Problem Chosen	2023	Team Control Number
C	MCM/ICM	2300348
	Summary Sheet	

重振经典游戏:通过数据分析揭示世界的秘密

摘要：在数字时代，语言通常通过缩写、表情符号和语音信息来传达。然而，由《纽约时报》提供的世界游戏提供了一个回归语言基础的机会。因此，我们对世界得出的结果进行了数据分析。

首先，我们建立 GRU 预测模型，预测 2023 年 3 月 1 日的报告结果数量。该模型采用有效的门控循环单元(GRU)算法。因此，训练集对测试所做的预测，相对错误率为 2.1569%，相对 RESE 为 6.4957%，说明模型预测的准确性较好。2023 年 3 月 1 日报告结果数的预测区间为 20367±2.01569%。

其次，我们对分数百分比定义的单词属性和分数进行了数据分析。然后，我们定义了单词的四个属性:词频、字母频率之和、字母重复模式(2/3 或无)和主要词性。对于前两个属性，我们使用变量“score”进行回归分析。fword and 得分之间的 Pearson 相关系数为-0.3165,fletter and 得分之间的 Pearson 相关系数为-0.4005。Rep 和 pos 可以用来对单词进行分类。箱形图结果显示，rep 的箱形图的 Median difference 为 0.13004，而 pos 仅为 0.05973。因此，我们认为 f、f 和 rep 可以影响得分的百分比 word，letter 而 pos 则不能。

第三，建立 GSRF 预测模型，预测 2023 年 3 月 1 日 EERIE 1 到 X 的百分比。网格搜索随机森林(Grid-Search Random Forest, GSRF)算法是一种改进的随机森林算法，它利用了超参数的最佳组合。我们选择了 fword、fletter 和 rep 这三个参数作为模型的输入。模型的训练结果显示 MSE 为 20.70641,MAE 为 3.24388，表明具有良好的预测性能。(表 10)EERIE 的预测结果为(1,7,23,30,23,13,3)。此外，我们分别对 fletter word and f 加入高斯噪声进行了灵敏度分析，结果表明该模型灵敏度较低，因此具有较高的稳定性。

第四，建立了基于 K-Means++的难度分类模型。我们首先定义了每个单词的难度日期δ。根据预测分布，EERIE 的难度系数为 0.35916。然后，我们使用 K-Means++，分析每个单词的δ，得到五个难度等级(表 11)。EERIE 被划分为第三级。最后，我们将模型的分类与采样词子集的手动难度评级进行了比较，达到了 93.33%的匹配率，证实了模型的准确性。

最后，我们探索了另外两个数据特征。之后，一封由我们的稳定模型支持的信被写进了《纽约时报》的拼图编辑。

关键词:GRU;回归分析;箱形图分析,K-Means++

目录

1 介绍

1.1 问题背景

在这个数字时代，我们已经习惯了使用缩写、表情符号和语音信息进行交流。然而，有时候这些交流方式会剥夺语言本身的美和深度。但作为一个猜字游戏，世界谜语让我们以一种全新的方式回归语言的本质。通过感受每个字母和单词的节奏和意义，我们可以更深入地了解语言的魔力，欣赏其内在的魅力。

《世界谜题》是《纽约时报》目前提供的一款流行的每日谜题，它要求玩家在 6 次或更少的时间内猜出一个秘密的 5 个字母的单词。在每一轮开始时，系统随机选择一个单词，玩家必须在分配的猜测次数内运用推理和逻辑猜测出答案。每猜一次，系统就会指出这个单词中出现了哪些字母，以及它们的位置是否正确。更重要的是，玩家可以在“困难模式”中进行游戏，该模式要求玩家一旦正确猜测出答案中的一个字母，他们必须在随后的所有猜测中继续使用该字母。

许多用户(尽管不是全部)在推特上分享他们的分数。Benjamin Leis 开发的推特机器人“世界统计”(world Stats)可以用来跟踪和分析世界的每日得分报告。通过跟踪和分析来自世界统计的每日数据报告，我们可以提高我们猜词的能力，更好地理解英语的模式和用法。

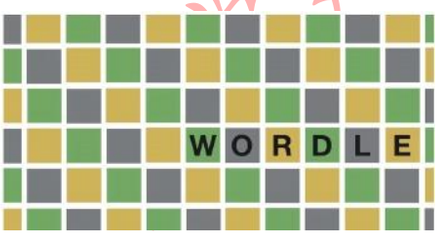


图 1:世界地图[2]

1.2 问题重述

我们需要分析纽约时报提供的关于世界的数据，并回答以下问题:

1.开发一个预测模型来预测未来一天的报告结果数量，并提供预测间隔。

2.分析单词属性是否对在困难模式中打出的分数百分比有影响，以及它们是如何影响的。

3.构建另一个预测模型，预测(1,2,3,4,5,6,X)的百分比，并以 2023 年 3 月 1 日的 EERIE 为例进行具体预测。还需要进行模型评价。

4.建立分类模型，对单词的难易程度进行分类，得到单词“EERIE”的难易程度。还需要进行模型评价。

5.探索数据中其他可能的相互作用，看看是否能发现有趣的特征。

1.3 数据清洗

观察所提供的数据，发现一些问题和相应的处理如下。

参考了 Wordle Status，我们发现附件中有一些错误的数据。其中包括不正确的单词长度，拼写错误，以及报告结果数量中的不正确值等。例如，314 中的单词被错误地写成了“tash”。此外，529 的报告

结果数被错误地记录为 2569。为了解决这些问题，我们进行了数据清理，以尽量减少错误，并提高数据的质量和准确性。以下是我们数据清洗过程的结果。

Original data	marxh(473)	tash(314)	clen(525)	rprobe(545)	2569(529)
Data after cleaning	marsh	trash	clean	probe	25569

表 1:数据清洗

在我们的数据检查过程中，我们发现了由于统计错误，某些日子的尝试比例总和不等于 100%的实例。为了解决这个问题，我们重新计算了每一天的比例，使总和正好是 100%。通过这样做，我们的目标是减少与不同日子的相同 PX 变量相关的错误，并提高我们模型的准确性。我们的处理结果是 $\sum_{i=1}^{X} p_i = 100\%.$

1.4 我们的工作

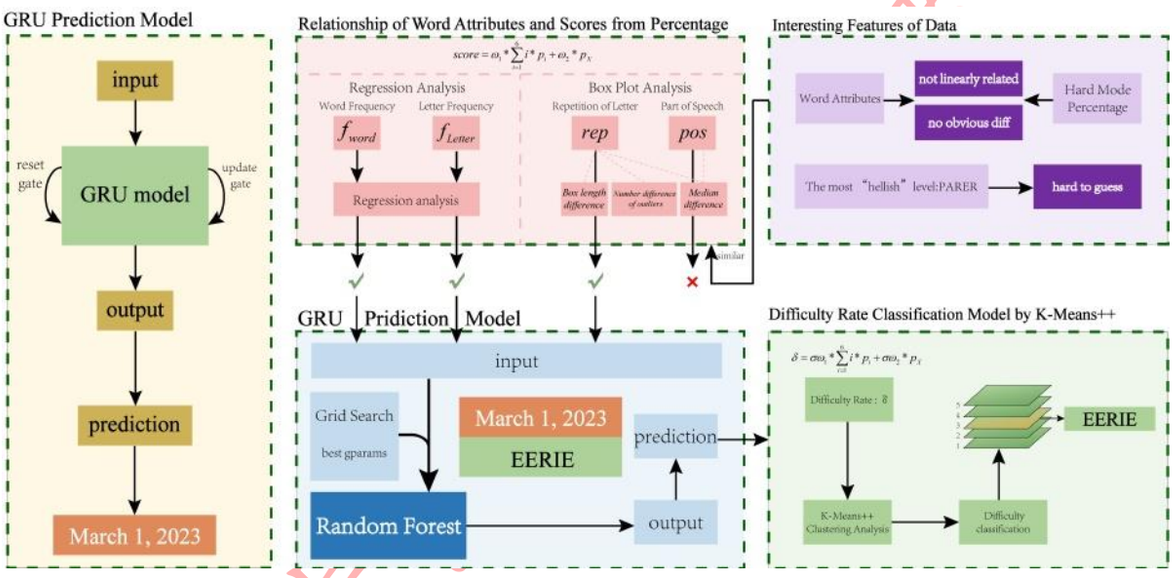


图 2:我们的工作

2 假设

为了简化我们的建模，我们做了以下假设:

假设 1 报告的在困难模式下的得分百分比可以用(1,2,3,4,5,6,X)的百分比来表示。因为世界排名之前在世界排名 207 中编译了整体模式和困难模式的值(1,2,3,4,5,6,X)，我们发现这两组值仅相差 1%[1]。这对我们后续的分析是有益的。

假设 2 无论玩家尝试了多少次，他们在 Twitter 上分享世界排名结果的可能性都是相同的。这个假设可以导致更准确的数据分析。

假设 3 每个玩家在玩《世界排名》之前都不知道答案，也不会在游戏过程中作弊。这保证了分数的百分比既客观又准确。

3 符号

Symbol	Decription
$score$	the combination of the percentage of scores
f_word	the word frequency
f_letter	the letter frequency
rep	the repetition of word
pos	the part of speech of word
σ	the difficulty rate
MSE	the Mean Squared Error
MAE	the Mean Absolute Error
d	the Euclidean distance
SSE	the Sum of Squared Errors

表 2:本文使用的关键符号

4 GRU 预测模型

报告结果的数量每天都在变化，分析这些数据的变化可以在一定程度上反映活跃世界排名的趋势。通过研究历史数据和趋势，甚至有可能对未来报告的结果做出有根据的预测。所以在这方面节中，我们应用门控循环单元(GRU)算法对提供的报告结果数量进行机器学习，并最终对 2023 年 3 月 1 日的报告结果数量进行预测。

4.1 GRU 算法描述

GRU(门控循环单元)是一种常用于时间序列分析的循环神经网络(RNN)。它具有与 LSTM(长短期记忆)架构相似的属性，但通常计算速度更快。

GRU 架构背后的主要思想是有两个门，一个重置门和一个更新门，它们控制通过网络的信息流。重置门决定有多少之前的隐藏状态应该被遗忘，而更新门决定有多少新的输入应该被添加到当前的隐藏状态中。

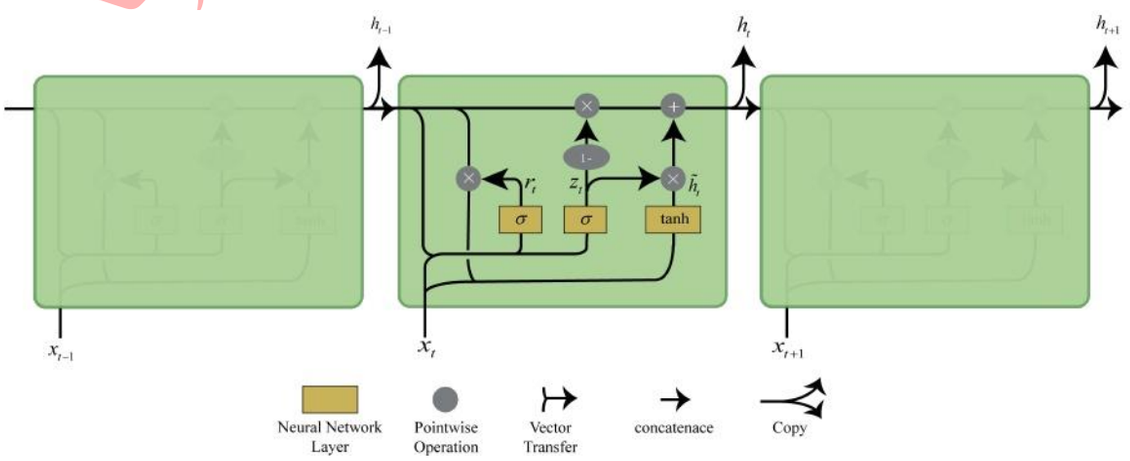


图 3:GRU 算法流程概述

GRU 的更新方程如下:

$$
\begin{aligned}
r_t &= \sigma\left(W_{ir}x_t + b_{ir} + W_{hr}h_{t-1} + b_{hr}\right) \\
z_t &= \sigma\left(W_{iz}x_t + b_{iz} + W_{hz}h_{t-1} + b_{hz}\right) \\
\tilde{h}_t &= \tanh\left(W_{in}x_t + b_{in} + r_t * \left(W_{hn}h_{t-1} + b_{hn}\right)\right) \\
h_t &= (1 - z_t) * \tilde{h}_t + z_t * h_{(t-1)}
\end{aligned} \tag{1}
$$

变量表：

Symbol	Decription
z_t	the update gate
r_t	the reset gate
h_t	the hidden state at time t
Symbol	Decription
h_{t-1}	the hidden state at time $t-1$ or at time 0
$\tilde{h}_t$	the new candidate hidden state
σ	the sigmoid function
$*$	the Hadamard product
x_t	the input
$W_{ir}, W_{hr}, W_{iz}, W_{hz}, W_{in}, W_{hn}$	the parameters that need to be trained
$b_{ir}, b_{hr}, b_{iz}, b_{hz}, b_{in}, b_{hn}$	the parameters that need to be trained

表 3:Equation1 中使用的符号

在这种方法中，时间序列的历史数据被输入到 GRU 模型中，以学习序列中的模式，然后可以用来预测未来的数据点。

4.2 2023 年 3 月 1 日的预测

在 Python 扩展库的支持下，我们选择使用 PyTorch 提供的 GRU 模型。PyTorch 是一个基于 Python 的机器学习库，其鲜明的特点是动态计算图，不同于静态计算图。动态计算图可以在运行时改变，这意味着模型可以根据我们的需要进行修改。这在处理变长序列数据时非常有用，非常适合预测我们需要预测的报告结果的数量。在 PyTorch 中，我们可以使用 torch.nn。GRU[3]类可以方便地构建和训练 GRU 模型，并使用该模型进行预测。

我们使用 2022 年 1 月 7 日至 2022 年 12 月 31 日每日“报告结果数”时间序列数据的 80%作为我们的 GRU 模型的训练集，其余 20%作为测试集。预测结果在测试集上的可视化如图 4 所示:

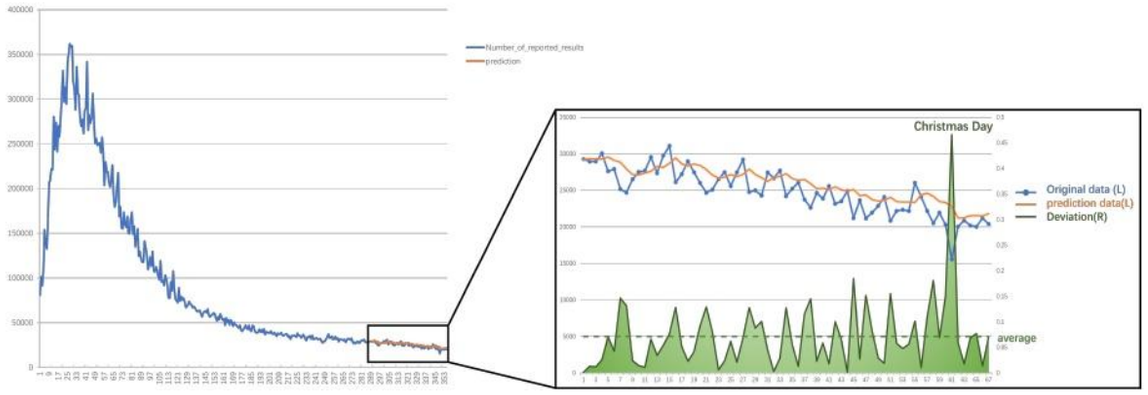


图 4:测试集上的预测结果。“偏差”表示预测数据与原始数据之间的相对错误率。

从图 3 可以看出，平均偏差在 0.06 左右，说明预测数据与原始数据的偏差仅为 6%左右。同时，还观察到一个有趣的现象:2022 年 12 月 25 日圣诞节的偏差接近 50%，这是由于原始数据中这一天报告结果的数量急剧下降。这很容易让人联想到这一天是一个重大节日。这样的结果在时间序列分析中属于离群值，有足够的理由去除该值，计算剩余数据的均值和均方根误差(RMSE)。RMSE 的计算方法如下:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_t - \hat{x}_t)^2}{n}} \tag{2}$$

如下表 2 所示，预测数据与原始数据的相对错误率为 2.1569%，说明两者值偏差较小。同时，相对 RMSE 是用 RMSE 除以原始数据计算得到的，在统计学中，当该值小于 10%时，认为误差较小。因此，可以得出结论，GRU 模型对报告结果的数量具有良好的预测性能，我们选择相对错误率 2.1569%作为模型预测的误差区间。

$\bar{x}_t$	$\bar{\hat{x}}_t$	Relative error	Relative error rate	RMSE	Relative RESE
24956.1667	25494.4569	538.29019	2.1569%	1621.0881	6.4957%

表 4 预测数据与原始数据的统计分析。

通过利用预训练的 GRU 模型，我们可以对接下来的 60 天进行预测，并获得 2023 年 3 月 1 日报告结果数量的预测区间:

$$x_{March1,2023} = 20367 \pm 2.01569\%$$

5 Word Attributes 和 Scores from Percentage 的关系

在本节中，我们对单词的四个属性和分数进行了数据分析，分数来源于分数的百分比。通过回归分析，我们发现单词的词频和字母频率与分数呈线性相关。通过箱线图分析，我们发现不同字母重复模式的单词得分存在一定差异，而不同词性的单词得分则没有显著差异。结果概述如图 5 所示

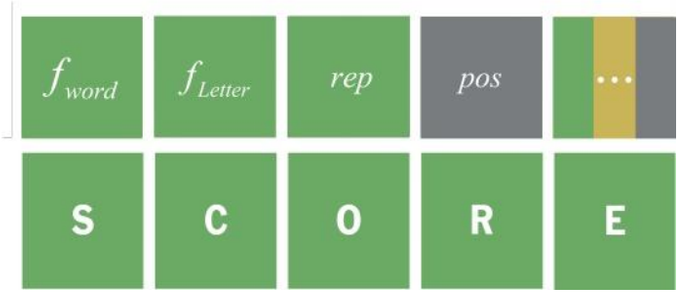


图 5:第 4 部分的结果概述

5.1 分数按百分比定义

每日尝试次数的百分比即为当天得分的百分比。但是，在调查词属性与分数百分比的关系时，分析一个词属性与由 7 个数字组成的分数百分比分布的关系是困难的，也不利于分析。因此，我们将这 7 个百分比值处理成一个数字，将其定义为分数。

分数由两部分组成。一部分是 1 到 6 的 tris 的加权平均值，另一部分是“X”的百分比。之所以考虑“X”的百分比，是因为 Wordle 每天只允许尝试一次，所以“X”的百分比实际上代表了当天猜测的失败率，这是不可忽视的。但由于“X”没有具体的尝试次数，无法参与加权平均的计算，所以我们将分数分为两部分，并使用熵权法对其进行权重分配。分数的定义如下:

$$score = \omega_1 * \sum_{i=1}^{6} i * p_i + \omega_2 * p_X \tag{3}$$

地点:

•pi 表示 i 次尝试(try)的百分比，i∈{1,2,3,4,5,6,X}

•ω1 和ω2 分别表示分数的两个部分的权重。我们用熵加权法将ω1设为0.5，ω2设为0.5

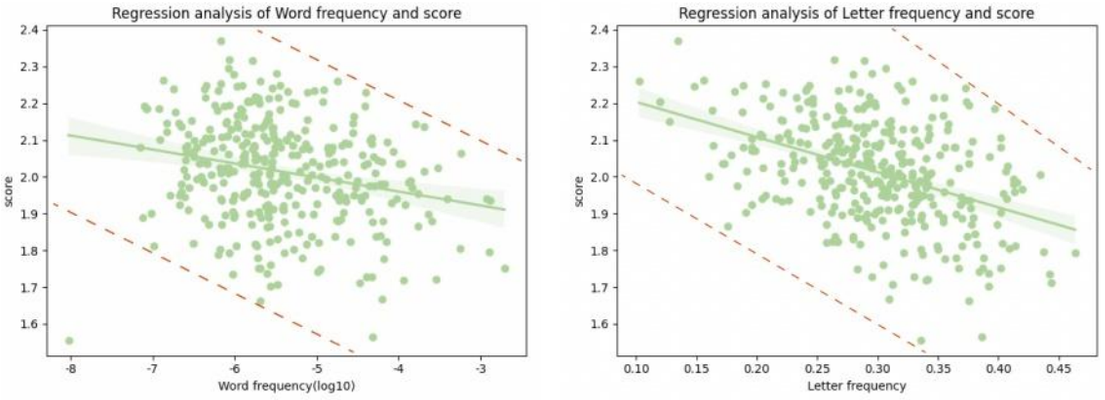
5.2 回归分析

5.2.1 f_{word}:词频

在尝试世界大战时，人们更容易回忆起日常语言中更常用的单词，比如“学习”和“训练”。因此，当解词的使用频率较高时，很可能尝试的百分比分布会向较少的尝试倾斜，导致得分值下降。为了解决这个问题，我们首先使用

网站[4]和 Python 的组合，以获得每个单词的可靠使用频率 word 。然后，我们对 fword and 得分进行了回归分析。

词频和得分的回归分析结果如图 6(a)和表 3 所示。Pearson 相关系数为-0.3165,Spearman 相关系数为-0.2956，说明词频与得分之间存在一定的线性相关性。另外，通过观察图 6(a)的散点图可以看出，在图像的左下角和右上角没有数据分布。这表明，单词频率高但难以猜测从而导致得分低的情况，以及单词频率低但容易猜测从而导致得分高的情况，都不太可能发生。因此，我们可以合理地得出词频与得分之间存在一定的线性相关性的结论。



(a) Word frequency (b) Letter frequency

图 6:词属性与得分之间的回归分析

Regression with *score*	Pearson	Spearman
Word frequency	-0.3165	-0.3256
Letter frequency	-0.4238	-0.4005

表 5:词属性与得分之间的回归分析

5.2.2 f_{Letter}:字母频率

每个字母都有自己的使用频率。当《世界大战》玩家使用字母频率更高的单词进行猜测时，他们可能更有可能猜中答案中的字母，从而获得更多线索，减少尝试次数。

字母频率，fLetter is 是将单词中每个字母的使用频率加起来得到的。字母频率的数据来自 Google Books Ngram Viewer[4]，其中包括 1500 年至 2008 年的书籍和其他出版物的语料库，可靠性很高。例如，字母“e”出现的频率为 11.1607%，而字母“z”出现的频率仅为 0.0772%。f_{Letter} 定义如下:

$$f_{Letter} = \sum_{1}^{5} f_{a,b,c...} \tag{4}$$

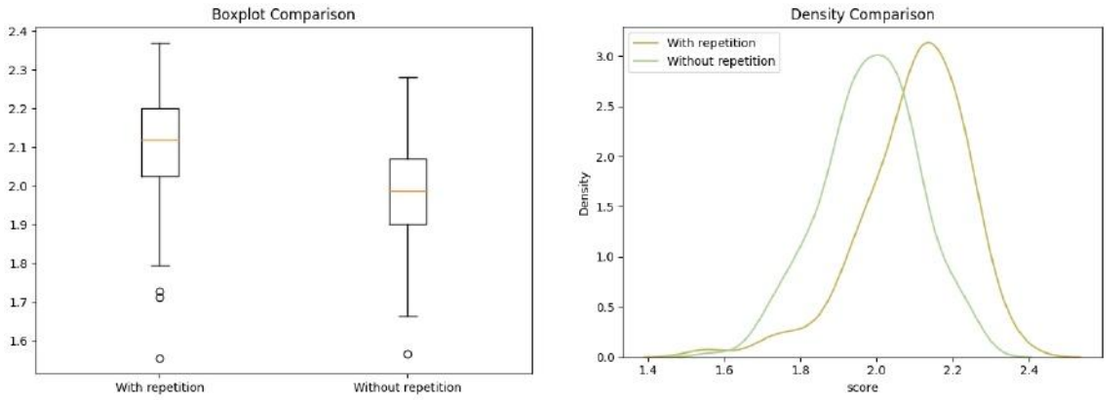
我们对 fLetter and 得分进行了回归分析，类似于词频。字母频率与得分的回归分析结果如图 6(b)和表 3 所示。Pearson 相关系数为-0.4238,Spearman 相关系数为-0.4005，说明字母频率与分数之间存在一定的线性相关性(甚至超过词频)。另外，通过观察图 6(b)的散点图可以看出，在图像的左下角和右上角没有数据分布。这也表明，单词频率高但难以猜测从而导致得分低的情况，以及单词频率低但容易猜测从而导致得分高的情况，都不太可能发生。因此，我们有理由得出这样的结论:字母频率与得分之间存在一定的线性相关性。

5.3 箱形图分析

5.3.1 重复:字母重复

当答案单词中有重复的字母时，在给定固定的单词长度的情况下，猜测和获得提示时，可能会有更大的机会击中字母，减少尝试次数。分析 2022 年 1 月 7 日到 2022 年 12 月 31 日的单词，我们发现在一

个单词中有字母重复两次或三次的情况(极少数情况)。因此，我们将单词分为两类:有重复字母和没有重复字母。我们想知道有和没有重复字母的单词在得分上是否有显著差异。为此，我们使用了箱形图分析:



(a) Boxplot of repetition (b) Density of repetition

图 7:字母重复与分数的箱线图分析

Median difference	Box length difference	Number difference of outliers
0.130037	0.00556	2

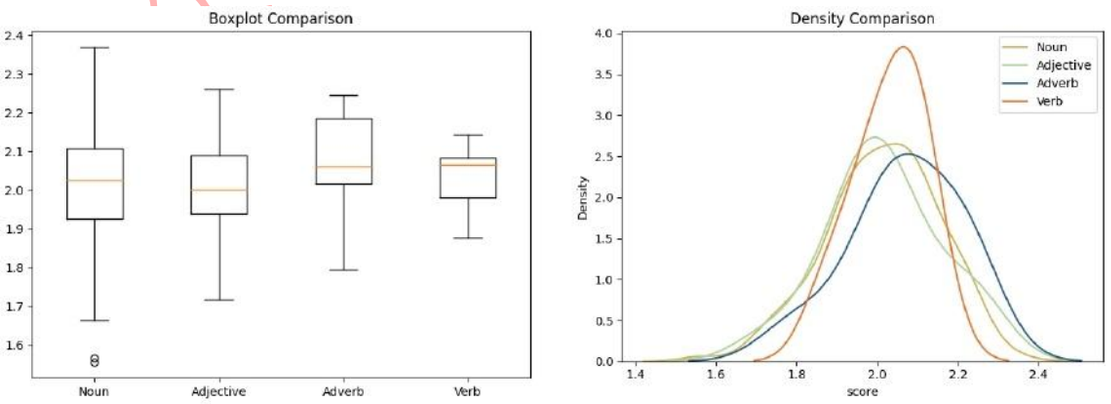
表 6:字母重复与分数的箱线图分析

从图 7(a)和表 6 可以看出，中位数差值为 0.130037，箱长差值为 0.00556。从图 7(b)中可以看出，两类数据的分布存在显著差异。因此，我们认为有和没有重复字母的单词在得分情况上存在一定的差异。

5.3.2 pos:Part of Speech

词性(POS)也是一个词的重要属性。我们感兴趣的是不同 POS 的词在得分上是否存在差异。

我们使用了流行的 Python 自然语言处理工具包 natural language toolkit[5]，其中包含各种文本处理和语言分析工具，包括词性标注(Part-of-Speech Tagging)。使用该工具，我们对 2022 年 1 月 7 日至 2022 年 12 月 31 日的单词进行了词性标注。单词主要分为四类:名词、形容词、副词和动词。随后，我们对这四类词的“得分”进行了箱形图分析:



(a) Boxplot of POS (b) Density of POS

图 8:POS 与评分的箱线图分析

Median difference1	Median difference2	Median difference3
0.024991	-0.05973	-0.05972

表 7:POS 与评分的箱线图分析

从图 8(a)和表 7 可以看出，名词和形容词的中位数差值为 0.024991，名词和副词的中位数差值为-0.05973，名词和动词的中位数差值为-0.05972。从图 8(b)中可以观察到，四类数据的集中趋势相当相似。因此，我们得出结论，在不同的 POS 中，单词的得分并没有表现出显著的差异。

6 GSRF 预测模型

在第 4 节中，我们将(1,2,3,4,5,6,X)的百分比处理为单个参数得分。虽然单个参数可以捕捉到这 7 个数字的一些整体特征，但这 7 个数字的具体含义不能单独表达，重要的细节和含义也可能丢失。

与第 4 节的时间序列预测不同，理论上报告结果的分布应该由当天答案词的属性决定，而不是由时间序列决定。因此，我们选择网格搜索随机森林(GSRF)算法来确定使用单词本身的三个属性来预测报告结果分布的最佳策略。利用在现有数据上训练的模型中获得的最佳策略，我们预测了 2023 年 3 月 1 日 EERIE 报告结果的分布，并取得了良好的预测性能。

6.1 GSRF 算法描述

GSRF 算法由随机森林和 网格搜索算法组成。

网格搜索算法是一种参数优化算法，通常用于微调机器学习模型中的超参数以优化其性能。它通过指定的参数网格进行迭代，并针对每个可能的参数组合对模型进行训练和评估，最终输出最佳参数集和相应的模型性能指标。

随机森林是一种机器学习算法，它是多个决策树的集合。随机森林的训练过程基于多棵决策树，算法在每棵决策树中随机选择一个特征子集进行训练。在预测过程中，随机森林通过平均或投票的方式将每个决策树的预测结果汇总，从而获得最终的预测结果。

与普通随机森林算法不同，GSRF 算法可以使用最佳的超参数组合来训练和预测随机森林模型，显著提高了模型的性能，避免了过拟合或过拟合等问题。GSRF 算法流程如图 9 所示。

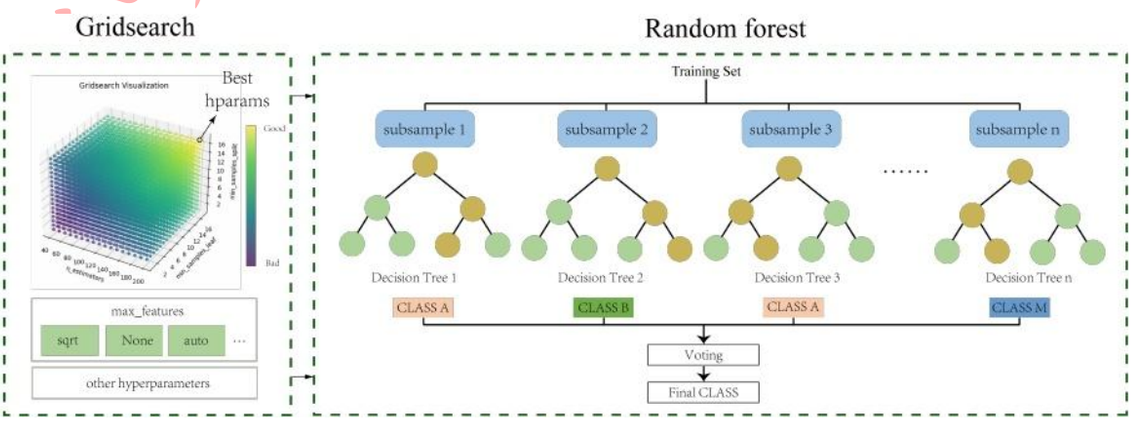


图 9:GSRF 算法流程

6.2 2023 年 3 月 1 日 EERIE 预测

在第 4 节中，我们对单词属性和分数百分比之间的关系进行了数据分析。我们发现，词频 fword、字母频率 fletter 和字母 rep 的重复对分数的百分比有一定的影响。因此，我们使用单词本身的这三个属性作为随机森林模型的输入参数来预测报告结果的分布:

$$(f_{word}, f_{letter}, rep) \xrightarrow{\text{GSRF}} (1, 2, 3, 4, 5, 6, X) \tag{5}$$

我们以想要预测的单词“eerie”为例，它的三个输入词属性如表 8 所示。需要注意的是，对于有重复字母的单词，我们将 rep 值设置为 1.5，对于没有重复字母的单词，我们将 rep 值设置为 1。这是合理的，因为 GSRF 在计算之前对输入数据进行了规范化。由于 rep 只有两个可能的值，所以它将被处理为 0 和 1。

$f_{word},$	$f_{letter},$	ref
0.00023%	0.418799	1.5

表 8:EERIE 的词属性

接下来，我们使用 2022 年 1 月 7 日至 2022 年 12 月 31 日期间每个单词的词频、字母频率和重复信息来训练 GSRF 模型，以了解其报告结果的分布。为了实现这一点，我们利用了 scikit-learn (sklearn)机器学习库的集成模块[6]中的 RandomForestRegressor 算法，以及 sklearn 中的 GridSearchCV 算法。Model_selection 模块[7]。使用训练好的 GSRF 模型，我们使用单词“eerie”的三个单词属性对报告结果的分布进行了预测。结果如表 9 和表 10 所示:

max depth,	max features,	min samples leaf	min samples split	n estimators
10	’sqrt’	2	10	200

表 9:GridSearchCV 的最佳超参数组合

1	2	3	4	5	6	X
1.07508%	6.51769%	23.30265%	30.29725%	23.13552%	13.27519%	2.40859%
MSE: 20.70641	MAE: 3.24388					

表 10:2023 年 3 月 1 日 EERIE 一词的预测

6.3 预测评价分析

当使用机器学习算法进行预测时，两个常用的评估指标是均方误差(MSE)和平均绝对误差(MAE)。MSE 是预测数据与原始数据差的平方的平均值，MAE 是预测数据与原始数据绝对差的平均值。它们的计算方法如下:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{6}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{7}$$

其中:

n 表示样本数量，yi 表示原始数据，y′表示 i 预测结果

数据。

从表 10 的数据可以看出，GSRF 预测模型对 2023 年 3 月 1 日“EERIE”报告结果的分布有很好的预测性能，MSE 为 20.70641,MAE 为 3.24388。这表明该模型能够根据单词的属性准确预测单词的报告结果，证明了 GSRF 算法在此任务中的有效性。

考虑到一个词的属性并不局限于我们应用的三个属性，其他未探索的属性也可能影响报告结果的分布。将这些附加属性作为模型的输入可能会潜在地提高其预测准确性。此外，单词出现的时间也可能对分布产生影响，就像“圣诞节”的特殊实例一样。

关于单词属性不完整的问题，我们的模型已经尝试全面考虑有利于预测的属性。

至于时间是否是一个因素，目前还不确定。

基于以上分析，我们认为我们的预测模型是全面而准确的。

7 基于 k - means++的难度率分类模型

在本节中，我们使用报告结果的分布和计算的“EERIE”和 2022 年 1 月 7 日至 2022 年 12 月 31 日的单词的δ来定义单词的难度率δ。然后，我们使用 K-Means++算法对单词的难易率进行聚类分析，得到了一个科学的难易率分类。EERIE 被列为三级。最后，我们随机抽取单词，手动标记其难度等级，并使用 K-Means++模型进行聚类分析。结果表明，我们的分类模型是相对准确的。

7.1 δ:难度率

单词的难度级别可以直接由(1、2、3、4、5、6、X)的百分比决定，可以观察到，当一个单词特别困难或难以猜测时，较大的尝试(例如 5 或 6)的百分比会增加。

与第 4 节的分数类似，难度率δ仍然由两部分组成。一部分是(1、2、3、4、5、6)百分比的加权平均值，另一部分是 X。然而，与分数中 X 分量以百分比计算不同的是，在计算δ时，X 与另一部分的量级差异相对较大。由于 X 反映的是某一天没能回答问题的人的百分比，所以在衡量一个单词的难度时，应该给予它更大的权重。为了平衡δ的两个参数，我们使用 Sigmoid 函数对这两个数据进行归一化，然后使用熵权法分配权重，得到难度率δ:

$$\delta = \sigma\omega_1 * \sum_{i=1}^{6} i * p_i + \sigma\omega_2 * p_X \tag{8}$$

地点:

pi 表示 i 次尝试(try)的百分比，i∈{1,2,3,4,5,6,X}

σ表示 sigmoid 函数

ω1 、2 ω分别表示分数两部分的权重。我们设ω为 1 0.5 和ω为 0.5 的熵权法。

7.2 K-Means++聚类分析

K-Means 聚类算法是一种常用的无监督机器学习算法，用于将数据分成几类。它预先指定了聚类的初始数量和初始

聚类中心，并根据样本之间距离的大小将样本集划分为不同的聚类。欧几里得距离用于度量数据对象之间的相似度，相似度与数据对象之间的距离成反比。相似度越大，距离越小。基于数据对象与聚类中心的相似性，不断更新聚类中心的位置，不断减小聚类的平方和误差(SSE)。当 SSE 不再变化或目标函数收敛时，聚类结束，得到最终结果。

数据对象与空间簇中心之间的欧几里得距离公式为:

$$d(X, C_i) = \sqrt{\sum_{j=1}^{m} (X_j - C_{ij})^2} \tag{9}$$

地点:

•X 表示数据对象

•ci 表示第 it 个簇中心

•m 表示数据对象的维数

•Xj and cij 分别表示 X 和 Ci 的第 j 个属性值

计算整个数据集的 SSE 的公式为:

$$SSE = \sum_{i=1}^{k} \sum_{X \in C_i} |d(X, C_i)|^2 \tag{10}$$

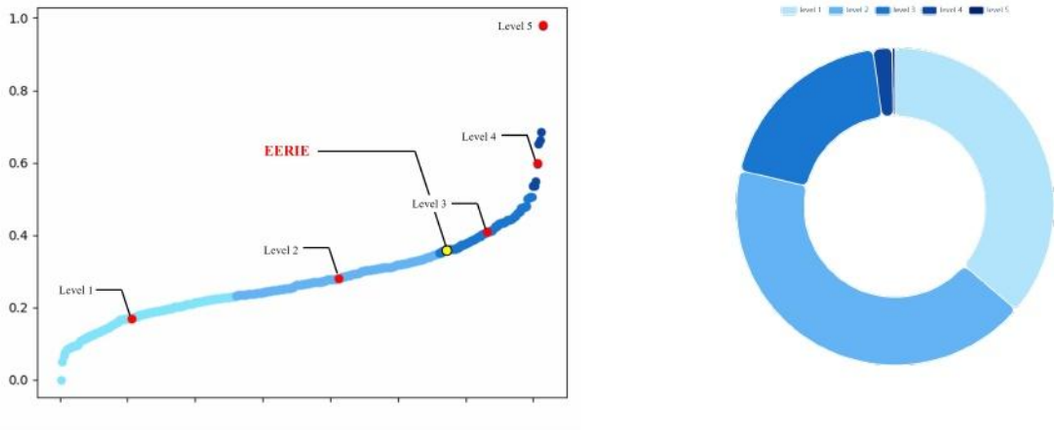
地点:

k 表示聚类数量

在传统的 K-Means 算法中，聚类中心的初始化通常是从 k 个样本点中随机选择的。然而，这种随机选择方法容易产生局部最优，导致聚类结果较差。K-Means++算法引入的概率选择过程可以使聚类中心更加分散，更容易找到全局最优解，从而提高聚类结果的质量。

7.3 EERIE 的难度分类

使用 Python 的 scikit-learn 库中的 K-Means 算法[8]，初始化参数集为“k- meme++”，我们可以对每个单词的计算难度率δ进行聚类分析。该分析将单词的难度系数分为 5 个级别，从 1 级到 5 级，级别越高表示单词的难度越大。计算出的单词“ERRIE”的难度系数δ为 0.35916，对应于难度等级中的第 3 级。结果如图 10 和表 11 所示:



(a) (b)

图 10:难度率分类

Level	Number	Percentage	Valuable Percentage
1	131	36.2%	36.2%
2	152	42.4%	42.4%
3	69	19.1%	19.1%
4	7	2%	2%
5	1	0.3%	0.3%
Sum	6.51769%	23.30265%	30.29725%

表 11:聚类分析结果

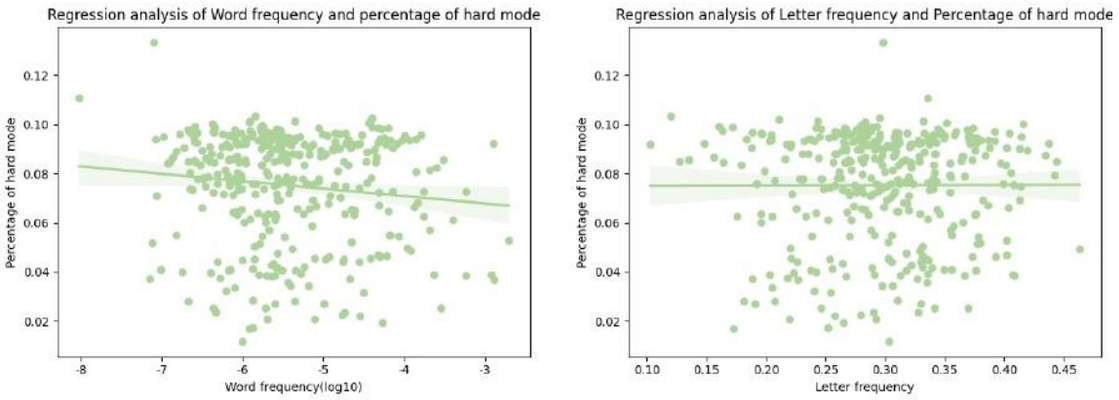
7.4 分类模型的准确性讨论

最后，我们随机抽取 30 个单词，手动标记它们的难度等级，并使用 k - means++模型进行聚类分析。将人工标注的样本词的难度等级与聚类分析得到的难度等级进行比较，30 个数据点中有 28 个数据点匹配，匹配率为 93.33%。这说明该模型得到的难度等级分类与我们对单词难度的主观判断是一致的。因此，我们认为我们的分类模型是相对准确的。单词样本的数据在附录中给出。

8 数据的有趣特性

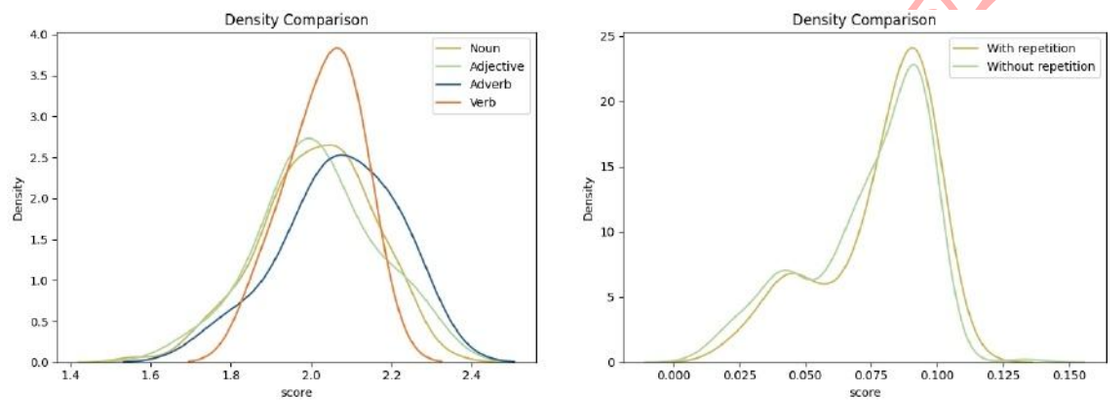
8.1 特性 1:单词属性与硬模式百分比的关系

我们感兴趣的是单词的属性是否与每天选择硬模式的用户比例有关。由于单词的属性在一定程度上与难度有关，如果某天单词难度过大，增加了用户的挫败感，用户第二天可能就不会选择硬模式。因此，我们对单词的四个属性和硬模式的每日百分比进行了数据分析，包括回归分析和箱线图分析。结果表明，硬模式的百分比与词频和字母频率没有线性关系。不同字母重复词和不同词性词的硬模式比例也无显著差异。结果如下所示:

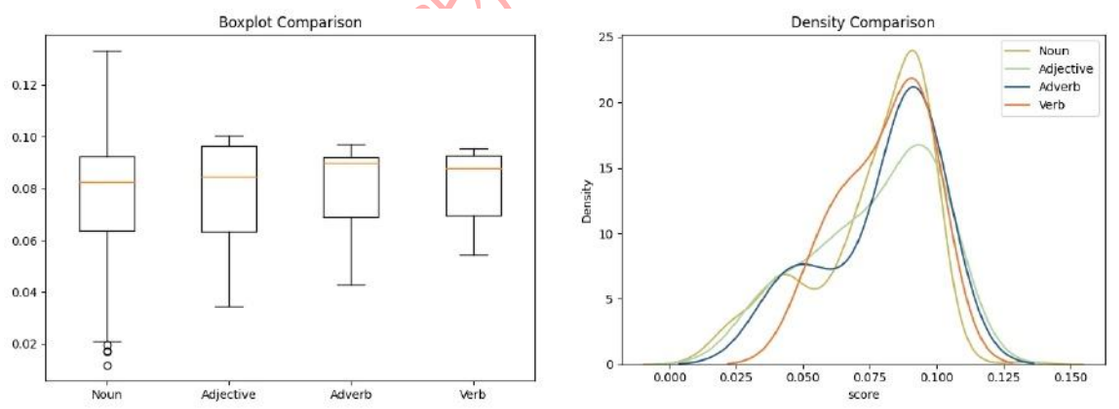


(a) Word frequency (b) Letter frequency

图 11:单词属性与硬模式 peercentage 之间的回归分析



(a) Boxplot of repetition (b) Density of repetition

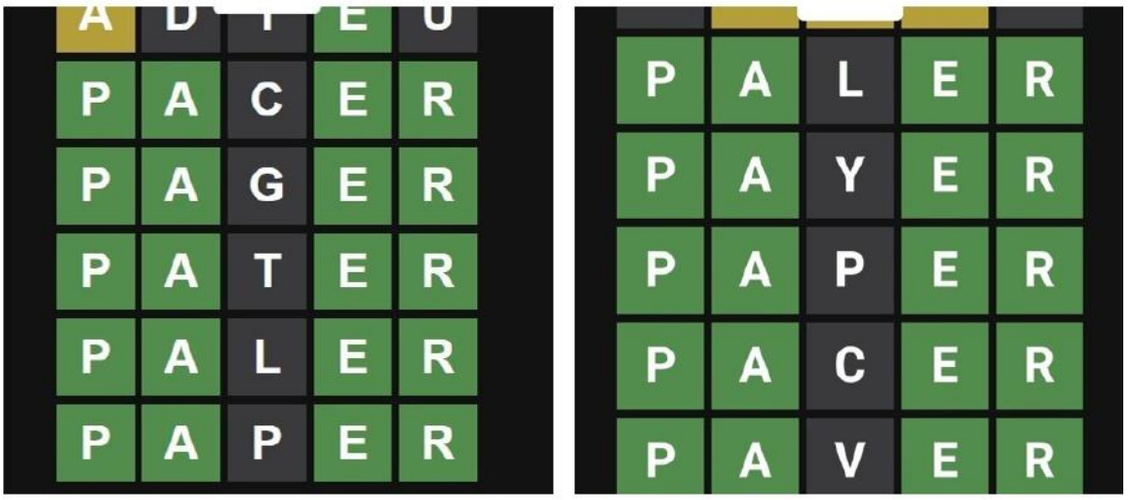


(c) Boxplot of POS (d) Density of POS

图 12:POS 与硬模式 peercentage 的箱线图分析

8.2 特征 2:为什么“PARER”具有最“地狱”的难度等级

在第 8 节中，我们发现“parer”这个词的难度日期高达 0.98，远远超过了得分 0.69 的第二名“木乃伊”。回顾原始数据，我们惊奇地发现，在“parer”当天，“X”百分比高达 48%，说明有 48%的玩家猜不出答案。在浏览 Wordle Stats 上的用户评论时，一些评论提供了一种可能的解释(图 13)，这表明当要猜的单词有很多相似的单词(即字母组成或位置相似)，并且这些相似的单词比解词的日常使用频率更高时，猜测正确的单词就变得困难了。



(a) User comment1[9] (b) User comment2[10]

图 13:用户评论

9 模型敏感性分析

我们对 GSRF 预测模型的输入参数进行敏感性分析，检验其在预测报告结果分布时对输入参数变化的敏感性。具体方法是在模型的输入参数中加入高斯噪声。但是，对于单词重复级别，只有两种类型的数据，在对输入数据进行归一化时，添加到 rep 中的噪声将被模型直接消除。因此，我们只对词频和字母频率进行了敏感性分析。高斯噪声定义如下:

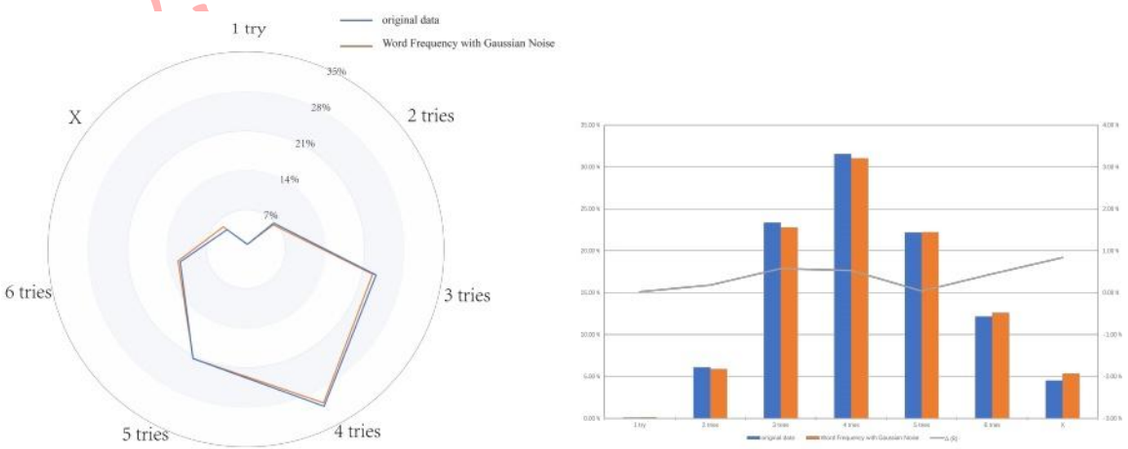
$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{11}$$

地点:

x 为随机信号的振幅，μ为均值，σ为标准差

9.1 word GSRF 预测模型的敏感性分析

对词频加 1 个高斯噪声后，结果如图 14 所示:



(a) (b)

图 14:词频的灵敏度

灵敏度分析的视觉结果表明，GSRF 预测模型对词频变化的敏感性较低，具有较高的稳定性。

9.2letter GSRF 预测模型的敏感性分析

在字母频率上加 1 个高斯噪声后，结果如图 15 所示:

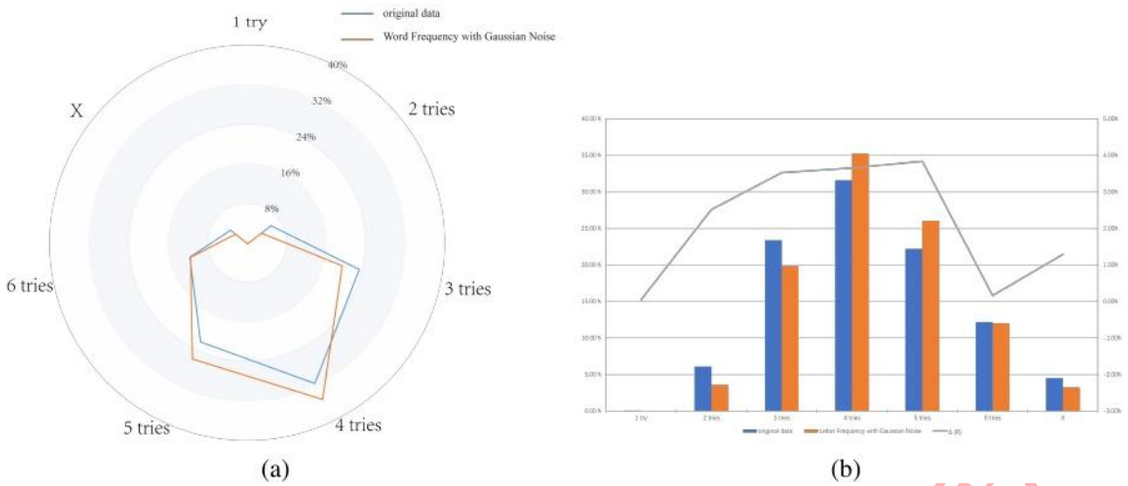


图 15:字母频率的灵敏度

灵敏度分析结果表明，GSRF 预测模型对字母频率变化的敏感性较低，具有较高的稳定性。

10 模型评估和进一步讨论

10.1 优势

1. GRU:第 4 节中的 GRU 算法在用于预测任务时具有几个优点。GRU 可以有效地处理具有可变长度输入的序列数据，使其非常适合于时间序列预测。与其他循环神经网络(如 LSTM)相比，GRU 通常需要更少的参数来训练，这使得它在计算上更高效，更容易训练。因此，我们的预测结果的相对错误率仅为 2.1569。

2. GSRF:与普通随机森林算法不同，第 6 节中的 GSRF 算法可以使用最佳的超参数组合来训练和预测随机森林模型，显著提高了模型的性能，避免了过拟合或过拟合的问题。

3. K-Means++:在传统的 K-Means 算法中，容易出现局部最优，导致聚类效果较差。K-Means++算法第 7 节引入的概率选择过程可以使聚类中心更加分散，更容易找到全局最优解，从而提高聚类结果的质量。

4. 我们在第 5 节的词属性分析和第 8 节的数据特征挖掘中的分析是比较全面和有建设性的。

10.2 薄弱之处及进一步讨论

1.单词有更多的属性，比如元音和辅音。

2.在预测报告结果的分布时，可能会考虑时间因素。

3.我们的模型使用了大量的机器学习算法。可以选择更高级的方法和数据训练技术。例如，K-Means 算法有各种变体，如 Mini-Batch K-Means 和 Genetic K-Means 算法。可以从每个变体中选择最合适的算法来分析当前的数据。

References

[1] https://twitter.com/WordleStats/status/1481687496241164291

[2] https://www.marca.com/tecnologia/2022/02/14/620a2ee522601da7288b4599.html

[3] https://pytorch.org/docs/stable/generated/torch.nn.GRU.html

[4] https://books.google.com/ngrams/

[5] http://www.nltk.org/

[6] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[7] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[8] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[9] https://twitter.com/WordleStats/status/1571182427007221764

[10] https://twitter.com/s_harmony_/status/1570624760530477057

11 信

亲爱的《纽约时报》拼图编辑。

我希望这条消息发现你一切都好。我写信是为了分享我们对世界数据的分析结果。

首先，我们建立 GRU 预测模型，预测 2023 年 3 月 1 日的报告结果数量。该模型采用有效的门控循环单元(GRU)算法。因此，训练集对测试所做的预测，相对错误率为 2.1569%，相对 RESE 为 6.4957%，说明模型预测的准确性较好。2023 年 3 月 1 日报告结果数的预测区间为 20367±2.01569

其次，我们对 Hard Mode 中打出的分数百分比所定义的单词和分数的属性进行了数据分析，探索它们之间的关系。然后，我们定义了单词的四个属性:词频、字母频率之和、字母重复模式(2/3 或无)和 pos:主要词性。对于 fordw 和 fetterl，我们使用变量“score”进行回归分析。ford 与 wscore 的 Pearson 相关系数为-0.3165,fetter 与 lscore 的 Pearson 相关系数为-0.4005。Rep 和 pos 可以用来对单词进行分类。箱形图结果显示，rep 的箱形图的 Median difference 为 0.13004，而 pos 仅为 0.05973。因此，我们认为 ford、fetterw 和 repl 可以影响得分的百分比，而 pos 则不能。

第三，建立 GSRF 预测模型，预测 2023 年 3 月 1 日 EERIE 1 到 X 的百分比。网格搜索随机森林(Grid-Search Random Forest, GSRF)算法是一种改进的随机森林算法，它利用了超参数的最佳组合。我们选择 fordw、fetterl 和 rep 这三个参数作为模型的输入。模型的训练结果显示 MSE 为 20.70641,MAE 为 3.24388，表明具有良好的预测性能。(表 10)EERIE 的预测结果为(1,7,23,30,23,13,3)。此外，我们分别 1 对 ford 和 wfetter 加入高斯噪声进行了灵敏度分析，结果表明该模型灵敏度较低，因此具有较高的稳定性。

第四，建立了基于 K-Means++的难度分类模型。我们首先定义了每个单词的难度日期δ。根据预测分布，EERIE 的难度系数为 0.35916。然后，我们使用改进的聚类分析算法 K-Means++对每个单词的δ进行分析，得到五个难度等级(表 11)。EERIE 被划分为第三级。最后，我们将模型的分类与采样词子集的手动难度评级进行了比较，达到了 93.33%的匹配率，证实了模型的准确性。

最后，我们发现我们分析的四个词属性并不影响选择硬模式的用户百分比。我们还调查了为什么“parer”是第 5 个难度级别中唯一的单词，其难度系数高达 0.98。事实证明，一个单词的难度可能与该单词与其他单词的相似度以及这些相似单词的使用频率有关。

如果你想了解更详细的信息，你可以阅读我们模型的全文。

Word	Manual	Model	Word	Manual	Model	Word	Manual	Model
atoll	3	3	foyer	5	5	twang	4	4
train	1	1	flock	4	4	bloke	4	4
madam	5	5	hairy	3	3	primo	5	5
peach	2	2	other	3	3	depth	2	2
admit	3	3	knoll	5	5	brine	3	3
trait	4	4	buggy	5	5	class	4	5
recap	3	3	favor	5	5	natal	5	5
carry	3	5	happy	1	1	atone	2	2
found	5	5	aphid	4	4	thyme	2	2
molar	4	4	bough	5	5	wacky	5	5

B 部分单词的难易率

Word	value	group	Word	value	group	Word	value	group
stein	0.139130856	1	treat	0.0894603	1	cloth	0.173900245	1
aloud	0.108252609	1	dream	0.097017869	1	poise	0.139130856	1
today	0.196054096	1	panic	0.151548495	1	glory	0.183128816	1
stair	0.064625023	1	doubt	0.12793931	1	caulk	0.310999245	2
grate	0.174870226	1	solar	0.173476571	1	infer	0.323441505	2
happy	0.203454774	1	choke	0.178493799	1	movie	0.317155672	2
metal	0.17185657	1	tepid	0.117904854	1	donor	0.23639953	2
tiara	0.168933189	1	begin	0.22372847	1	bluff	0.308941723	2
hoard	0.181350828	1	thyme	0.205706341	1	piney	0.349255486	2
avert	0.226157463	1	robin	0.208597166	1	beady	0.285658995	2
Word	**value**	**group**	**Word**	**value**	**group**	**Word**	**value**	**group**
cynic	0.246710978	2	showy	0.327805763	2	eject	0.360750519	3
lofty	0.313559582	2	cargo	0.289792582	2	gully	0.475589662	3
unfit	0.258769444	2	blown	0.253373134	2	sever	0.435166278	3
flock	0.263786672	2	glyph	0.233683305	2	vivid	0.407453933	3
carry	0.293076879	2	nasty	0.236784568	2	comma	0.373391992	3
condo	0.339162962	2	creak	0.303318231	2	wedge	0.376850239	3
sweet	0.304997854	2	shard	0.292839653	2	motto	0.355507495	3
soggy	0.308474029	2	elope	0.293112002	2	droll	0.362206003	3
flood	0.271312514	2	howdy	0.38417086	3	mummy	0.685643564	4
story	0.23657413	2	gamer	0.368070525	3	parer	0.978421619	5