Problem Chosen	2023	Team Control Number
C	MCM/ICM	2307946
	Summary Sheet	

世界背后的话语:使用机器学习和时间序列理论的益智游戏分析

摘要：《世界谜题》是目前《纽约时报》每天提供的一个很受欢迎的谜题。玩家尝试在 6 次或更少的时间内猜出一个 5 个字母的单词来解决这个谜题，每次猜出都会收到反馈。充分利用相关信息，可以有效帮助编辑提高操作性能。

首先，为了解释变化并预测未来值，引入了基于报道结果数量的时间序列模型。确定最优订单组后，利用 ARIMA(0,1,1)模型预测 2023 年 3 月 1 日报告结果数量的预测区间为[10139.23,30808.07](80%置信度)。为了研究词的属性是否会影响词的硬模率，引入了词的属性系统和 LightGBM 模型。结果表明，有一些滞后属性比滞后硬模百分比本身有一些但效果较小的影响。

其次，为了预测(1,2,3,4,5,6,X)的关联百分比，建立了基于 GBDT 和 MMoE 的两个模型。结果表明，MMoE 模型显著优于 GBDT 模型，MSE 为 145。然后，我们尝试使用数据增强和特征工程方法来改进模型。前者会导致大量的噪声，无法达到预期的效果，后者则略微提高了模型的性能。对单词 EERIE 的最终模型预测值为(0.649,7.579,26.298,32.614,20.930,9.63,2.298)。

第三，引入 K-means 模型，以尝试次数的分布作为难度特征，将样本聚为 4 组;为了确定单词的哪些特征与分类相关联，我们将分类作为输出特征，将单词的所有属性作为输入特征，建立 LightGBM 模型进行训练。测试集的准确率达到了 70%。对输出特征的重要性进行排序。最后，利用该模型对 EERIE 词的类别进行预测，预测结果为第二组。

最后，在 dataset 中发现了该数据集的一些有趣的特征。讨论了大频率词的特征、尝试数的分布形状和词特征的相关性。

此外，我们对模型的优缺点进行了评价并提出了一些建议，并对模型对委托率进行了敏感性分析，从而证明了模型的可靠性和稳定性。

关键词: Wordle;ARIMA;LightGBM;MMoE;数据增加;工程特点;K-means;敏感性分析

目录

1 介绍

1.1 问题背景

在 2022 年初，一个简单而新颖的游戏在推特上大受欢迎。这是一个由 Josh 沃德尔编写的网络文字游戏世界，由纽约时报公司出版。

这款游戏一开始并不为人所知，但在沃德尔创造性地添加了一个功能，允许玩家将结果复制到彩色方形表情符号的网格中进行分享后，它立即引起了公众的关注。截至 2022 年 1 月中旬，已有超过 200 万人参与其中，推特上发布了超过 120 万份世界大战结果。

在《世界大战》中，玩家必须在一天内 6 次猜出一个有 5 个英文字母的单词。每次尝试后，玩家可能会得到三种类型的反馈:如果字母在正确的位置，则显示绿色;黄色表示答案中包含字母，但字母在错误的位置;灰色表示答案中根本没有字母。游戏玩法与《Mastermind》类似，但《世界大战》会清楚地指出哪些字母猜对了。[1]

除此之外，《世界大战》还有另一种游戏模式。在上述规则的基础上，“困难模式”要求玩家一旦在单词中找到正确的字母，就必须在随后的猜测中使用这些字母。

实际上，在这个看似简单的游戏背后，有着深刻的数学机制。我们不禁想知道是什么机制影响了玩家做出正确猜测的效率，以及推特上不断变化的报告结果数量背后存在什么规律。玩家选择困难模式的依据是什么?

我们期望通过数学建模解决以上问题，有效预测游戏未来的运营，并为《纽约时报》的 Puzzle Editor 提供商业建议。

图 1:纽约时报世界

图 2:解决方案示例

1.2 问题重述

因为我们有一个数据集，包含日期、比赛号码、当天的单词、当天报告得分的人数、困难模式的玩家人数以及报告结果的分布。我们需要建立数学模型，为纽约时报公司解决以下问题:

问题 1:

1.开发一个模型来解释报告结果数字的变化，然后使用开发的模型对 2023 年第 1 场比赛的这个数字进行预测。

2.找出可能影响在困难模式中打出的分数百分比的给定单词的属性，并给出影响的内在机制。

问题 2:建立一个模型，预测某一天给定单词的报告结果的分布。然后讨论预测模型的不确定性和准确性。

问题 3:采用数学模型按难度对解词进行分类，识别与每个分类相关联的给定词的属性，并评估分类的准确性。

问题 4:讨论并发现数据集中的其他特征。

2 一般假设和模型概述

2.1 假设

为了简化问题，我们做了以下基本假设，每一个假设都是合理的。

1.Twitter 上报告的结果数量可以有效地代表当天的玩家总数，并且报告的困难模式分数百分比与所有玩家的分数百分比相同。

2.数据集中记录的报告结果分布是完全准确的。

3.1 次尝试、2 次尝试、…、X 的关联百分比之间存在相关性和差异性。

4.单词难度与猜测结果的平均尝试次数成正比。

2.2 模型概述

综上所述，整个建模过程可以显示如下:

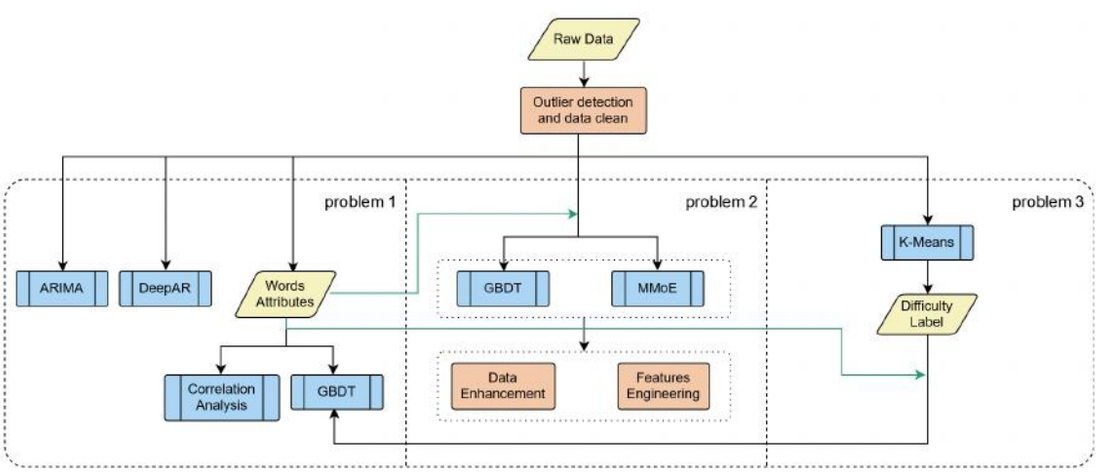

图 3:模型概述

3 模型准备

3.1 符号

本文使用的重要符号列于表 1，

Symbols	Definitions
y_t^T	The number of reported results at day t
$y_t^{T'}$	The first-order difference sequence of y_t^T
$y_t^{T''}$	The first-order difference sequence of y_t^T (after May 16, 2022)
y_t^H	Number of reported results in Hard Mode at day t
pct_t	The percentage of scores reported that were played in Hard Mode at day t
$distribute_i$	The percentage that guessed the word in i tries

表 1:标注

3.2 数据预处理

我们使用的数据包括题目 C data Wordle.xlsx 给出的数据文件

这个文件几乎提供了我们解决这个问题所需的所有信息。但在使用它之前，需要对数据进行预处理。

首先，我们需要排除数据集中的异常值，即去除在困难模式中打出的分数所占的百分比(以下简称困难模式百分比)与其他数据相差太远的点。将报告结果数、硬模式数、困难模式百分比与时间的关系用一条平线绘制散点图时，我们很容易发现异常值的存在，如下图中用红点表示。

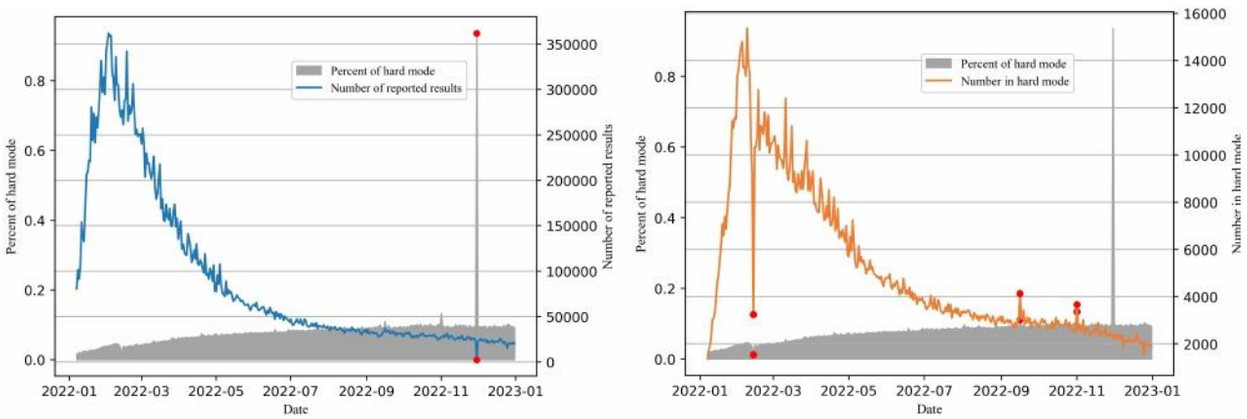

图 4:数据集中三个时间序列的散点图

采用滚动边界统计的方法，选取某日前 30 天和后 30 天的硬模式百分比作为样本组，按下式构建滚动区间:

$$Means \pm 3 \times (Standard\ Deviation) \tag{1}$$

如果当天的硬模式百分比超出此间隔，则为离群值。按照这种方法，我们对数据集进行了测试，得到了 4 个离群点。在剔除离群点后，我们使用线性插值方法对原始数据进行补全，得到一个更加平稳的数据集。

此外，我们还发现数据集中存在拼写错误。例如，一些给定的单词只包含 4 个字母，这与游戏中 5 个字母的规则相矛盾。我们删除了有拼错单词的样本数据，以保证数据的有效性。

所有待处理的数据样本和处理结果如下表所示。

表 2:数据预处理清单(Outliers and Misspelling)

Date	Error location	Adjustment	Adjusted number
2022/11/30	Number of reported results	interpolation	10.37%
2022/11/26	Word	Deletion	N/A
2022/11/01	Number in hard mode	interpolation	9.48%
2022/10/05	Word	Deletion	N/A
2022/09/16	Number in hard mode	interpolation	8.15%
2022/04/29	Word	Deletion	N/A
2022/02/13	Number in hard mode	interpolation	3.54%

4 模型一:时间序列预测模型

4.1 时间序列的概念

时间序列是按时间顺序排列的数字序列。最常见的是，时间序列是在连续等间隔的时间点上拍摄的序列。时间序列分析包括分析时间序列数据以提取有意义的统计数据和数据的其他特征的方法。时间序列预测是利用模型根据先前的观测值来预测未来的值。

4.2 时间序列的平稳性

如果我们要做一个时间序列预测，首先要保证它的平稳性。一个平稳的时间序列应该没有趋势性和季节性，也就是说，它的均值和方差是恒定的。

然而，从图 4 中，我们可以很容易地推导出，报告结果数量的时间序列具有明显的趋势，并不平稳。为了保证我们的判断，我们随后采用了 ADF (Augmented Dickey-Fuller)检验。这个检验的零假设是时间序列不是平稳的。[2]得到的显著性检验统计量为-1.9608，而 p 值为 0.5934，不能拒绝原假设。这证实了我们的判断。

鉴于数据的非平稳性，我们对数据进行了一阶差分处理。我们将报告结果数的时间序列设为 y_t^T ，并构造其一阶差分序列 $y_t^{T\prime} = y^T{}_t - y^T{}_{t-1}$ 。用一条光滑的线做了一个散点图，如下图所示:

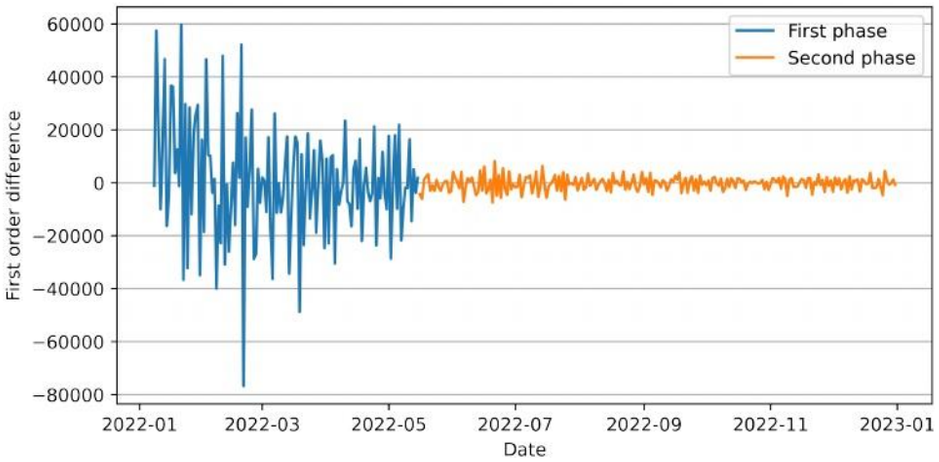

图 5:y_t^T After Difference Process 的时间序列从

图中可以看出，正在处理的报告结果数量的时间序列在 2022 年　5 月 16 日之前方差较大，在 2022 年 5 月 16 日之后方差较小，因此选择 5 月 16 日之后的数据作为新的时间序列来预测未来值。记录为 $y_t^{T"}$。

之后，我们再次对 $y_t^{T"}$进行 ADF 检验，这次 p 值为 0.01，拒绝原假设，表明一阶差分后的时间序列是平稳的。

4.3 模型建立

在保证时间序列的平稳性后，我们可以使用 ARIMA(自回归综合移动平均模型)模型进行时间序列预测。[3]模型的一般表达式为 ARIMA(p,d,q):

$$(1-\sum_{i=1}^{p}\alpha_i L^i)(1-L)^d y_t = \alpha_0 + (1+\sum_{i=1}^{q}\beta_i L^i)\epsilon_t \qquad (2)$$

其实质是将差分(d)、自回归模型(AR(p))和移动平均模型(MA(q))相结合。P 是自回归模型的阶数，d 是差的阶数，q 是移动平均模型的阶数。通过绘制时间序列 ytT ″的自相关(ACF)图和部分自相关(PACF)图，发现 ACF 是一阶截断的，PACF 是尾化的，因此我们可以选择 ARIMA(0,1,1)作为目标模型。

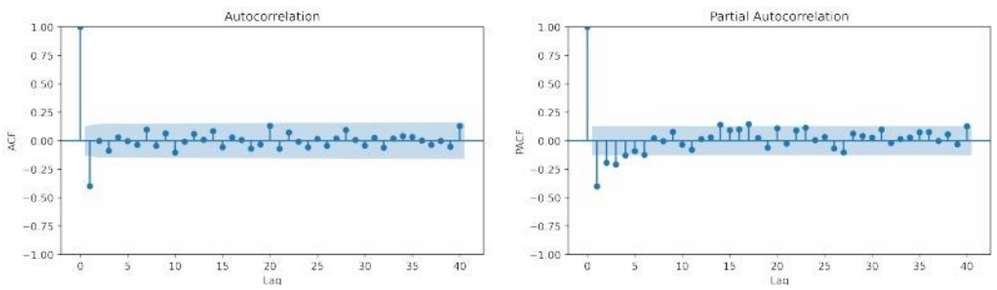

图 6:自相关(Autocorrelation)和偏自相关(Partial Autocorrelation)

同时，利用该方法也可以得到最佳的 ARIMA 参数。在 R 语言中的 arima 命令，产生 ARIMA(1,1,1)和 ARIMA(3,1,1)。

为了从所有获得的 ARIMA 参数中确定最佳参数，我们使用了信息准则(赤池信息准则)来辅助判断。三个可选模型的 AIC 值分别为 4200.897/4202.825/4205.466，其中 ARIMA(0,1,1)的 AIC 值最小，因此选择其作为最终模型。

为了确定 ARIMA(0,1,1)模型的有效性，我们进一步进行了白噪声残差检验。检验结果显示 LB 统计量的 p 值都在 0.05 的阈值以上，所以模型通过了检验，我们可以用 ARIMA(0,1,1)来解释报告结果数量的变化。

4.4 预测结果

利用 ARIMA(0,1,1)模型，代入数据，预测结果如下图所示:

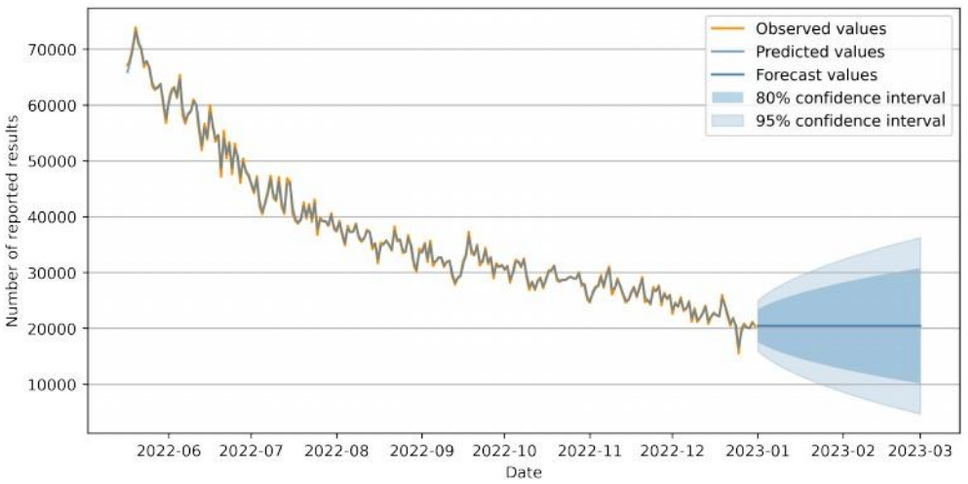

图 7:预测结果

对于 2023 年 3 月 31 日，报告结果数的置信水平为 80%的预测区间为[10139.23,30808.07]，置信水平为 95%的预测区间为[4668.516,36278.78]，预测期望值为 20473.65。

5 Word 的提取和分析属性

5.1 提取词的属性

像《世界大战》这样的字谜游戏的难度与给定单词的属性密切相关。鉴于《世界大战》游戏规则的特点，其难度主要取决于给定单词的可记忆性和玩家回忆起该单词的难度。许多教育工作者和语言学家对影响学生词汇记忆的因素进行了研究。例如，Laufer, B.(1990)[4]列举了许多影响词汇学习表现的单词属性。对于世界来说，为了得到答案，除了学习和记忆单词之外，还必须根据已知的信息来推断结果，这一难度也受到单词的许多属性的影响。

在选择词的属性时，参考 Jakub Jagoda & Tomasz Boi´nski(2018)[5]等已有的研究，我们可以得到一些影响测验难度的常见词属性，如词频、词性、元音数、重复字母数和词的情绪倾向。此外，psy- cholinguists 的研究也为我们提供了许多创新的指标，比如一个单词所拥有的正字法邻接数、主发音中的音节数等等。

5.2 字属性概述

我们的属性数据主要来源于 The English Lexicon Project[6]，这是一个由多所大学联合参与的语言学项目，旨在为 40,481 个单词和 40,481 个非单词提供标准化的行为和描述性数据集。数据可通过访问 elexicon.wustl.edu 获取。此外，一些 Python 包和算法也可以为我们提供关于词属性的数据。

我们使用的其他研究成果包括:x 语言库、Bysbaert 等人(2014)对具体等级的研究[7]、Hoffman 等人(2013)对语义多样性的研究[8]和 DeDeyne 等人(2018)对关联频率的研究[9]。

值得注意的是，不同时间段的词属性可能对游戏难度有不同的影响。比如，实际上玩家在开始游戏之前并不知道今天给定的单词到底是什么，所以我们可以很自然地猜测，今天单词的单词属性对今天游戏难度的影响是有限的。但与此同时，过去的单词属性可能会显著影响玩家所假定的游戏难度。如果过去几天的单词难以推断，那么当天的玩家可能不太倾向于选择 Hard Mode，甚至会因为缺乏信心而输掉游戏。对过去几天词属性的滞后效应的研究，将成为下面建模的重点之一。

下表列出了我们研究中提取的所有词语特征及其含义，以及数据来源:

表 3:单词属性概述

Attribute Notation	Meaning of the Attribute	Data Sources
word_freq	How often the word is used in everyday life	Database from Kaggle
num_vowel	Number of vowels in the word	Counted using python
num_repeat	Number of repeated letters	Counted using python
part of speech	Such as nouns, pronouns, etc.	Python package NLTK
sentiment	Sentimental attributes of words[1]	Package vaderSentiment
Ortho_N	The number of orthographic neighbors that a word has	English Lexicon Project
Phono_N	The number of phonological neighbors[2] that a word has	English Lexicon Project
OG_N	The number of phonographic neighbors[3] that a word has	English Lexicon Project
BG_Sum	The sum of the bigram count for a particular word	English Lexicon Project
NPhon	The number of phonemes in the main[4] pronunciation	English Lexicon Project
SUBTLCD	The SUBTLEX contextual diversity[5]	SUBTLEX
OLD	The mean of the closest 20 LD neighbors for the orthograph	English Lexicon Project
PLD	The mean of the closest 20 LD neighbors for the phonology	English Lexicon Project
Concreteness_Rating	The mean of the Concreteness Ratings	Bysbaert et al. (2013)
Semantic_Diversity	The Semantic Diversity	Hoffman et al.(2013)
Assoc_Freq_R123	Number of Times Word is one of first three associates	DeDeyne, et al.(2018)
NMorph	The number of Morphemes	English Lexicon Project

[1] Such as positive, negative, etc.

1 如正面、消极等

2 这个统计不包括同音异义词

3 这个统计数字不包括同音异义词

4 双元音/aI/， /aU/， /OI/，以及闪音/tS/和/dZ/，每一个都算作单个音素

5 包含单词的电影的%

6 模型二:用 Light- GBM 解释硬模式百分比

6.1 LightGBM 简介

LightGBM 是 Ke 于 2017 年提出的一种新型 GBDT(梯度提升决策树)算法(Ke 等，2017)[10]。GBDT 具有梯度增强和决策树的功能特点，具有训练效果好、不易过拟合的优点。其优点包括训练速度快、准确率高、内存占用少、

以及对并行计算的支持。它可以用来解决 GBDT 在海量数据处理中遇到的问题。

LightGBM 的特点之一是使用了基于直方图的决策树算法，该算法首先将连续特征值离散为 k 个值，然后生成宽度为 k 的直方图。在遍历样本时，将离散值作为索引。经过一次遍历后，直方图累积所需的统计量，然后通过直方图的离散值进行遍历，找到最优的分割点。

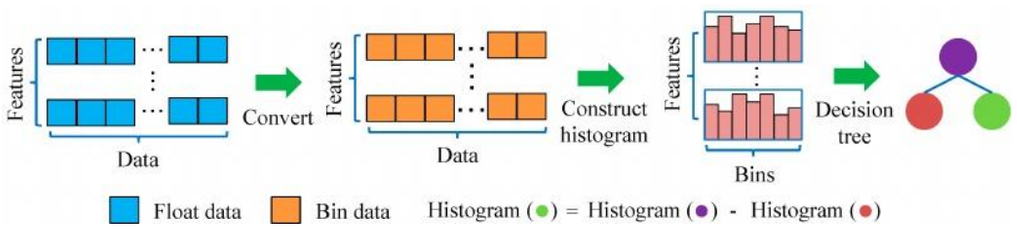

图 8:基于直方图的决策树算法

LightGBM 的另一个特点是采用更有效的叶片生长策略，即具有深度限制的叶片生长策略(Leaf-wise)。在分裂之前，该策略首先遍历树中所有的叶子，然后找到分裂增益最大的叶子再次分裂，并重复此操作。实验证明，在相同的分割次数下，Leaf-wise 可以获得更高的精度，并且在 Leaf-wise 中增加了防止过拟合的最大深度限制。

leaf-wise 的叶片生长策略如下图所示，其中白点和黑点分别代表了分裂增益最大和非最大的叶片:

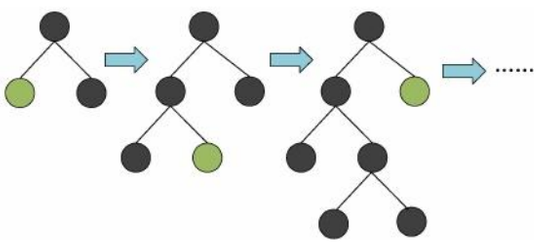

图 9:Leaf-Wise Tree Growth 示意图

6.2 数据描述与预处理

本模型的主要目的是识别词的属性是否会影响硬模百分比，并探讨影响的机制。因此，模型的标签是当期硬模式的百分比，用 pct_t 表示。

考虑到前一个词属性和标签本身对当期标签的滞后效应，我们的特征序列包括表 3 所涉及的所有属性指标 1 期(1 天)到 5 期(5 天)的滞后项，以及硬模式百分比的当期值和 1 期到 5 期的滞后项。如果将给定单词在周期 t 内的所有属性记为 X，则模型输入包含:

$$pct_{t-1}, pct_{t-2}, \cdots, pct_{t-5}; X_t, X_{t-1}, \cdots, X_{t-5} \tag{3}$$

为了保证模型的有效性和可靠性，需要对数据集进行预处理。一个非常重要的步骤是对每个属性的值进行规范化，并将数据映射到[0,1]。方法如下所示:

$$v_s = \frac{v - v_{min}}{v_{max} - v_{min}} \tag{4}$$

其中，v_s 为标准化值，v 为原值，v_max、v_min分别表示属性的最大值和最小值。归一化的数据统称为 $data_{input}$

6.3 模型结果与评价

我们利用上述数据，输入特征序列，以训练集:测试集= 4:1 的比例对模型进行训练。然后通过大量的历史数据关联计算，对不同的特征赋予不同的权重，并利用这些权重输出硬模式百分比的估计值(或预测值)。

为了验证 LightGBM 模型的可靠性，我们使用预测结果的 MSE、RMSE 和 SMAPE 作为模型精度的度量指标。参数优化后，训练集和验证集内各指标的计算结果如下:

表 4:模型精度测试结果

	MSE	RMSE	SMAPE
Training set	3.623×10^{-6}	0.0019	0.0253
validation set	3.382×10^{-5}	0.0058	0.0690

由于所有指标显然都很小，我们可以很容易地断言模型结果与真实数据非常吻合。这个结论也可以从下面预测值和观测值的散点图中得出。

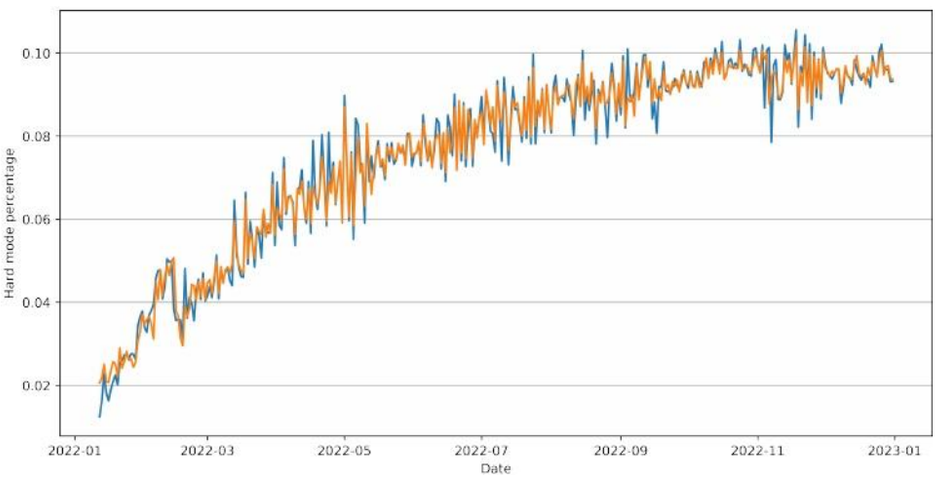

图 10:LightGBM 预测结果与观测值对比

由于模型的可靠性，我们可以使用每个特征的权重来判断单词的属性是否影响困难模式百分比。从特征的权重图中可以得出，标签本身的时滞项影响最大，而单词属性的时滞

项次之。因此，当前游戏的困难模式百分比与前一时期给定的单词属性有关，几乎与今天的单词属性无关，尽管该比率主要受到其自身滞后条件的影响。

如第 5.2 节所述，如果前一阶段的单词是困难的，那么玩家将更倾向于不选择当前阶段的困难模式。更重要的是，由于玩家在选择之前并不知道当前时期单词的难度，当前时期的单词属性应该不会影响玩家的选择，这与我们模型的预测是一致的。

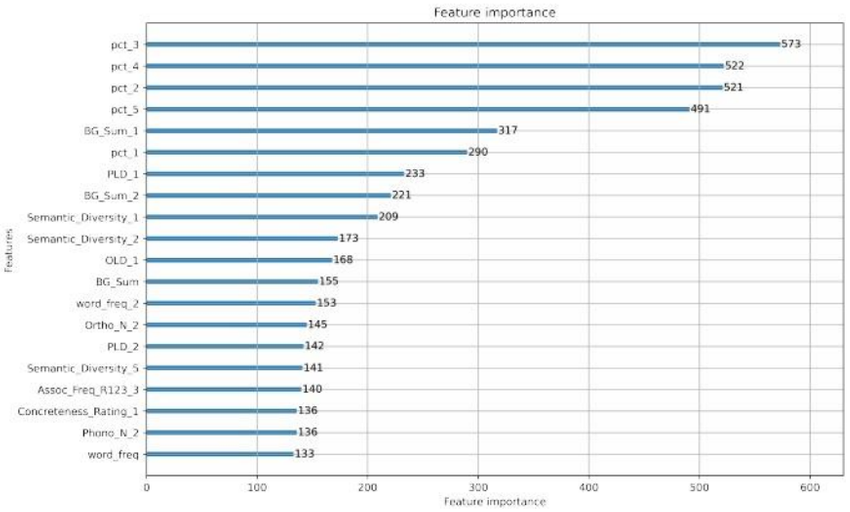

图 11:特征的相对重要性

7 模型 III:多输入-多输出回归模型

多输入-多输出回归模型主要用于解决给定输入示例时涉及预测两个或多个值的回归问题。一般有两种思路——一种是在机器学习模型的基础上训练多个回归量，另一种是在深度学习模型的基础上修改输出层数。经验表明，由于神经网络参数量较大，在数据量较小的情况下，使用特征工程和集成学习方法可能比直接使用深度学习模型有更好的预测结果。

7.1 白噪声验证

首先，我们对未来日期 2 的每个相关猜测尝试百分比进行了白噪声测试 1。如果数据通过了白噪声测试，就意味着它不受时间趋势的影响。因此，我们可以使用单词属性和分布的横截面数据进行后续的建模和预测。

以日期为横轴，以 1 ~ 7 次或更多次尝试的百分比为纵轴，做一个线条平滑的散点图。从图中可以看出，2-6 次的趋势是相对平稳的，而 1 和 7 次的百分比有一些不太平稳的值。究其原因，可能是在某些日子里有极易题和极难题。

同时，从 ACF 图中可以看出，2-7 次尝试的自相关系数相对较小，而 1 次尝试的自相关系数略大。

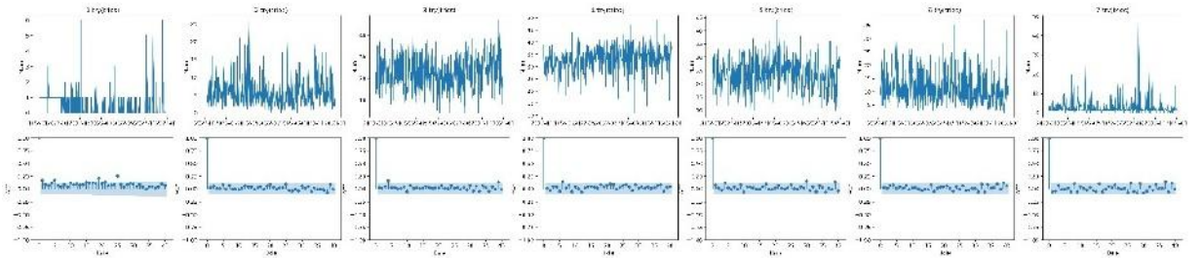

图 12:报告结果分布的散点图和 ACF 图

我们对这 7 个百分比进行了 Box-Pierce 白噪声检验，p 值均大于 0.05，通过了检验。

1 白噪声意味着一个纯粹随机的序列，昨天的值和今天的值之间没有关系

2(1 次尝试，2 次尝试，3 次尝试，…，6 次尝试，X)

7.2 模型介绍

因此，我们的分析试图比较以下两种模型在这个数据集上的效果，然后采用数据增强、特征工程等方法来提高模型效果。

•基于 GBDT 算法的多输出回归模型:

该模型的思想是基于 GBDT 算法训练多个回归量，分别拟合 1 次尝试到 7 次或更多次尝试(X)的 7 个因变量，回归量之间没有相关性。

•基于 MMoE 的多输出回归模型:

多任务学习是指同时训练多个目标函数的方法。它的主要优点是可以提高每个任务的学习效率和质量。此外，它还能有效克服任务噪声大、训练样本不足、数据维数高、数据集稀疏等缺点。

多任务学习的框架广泛采用底层共享结构，即底层的隐藏层在不同任务之间共享。这种结构本质上可以降低过拟合的风险，但效果可能会受到任务差异和数据分布的影响。

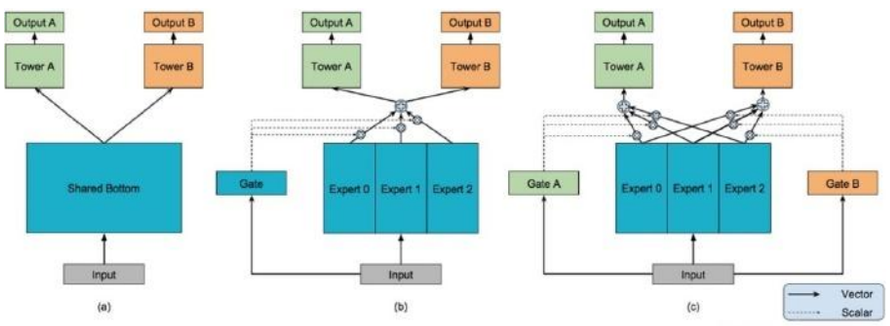

Figure 1: (a) Shared-Bottom model. (b) One-gate MoE model. (c) Multi-gate MoE model.

图 13:MMoE 的优势对比

因此，Google 的一个内容推荐团队提出了一种多门混合专家(Multi-gate Mixture- of-Experts, MMoE)多任务学习结构。MmoE 的改进之处在于，与基本的共享底部结构相比，它在不显著增加模型参数要求的情况下捕获了任务的差异。与所有任务共享一个门控网络(如上所示的单门 MoE 模型)相比，MMoE 中的每个任务使用单独的门控网络。每个任务的门控网络通过不同的最终输出权值实现对专家的选择性利用。不同任务的门控网络可以学

习不同的专家组合模式，因此模型考虑考虑捕获任务的相关性和差异性，非常适合预测这个问题中的玩家尝试次数。

我们输入单词属性并运行 7.2 节中描述的两个模型来预测报告结果的分布，但预测结果仍然需要优化。

7.3 模型细化

7.3.1 数据增强

对于多输入多输出问题，一般使用神经网络进行建模处理，但神经网络或深度学习需要使用大量数据进行训练。由于这个问题的总样本量是 359，很难获得足够的数据来训练神经网络。不过，或许可以通过数据增强来扩展数据集。

CTGAN 是基于深度学习的单表格数据合成数据生成器集合，能够从观测数据中学习，生成高保真度的合成数据[11]。需要注意的是，数据生成器不能限制变量之间的关系，生成的样本中从 1 次尝试到 7 次尝试的百分比之和可能与 100%相差甚远。所以我们使用 CTGAN 生成新的样本，过滤出符合条件的样本并合并到原始数据集中。最后，我们的样本量大约是 4000 个。

在获得增强样本后，对上述两个模型进行再训练。但是，我们发现训练结果并没有明显改善，可能是因为原始样本量太小。

7.4 特征工程

一般来说，数据和特征决定了机器学习的上限，模型和算法只接近这个上限。除了从数据的角度对模型进行优化之外，还可以最大程度地从原始数据中提取特征，即特征工程。

OpenFE 是一个为表格数据自动生成特征的新框架。[12]使用 OpenFE 作为工具，我们分别对未增强和增强的数据集进行特征工程，以生成包含新特征的数据集。在这个过程之后，我们使用新的数据集对上述两个模型进行了重新训练。幸运的是，结果得到了明显的改善。

7.5 模型结果与评价

我们对 2023 年 3 月 1 日 EERIE 这个词的预测如下:

两类模型的预测有效性表现出明显的差异。基于 GBDT 的多输出回归模型在训练集和验证集的总 MSE 上存在较大的方差。这表明当数据集较小时，机器学习模型的模型泛化性能较差。MSE 在使用基于 MMoE 的多输出回归模型时，训练集和验证集是比较接近的。更重要的是，它们明显低于前一种模型，显示出明显的优势。

出乎意料的是，数据增强导致两类模型的性能都出现了明显下降，说明新数据给模型训练带来了更大的噪声;特征工程对前者没有影响，但对后者的性能带来了小幅提升。因

此，我们最终选择了基于 MMoE +特征工程模式的多输出回归模型。经过参数优化，得到 2023 年 3 月 1 日单词 EERIE 的预测结果如下:

Percent in	1 try	2 tries	3 tries	4 tries	5 tries	6 tries	7 or more tries (X)	Total
Number	0.649	7.579	26.298	32.614	20.930	9.63	2.298	99.998

表 5:分布的预测结果

8 模型四:基于 k -均值聚类模型的 LightGBM 分类器

该模型的目的是根据难度对解词进行分类。在进行分类时，我们首先使用 K-means 聚类模型，根据平均解谜次数将数据集中的单词样本分成几组。然后，使用 LightGBM 算法根据给定单词的属性将它们分类到相应的组中。

8.1 k -均值聚类的概念

基于距离的聚类通常使用度量方法，如 K-means、k - medidoids 等。其中，目前最流行的启发式方法是 K-means 算法。因此，我们使用数据集中给出的 1 到 7 次猜对单词的百分比作为聚类基础，并运行 K-means 聚类模型。

给定一组数据点和所需的聚类数量，K-means 算法根据特定的距离函数将数据点迭代地移动到每个聚类域，一般实现步骤如下:

1.给定一个大小为 n 的样本词数据集，设迭代次数为 R，根据指定的聚类个数 k 随机选择 k 个词作为初始聚类中心，标记为 C_j (r)， j= 1,2,3，⋯⋯，k;r= 1,2,3，⋯，R。

2.计算样本中每个数据对象与初始聚类中心之间的相似距离 D(X_i, C_j (r));其中，i= 1,2,3，…， n，则形成一个簇 W_j ，如果满足式(5)

$$\sum_{i=1}^{n} |D(X_{i+1}, C_j(r)) - D(X_i, C_j(r))|^2 < \varepsilon \tag{5}$$

则 xi ∈W_j ,X_i 记为 W，其中ϵ任意给定的正数。

3.计算 k 个新的集群中心，计算公式如下:

$$C_j(r+1) = \frac{1}{n} \sum_{i=1}^{nj} X_i^{(j)} \tag{6}$$

计算聚类准则函数值的公式如下:

$$E(r+1) = \sum_{i=1}^{k} \sum_{w \in W_j} |w - C_j(r+1)|^2 \tag{7}$$

4.判断聚类是否合理，判别公式如下:

$$|E(r+1) - E(r)| < \varepsilon \tag{8}$$

如果结果合理，则迭代终止;而如果结果不合理，则返回步骤 2 和 3。

8.2 聚类模型构建

为了确定合理的聚类个数 k，我们首先让 k= 1,2,3…，10 作为实验聚类数，然后使用 Python 运行 K-means 聚类模型。最后，计算不同 k 值的结果的误差平方和(SSE)。以聚类数量 k 为横轴，SSE 为纵轴，绘制出带有趋势线的散点图如下:

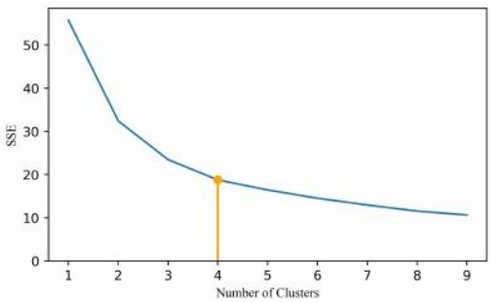

图 14:确定集群数量(determined the Number of Clusters)

我们必须在最小的 SSE 值和尽可能少的组数之间取得平衡。由图可知，当 k 值小于 4 时，随着聚类数量的增加，SSE 迅速减小。相反，当 k 大于 4 时，下降趋势不明显。所以达到上述平衡的最合适的 k 值是 4。

8.3 聚类结果评价

为了验证聚类的有效性，我们计算了每组的平均难度值，这是由玩家猜测单词的平均尝试次数来表示的。玩家尝试次数越多，给出的单词就越难猜。计算公式如下:

$$Average\ Number\ of\ Tries = \sum_{i=1}^{7} distribute_i \times i \tag{9}$$

每组的单词数、每组尝试猜单词的平均次数以及 skewness 如下表所示。

表 6:聚类结果评价

Group	0	1	2	3
No.of Words	34	127	125	57
Average Difficulty	3.574	3.951	4.326	4.800
Skew	0.320	0.246	0.0918	-0.181

从表中的数据可以看出，每组的平均尝试次数和偏度都有明显的不同，说明我们的聚类是比较有效的。

8.4 重要属性的识别

在上面的分析中，我们根据难易程度将单词分成了几组。接下来，我们将进一步进行深入分析，希望将单词难度的分类与其自身属性联系起来，解决“为什么这个单词比较难”的问题。

我们再次使用 LightGBM 模型，将聚类算法得到的 4 个聚类组作为标签，并使用第 5.2 节中列出的单词属性作为输入特征来训练 LightGBM 模型。训练结果(图 15)反映了单词的每个属性在影响单词难度等级方面的重要性。更重要的是，我们可以利用该模型将给定的单词分类到上面的聚类组中。

因此，这个词的大部分属性都与分类有关，而最相关的属性是 BG_sum、num_repeat、Semantic_diversity、PLD 和 Phono_N。单词属性对难度的影响也主要来自于这 5 个指标。

8.5 分类结果与评价

参考 7.4 的结果，“EERIE”一词的关键属性为:

我们将这些属性输入到 LightGBM 分类模型中，输出结果显示这个词属于第二类，也就是难度第二低的类别。

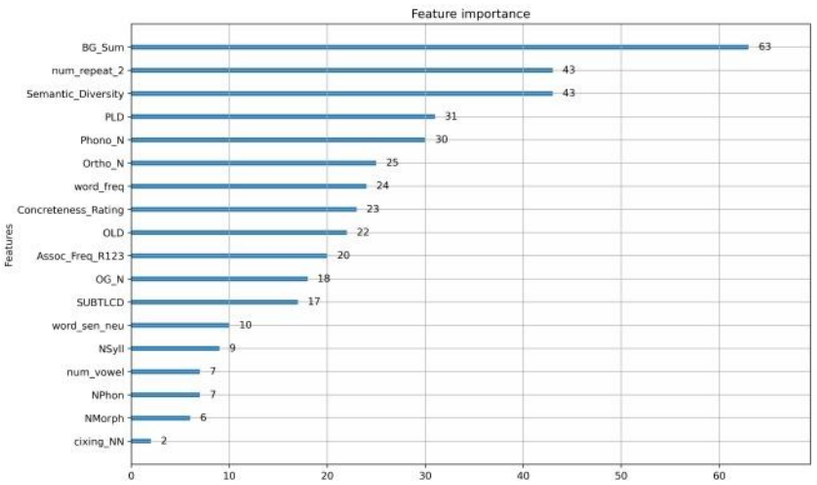

图 15:分类中特征的重要性

表 7:EERIE 部分关键属性

Word	Phono_N	BG_Sum	PLD	Semantic_Diversity	num_repeat
EERIE	8	11159	1.4	1.487	4

我们使用多种指标来衡量 LightGBM 分类模型的准确性。结果如下表所示。通过分析结果，我们可以断言我们的模型对数据集中的单词进行了相当准确的分类。

表 8:分类测量精度

	Accuracy	Precision	Recall	F1-score
Validation set	71.13%	71.85%	66.54%	69.09%
Traning set	98.28%	98.82%	98.79%	98.28%

9 数据集的其他有趣特征

1.从图 4 可以看出，游戏的用户数量在短时间内迅速上升，随后又迅速下降，反映了游戏的短期热度。此外，选择硬模式的玩家比例随着时间的推移逐渐增加，这表明玩家逐渐掌握了游戏。

2.图 16 是根据使用频率绘制的词云图。高频词以代词和助动词居多，其次是名词。

3.观察图 17 中用户尝试次数的分布，难度较低的单词的核密度函数是平坦且向右偏斜的;而较难词的核密度函数峰值较高，多集中在 4 次。

图 16:The Word Cloud

4.从图 18 可以看出，在各种词属性中，OrthoN 、PhonoN 、词频之间的相关性很强。

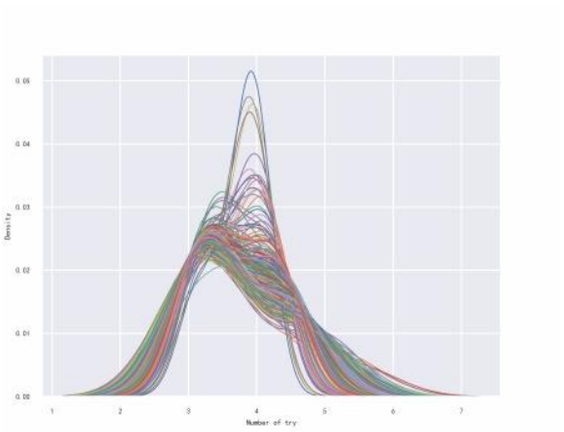

图 17:核密度函数图(Kernel Density

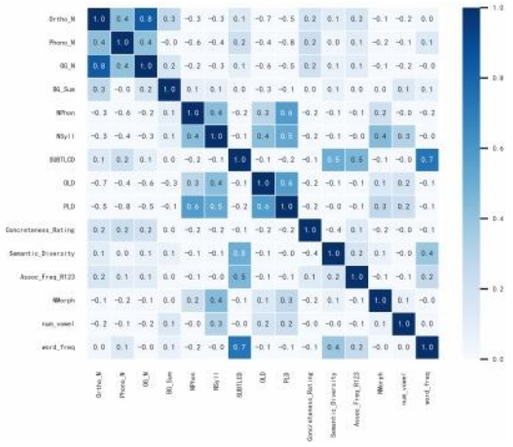

图 18:属性的相关性

10 灵敏度分析

10.1 问题 1 的敏感性分析

通过构建 LightGBM 模型，我们解决了哪些因素(同期或滞后)影响当前硬模式百分比的问题。为了测试模型的灵敏度，我们将随机扰动应用于 8.4 中导出的前五个影响属性。然后观察属性的相对重要性是否有显著变化。结果如下表所示:

其中，扰动范围 p 表示对于原始值 x，使用[(1−p)x， (1 + p)x]范围内的随机数对 x 进行重新赋值。无论取哪个数，都不会对预测硬模式百分比时属性的重要性造成很大影响。

表 9:问题 1 的模型敏感性分析

Attribute Name	Disturbance range p
BG_Sum_1	When p is less than 0.15, the ranking of word attributes remains unchanged
BG_Sum_2	Ranking of word attributes with the second-largest to fifth-largest influence keeps steady when p is less than 0.1
Semantic_Diversity_1	
Semantic_Diversity_2	
OLD_1	

从表中可以得出，在一定程度上，小规模的数据扰动不会影响单词属性的重要性排序，这意味着该模型通过了敏感性检验。

10.2 问题 3 的敏感性分析

为了分析本文分类模型的敏感性，我们从总样本中反复删除了 x 个随机选取的样本。随后，我们分析了分类结果是否发生了显著变化，如下表所示:

表 10:问题 3 模型的敏感性分析

heightx	The number of times among 100 repeated tests when the classification result didn't change
1	100
2	100
3	98
4	95
5	90

从表中我们可以得出，从整个数据集中随机删除 1-5 个样本 100 次后，100 次重复测试中分类结果没有变化的次数仍然达到 90 次以上。这意味着，在一定程度上，小规模的样本变化不会对分类结果产生显著的影响。问题 3 的模型也通过了敏感性测试。

11 优点和缺点

11.1 优势

1.我们充分利用了各种词属性，尽可能充分地反映了词的信息内容。

2.模型充分考虑了当期数据和滞后项。

3.我们尝试了各种机器学习和深度学习模型，并对它们的性能进行了深入的比较。

11.2 缺点

1.我们没有尝试采用基于深度学习的时间序列预测模型。

2.我们没有完全克服数据量少导致的预测精度差的问题。

12《致纽约时报拼图编辑的备忘录》

亲爱的编辑:

我们是来自 MCM 的专业商业分析团队。非常荣幸能受邀为您解答运营方面的问题。

我们非常高兴地看到 2022 年世界运动会取得了巨大的成功。但要保证游戏长期稳定运行，还需要对游戏的运行机制有深入的了解。应大家的要求，我们对从推特上挖掘出来的数据进行了分析，我们在这里自信地回答大家的问题:

首先，我们采用了时间序列预测模型，有效地解释了报告结果数量的变化。使用该模型，我们有 80%的把握认为 2023 年 3 月 1 日报告的结果数量应该在[10139,30808]之间。然后，通过采用机器学习模型，我们发现过去的单词属性对在困难模式下打出的报告分数百分比的影响有限，而今天的单词属性几乎没有任何影响。

然后，我们使用一个改进的深度学习模型来预测报告结果的分布。例如，对于 2023 年 3 月 1 日的单词 EERIE，从 1 到 7 次或更多次尝试，预测的概率分布应该是 [0.649,7.579,26.298,32.614,20.930,9.63,2.298](%)。这个结果通过了一组精度评估，所以它是可信的。

在此之后，我们建立了一个难度分类模型，该模型能够通过分析给定单词的属性来对其进行分类。在考虑单词 EERIE 时，分类模型推断它应该属于所有 4 个按难度分类的组中第二个最容易的组。更重要的是，我们推导出了与分类相关的前 5 个属性。

最后，我们进一步探索了该数据集的其他有趣特征，包括词云图、核密度分类图等。

通过分析世界最近的经营业绩，我们进一步对世界未来的经营提出了几点建议:

1.玩家数量有下降的趋势，所以应该引入新的玩法模式来吸引新玩家。

2.硬模式 pct 稳步增长。为了提高游戏体验，应该考虑提高某些天游戏的难度。

3.以后可以考虑根据我们上面得到的分类结果，设置 4 个不同的游戏难度。

希望我们的建议对您有所帮助，祝愿世界变得越来越好!

真诚的 MCM 团队# 2307946

References

[1] W. contributors, “Wordle. in wikipedia, the free encyclopedia,” Wikipedia, 2023.

[2] C. R. Nelson and C. I. Plosser, “Trends and random walks in macroeconmic time series: Some evidence and implications,” Journal of Monetary Economics, vol. 10(2), pp. 139 – 162, 1982.

[3] T. Jakaša, I. Andro ˇ cec, and P. Spr ˇ ci´ c, “Electricity price forecasting — arima model approach,” in 2011 8th International Conference on the European Energy Market (EEM), pp. 222 – 225, 2011.

[4] B. Laufer, “Ease and difficulty in vocabulary learning: Some teaching implications,” Foreign Language Annals, vol. 23, pp. 147 – 155, 1990.

[5] J. Jagoda and T. Boi ´nski, “Assessing word difficulty for quiz-like game,” pp. 70 – 79, 2018.

[6] D. Balota, M. Yap, and t. Hutchison, K.A., “The english lexicon project,” Behavior Research Methods, vol. 39, p. 445 – 459, 2007.

[7] M. Brysbaert and V. Warriner, A. B.and Kuperman, “Concreteness ratings for 40 thousand generally known english word lemmas,” Behavior Research Methods, vol. 46(3), p. 904 – 911, 2014.

[8] P. Hoffman, M. Lambon Ralph, and T. Rogers, “Semantic diversity: A measure of semantic ambiguity based on variability in the contextual usage of words,” Behavior Research Methods, vol. 45, p. 718 – 730, 2013.

[9] S. De Deyne, D. Navarro, A. Perfors, and et al., “The “small world of words” english word association norms for over 12,000 cue words,” Behavior Research Methods, vol. 51, p. 987 – 1006, 2019.

[10] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, “Lightgbm: A highly efficient gradient boosting decision tree,” vol. 30, 2017.

[11] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, “Modeling tabular data using conditional gan,” NeurIPS, 2019.

[12] T. Zhang, Z. Zhang, Z. Fan, H. Luo, F. Liu, W. Cao, and J. Li, “Openfe: Automated feature generation beyond expert-level performance,” 2022.