预测模型:Lasso 回归预测模型，该模型对测试集的表现非常好，均方根误差为 0.8。因此，我们建立 Lasso 回归预测模型对报道结果进行预测，首先对“EERIE”进行处理，得到其尝试次数分布结果为(0,4,17,34,30,13,2)。

在此之后，我们提出了一种排序重要性分析算法来评价已建立的 but 这一属性，最终发现辅音字母数、元音字母数和频次这三个属性对报道结果分布的影响较大，影响因子分别为 4.226、3.993 和 1.253。

然后，我们使用建立的 Lasso 回归预测模型对词典数据库中所有 5 字母单词的报告结果进行分布，然后使用 K-Means 算法将上述结果分为难度高(≥4.37)、中(4.13-4.37)、低(< 4.13)三类，发现不同分类中重复数(Number of duplicate)、重复最大值(Maximum of repeats)、患病率(患病率)和频率(Frequency)存在显著差异，并对各属性进行区间划分。

然后我们将“EERIE”引入区间划分模型，由于其所有属性都符合难度区间值，因此其被猜测的程度是困难的，并且其平均尝试次数与使用 Lasso 回归模型预测的难度区间划分相匹配。因此，猜测“EERIE”的程度是困难的。同时，我们通过对不同难度词的属性区间进行匹配，得到了我们模型的准确率为 91.36%，可以看出我们的模型是鲁棒的。

最后，我们对模型进行了灵敏度分析，如结果所示，通过改变测试集的比例，最大变化率为10%，而模型的 RMSE 均小于 1.0，这证明我们的模型对该参数在自身 0.5 倍范围内的变化不敏感，我们的模型是稳健可靠的。此外，对数据集的研究揭示了《world》人气下降和难度模式挑战比例上升的现象，并为恢复游戏人气提供了两条建议:

引入在线多人挑战模式，可以邀请好友一起挑战，从而吸引更多人参与。

引入 Kids Mode，家长可以让孩子用这款软件锻炼英语技能，边玩边学。

以上就是我们研究的总结。衷心希望能给您提供有用的信息，期待您的回复。谢谢你！

你诚挚的，2314151 团队

References

[1] Benton J. Anderson and Jesse G. Meyer. Finding the optimal human strategy for wordle using maximum correct letter probabilities and reinforcement learning, 2022.

[2] Tarun Kumar. Solution of linear and non linear regression problem by k nearest neighbour approach: By using three sigma rule. In 2015 IEEE International Conference on Computational Intelligence Communication Technology, pages 197–201, 2015.

[3] Michal S Gal and Daniel L Rubinfeld. Data standardization. NYUL Rev., 94:737, 2019.

[4] Sean J Taylor and Benjamin Letham. Forecasting at scale. The American Statistician, 72(1):37–45, 2018.

[5] Qibin Zhao, Cesar F. Caiafa, Danilo P. Mandic, Zenas C. Chao, Yasuo Nagasaka, Naotaka Fujii, Liqing Zhang, and Andrzej Cichocki. Higher order partial least squares (hopls): A generalized multilinear regression method. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(7):1660–1673, 2013.

[6] Ertunga C ¨Ozelkan and Lucien Duckstein. Multi-objective fuzzy regression: a general framework. Computers & Operations Research, 27(7-8):635–652, 2000.

[7] Mark Harman. Making the case for morto: Multi objective regression test optimization. In 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops, pages 111–114, 2011.

[8] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. Pattern recognition, 36(2):451–461, 2003.

[9] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. Journal of the royal statistical society. series c (applied statistics), 28(1):100–108, 1979.

[10] Greg Hamerly and Charles Elkan. Learning the k in k-means. Advances in neural information processing systems, 16, 2003.