

世界赢家

摘要：《世界大战》是一款现象级网络游戏。它的出现强烈地引起了人们的关注。虽然看起来很小，但它背后隐藏的信息却是巨大而有意义的。获取和理解这些信息将有助于《纽约时报》更好地设计和运营世界。

我们构建了三个模型来完成任务。模型一使用 LSTM 来预测未来报告分数的数量。模型 II 使用七个 XGBoost 回归量来预测给定单词的百分比分布。模型 III 使用 RBF 核的 SVM 对词进行难易分类。基于我们的三个模型，我们可以提供一些建议来帮助改进世界。

具体细节如下：

模型一:LSTM 是一种改进的递归神经网络，可以解决其他神经网络无法解决的长距离依赖问题。我们为模型训练了报告分数数量的处理数据，并使用迭代方法预测到 3 月 1 日(2023 年)的数量。经过 150 次独立模型训练后，预测区间为[20745.72,22914.74]。另外，从硬模式与词属性的比例的线性回归中，我们也可以发现硬模式比与目标词之间没有相关性。

模型 II:为了获得某一天与特定单词相关的百分比分布，我们训练了 7 个独立的 XGBoost 模型。我们的模型的 R2 of 为 0.68，经过测试可以准确预测，不确定性很低。我们将“EERIE”应用到模型中，得到一个预测百分比分布，表明 EERIE 应该被认为是一个有问题的词。

模型 III:我们通过百分比分布的不均匀加权平均值来量化单词的难度，并将其分为三个级别:容易、中等和困难。然后用标记法对 SVM 模型与 RBF 核进行拟合，得到准确率分数为 0.6556,F1 分数为 0.6634。EERIE 的分类结果也比较硬，与模型 2 的结果一致。

除了这三个模型，我们还从数据集中发现了一些有趣的观察结果，其中一个讨论了人类思维和机器学习之间的差异。

最后，我们给《纽约时报》世界版编辑写一封信，包括我们的模型、结果和建议。我们希望这封信将成为世界进一步发展的宝贵参考。

关键词:世界;LSTM;递归回归;XGBoost;工程特性

目录

世界赢家 1

1 介绍 3

 1.1 背景 3

 1.2 问题重述(Restatement of Problem 3

 1.3 文献综述4

 1.4 我们的工作4

2 假设5

3 符号5

4 模型的制定6

 4.1 数据归一化8

 4.2 LSTM 的实现 8

 4.3 来自结果的预测区间9

 4.4 Hard-mode Percentage 与 Words 的相关性9

5 任务 2:预测百分比分布的模型10

 5.1 用于衡量报告结果的的性能的特征 10

 5.2 特征工程10

 5.3 XGBoost 模型训练 11

 5.4 模型中涉及的不确定性 11

 5.5EERIE 预测分布 12

6 任务 3:单词分类 13

 6.1 特征工程14

 6.2 模型构建与预测 14

 6.3 模型评价14

7 任务 4:其他有趣的发现15

 7.1 人类在世界大战策略上的根本差异 15

 7.2 其他有趣的特性 16

8 强项和弱项17

 8.1 优势 17

 8.2 缺点 17

References 18

1 介绍

1.1 背景

如今，一款名为“世界”的益智游戏以其轻松的方式和令人困惑但吸引人的黄色、绿色和灰色方块风靡全球。

一般来说，这是一款要求玩家在不到 6 次的时间内猜出正确单词的游戏。玩家只能猜出一个官方认可的单词。玩家每猜一次(猜错次数不超过 6 次)，就会得到以黄、绿、灰三色方块形式出现的提示。绿色表示猜对;黄色表示字母正确，位置错误;而灰色则表示这个字母没有出现在单词中。Wordles Hard Mode 通过强制玩家在随后的猜测中使用之前找到的正确字母，使游戏更具挑战性。“世界”的示例如图 1 所示。

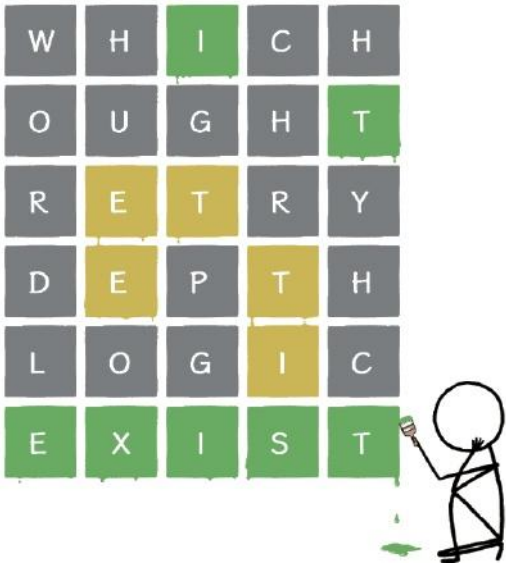


图 1:世界谜题的例子

为了帮助纽约时报更好地了解“世界大战”的难度和受众，我们希望利用 2022 年球员在 Twitter 上发布的日常表现数据构建一个预测模型。换句话说，为了让“世界”更好地发展，我们应该使用预测模型，对未来玩家的数量、报告结果的分布以及单词难度的测量进行合理的估计。

1.2 问题重述

构建一个预测模型来解释 2023 年 3 月 1 日报告结果数量的总变化和预测区间。

验证单词属性是否会影响在 Twitter 上发布困难模式结果的玩家比例。

给定固定的单词，预测未来日期(1,2,3,4,5,6,x)的相关百分比，并解释不确定因素。2023 年 3 月 1 日应用该模型，具体单词为“EERIE”。

开发并总结一个模型，根据难度对解决方案单词进行分类，并识别给定单词的属性。应用该模型对“EERIE”一词进行分类，并对其准确性进行了讨论。

列出并描述该数据集的其他有趣发现。

1.3 文献综述

该模型有两个主要任务:沿时间轴预测报告结果的数量及其相关百分比(1,2,3,4,5,6,X)，并测量给定单词“EERIE”的难度等级。

在时间序列领域，模型主要集中在长短期记忆(Long- Short-Term Memory, LSTM)，这是一种人工神经网络，适用于处理和预测时间序列中具有很长间隔和延迟的重要事件 [1]。通过使用 LSTM，可以获得比其他传统技术更好的性能。更具体地说，在预测时间序列时，与 ARIMA 相比，LSTM 获得的错误率平均降低在 84%到 87%之间[2]。

在单词分类方面，Laurent 已经通过决策树找到了解决“世界”谜题的最优策略 [3]。他指出，最优的过程平均需要 4 步，最多需要 6 步来匹配单词。Selby[4]基于他的决策树实现了最优策略，并提供了一个文件来说明，其工作原理如图图 2 所示。以图 2 中的例子为例，如果我们想要获得目标单词“nymph”，我们应该遵循红色轨道，预期深度(猜测次数)为 3。

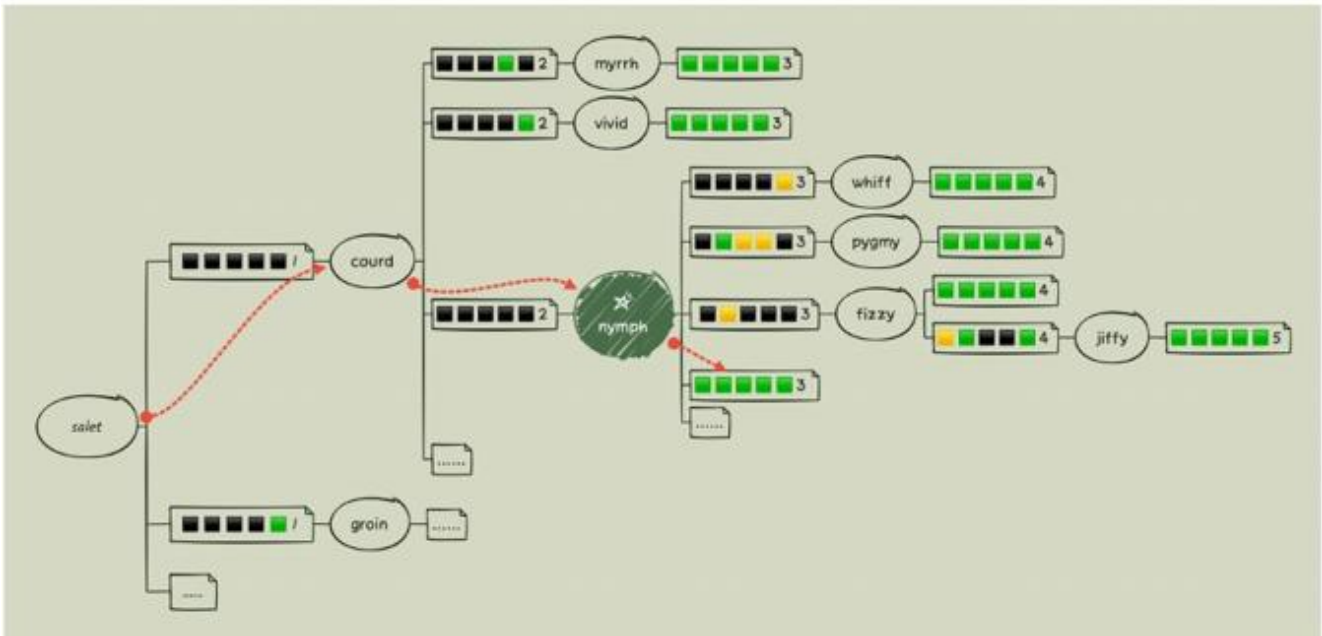


图 2:决策树的工作过程

1.4 我们的工作

图 3 展示了我们的工作流程:

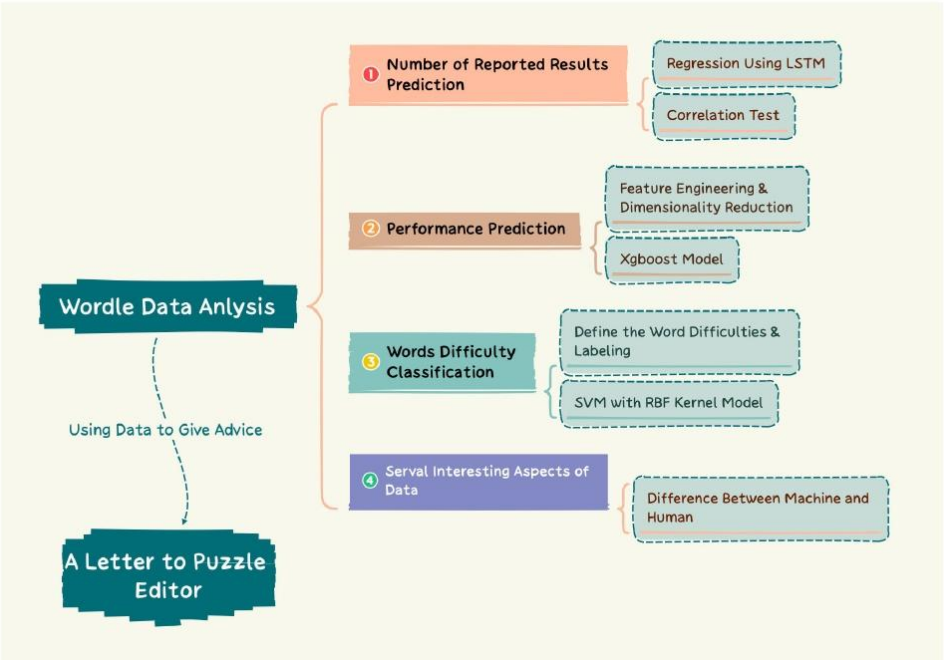


图 3:The Process of Our Work

2 假设

为了简化问题，我们做了如下假设。每一个假设都是合理的。

忽略极端事件的影响，如假期和自然灾害，因为这些事件很少发生。所以，我们可以忽略它对我们模型的影响。

2020 年词频表可以代表所有参与者日常使用的单词频率。而且在过去的三年里，使用频率并没有明显的变化。

玩家上报的成绩均为自己独立制作，无作弊行为。

其他具体的假设，如有必要，将在我们建立模型时被提及和说明。

3 符号

表 1 显示了方程中应该使用的符号:

Symbol	Definition
C_t	The output of the long-term memory
h_t	The output of the short-term memory
μ	Average of the sample data
sd	Standard deviation of the sample data
i_{try}	Percentage of i try, i can be 1 to 6 or X
w_t	Weighted average of performances in reported results
E_w	Unequally weighted average of score distribution

4 任务 1：预测结果的基本模型

4.1 数据清理

通过对数据的统计和处理，发现样本数据中存在两种错误：第一种是词汇错误，包括长度不一致、空字符等问题。我们根据官方推特进行了更正，如表所示。2；第二个问题是数据错误。2022 年 1 月 30 日报告的总分数是一个异常值，因为它缺乏大小。所以我们回顾了那天的官方推特，发现正确的数字是 25,569。

表 2：纠正词汇错误

Previous Word	rprobe	clen	tash	favor (with space)	na?ve	marxh
Corrected Word	probe	clean	trash	favor	naive	marsh

4.2 模型的制定

与其他类型的人工神经网络不同，LSTM 结合了长期记忆和短期记忆，以确保模型沿时间序列的稳定性和准确性。这种独特的特征可以在图 4 中以重复单元的形式表现出来。在这里，我们将简要描述我们的预测模型[5]中使用的原理。

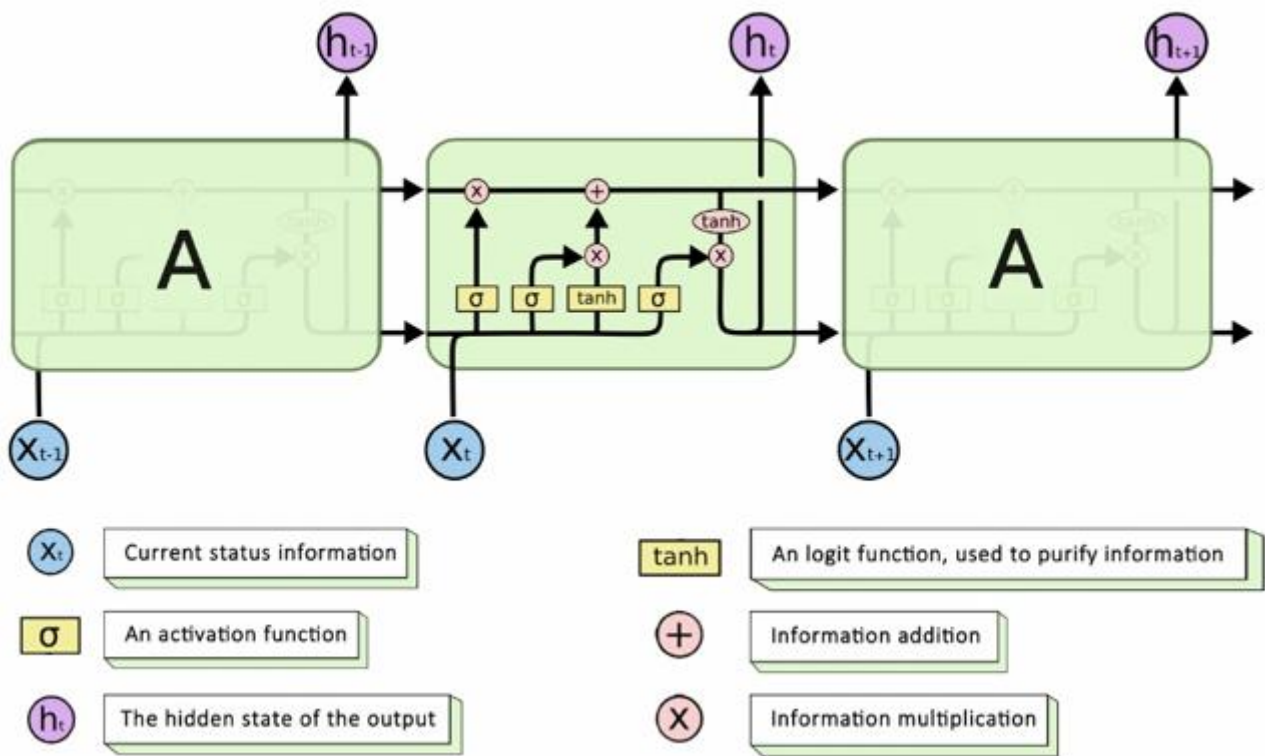


图 4:LSTM 原理图

4.2.1 长期记忆和短期记忆

为了理解长期记忆的输出，我们应该实现遗忘门的功能（图 5）。遗忘门用于比较上一层的隐藏状态与当前状态信息 x_t 之间的信息，并选择权重较大的状态状态。

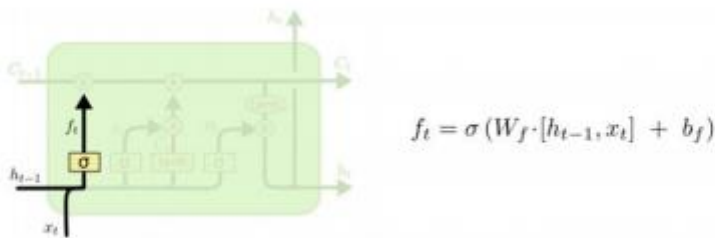


图 5：遗忘门

长时记忆输出(C_t)为上图输出，公式为:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{1}$$

可以用两部分相加来描述。第一部分是遗忘门输出 f_t 和 前一层 C_{t-1} 的长时记忆输出的重叠，第二部分是从当前状态信息中提纯的信息。短期记忆输出(h_t)为下图输出，公式如下图(图 6)所示

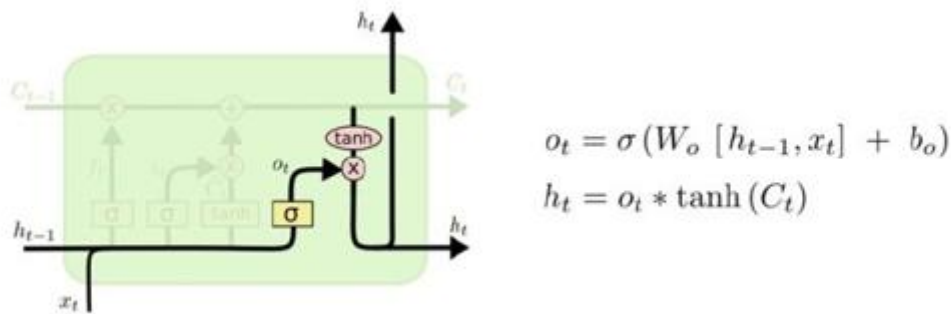


图 6:短期记忆输出

可以用两部分的乘法来描述，第一部分是和遗忘门一样的激活过程，第二部分是从当前长期记忆输出(Ct)中纯化出来的信息，最终的结果也会呈现激活信号和纯化后的长期记忆输出的重叠。

4.2.2 数据归一化

LSTM 模型的输入数据应归一化。然而，报告分数的数量分布是相当不平衡的。例如，从 07/24/2022 到 11/01/2022 的报告数变化为 39813–27502 = 12311。然而，按最大数字缩放，变化比例仅为 3.4%。换句话说，传统的 Min-Max 归一化在这种情况下将不起作用。而且，报告分数的数量大致遵循 gamma 分布，其中包含了自然常数项。作为解决方案，将在第一步中对报告分数的数量应用自然对数。将差值缩小后，通过 极值 方法进行归一化，最终得到 LSTM 模型的输入数据，总过程如图 7 所示。

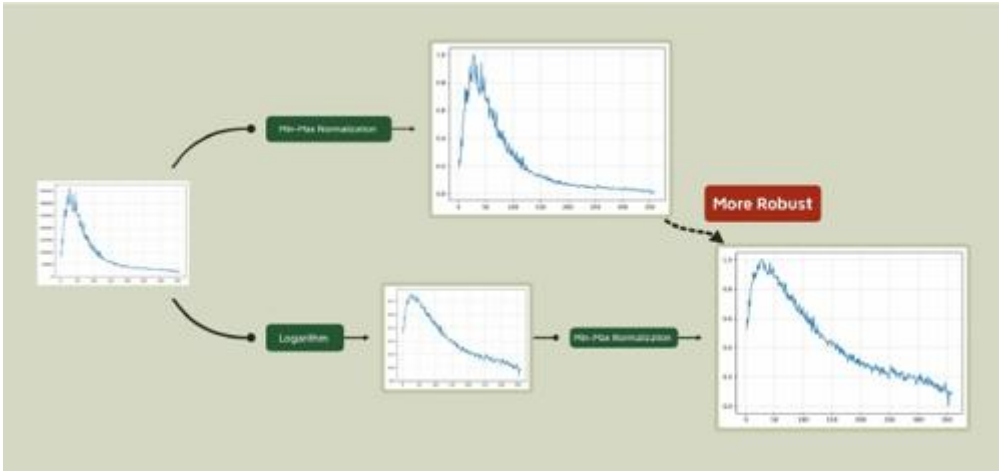


图 7:数据归一化过程

4.2.3 LSTM 的实现

在我们的任务中，我们用 Pytorch 实现了 LSTM 神经网络。我们的目标是基于 359 天的报告分数的样本数据预测 3 月 1 日的报告分数数，它是一个单输入-单输出的 LSTM 模型。此外，基本 LSTM 一次只能做出一个预测。然而，我们的目标是预测 60 天后的期望值，因此我们递归地应用了 60 次回归。这个递归过程可以通过添加最新的输出来持续更新。

由于样本小，我们只使用了一个 LSTM 层和一个全连接层。此外，经过交叉验证，我们发现 LSTM 单元的最佳数量为 5 个。然后，我们重点调整了“回看”参数，该参数决定了短期记忆需要记录的天数，最终找到了 33 天的回看才能获得最佳性能。

由于神经网络的随机性，在经过 150 次训练后，我们得到了 2023 年 3 月 1 日的报告数预测如下，以及预测的 60 天的平均得分报告数(图 8)。

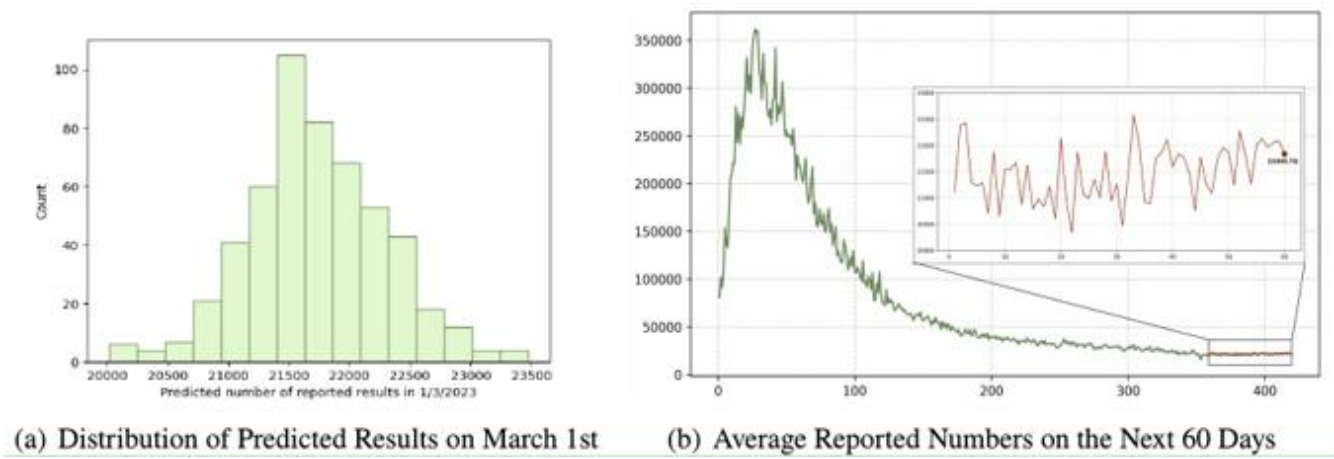


图 8:模型预测结果图(The Prediction Result Graph)

4.2.4 来自结果的预测区间

创建的 LSTM 模型训练了 150 次。然后，通过统计计算，我们可以得到 2023 年 3 月 1 日变化的置信区间。预测数据的平均值为 21,843.73，标准差为 560.21。我们用|z|值 1.96. 选择 95%置信区间，CI 可以通过以下方式计算:

$$[\mu - |z| * sd, \mu + |z| * sd] \tag{2}$$

因此，预测区间为[20745.72,22914.74]，置信度为 95%。

4.3 Hard-mode Percentage 与 Words 的相关性

如果我们从样本数据中考虑词汇属性，相关分析的结果会有偏差。因此，我们提取单词本身的一些基本属性。这里我们使用了四个属性:每日频率、首字母、连续字母和平均字母频率。

每日频率:从莱比锡语料库(Leipzig corpus Collection)[6]中，我们找到了一个单词列表，统计了 2022 年日常生活中单词的频率。然后，我们提取所有五长英语单词，并将其归一化排名作为第一个特征。

首字母:一个二元变量，显示首字母是否为元音。

连续字母:另一个二进制变量表示单词是否包含连续字母，如“carry”中的“rr”。

平均字母频率:通过应用《简明牛津词典》(1995 年第 9 版)字母表中字母的频率，我们创建了一个变量来记录每个单词的五个字母之间的平均频率

我们对困难模式比率进行了线性回归，以确定这四个词汇属性的影响。详细结果如下图 9 所示:

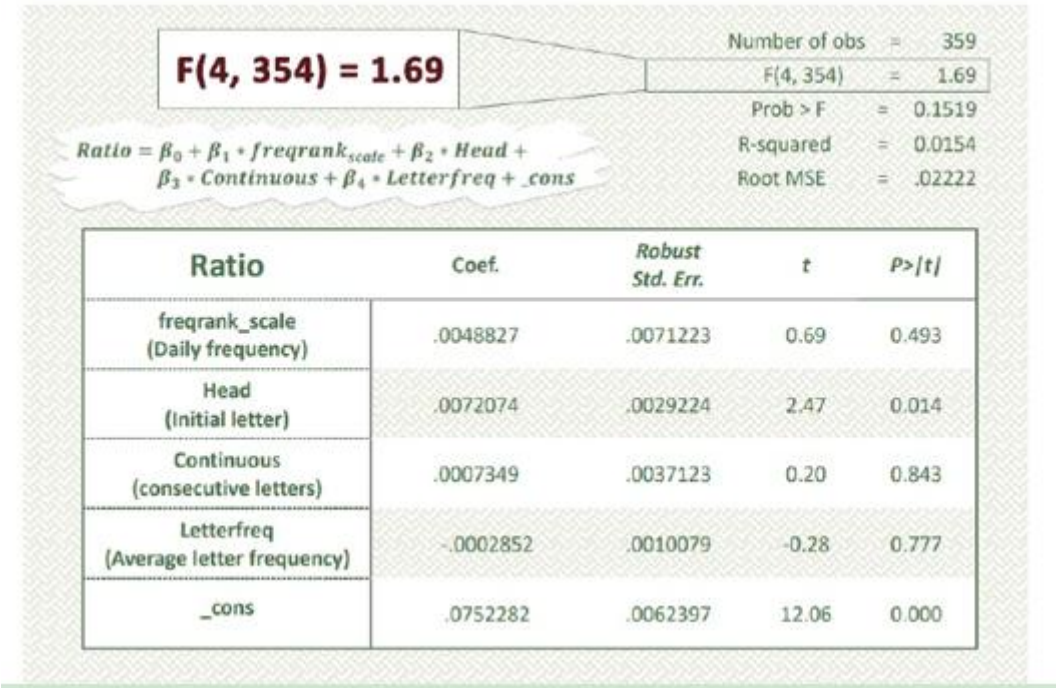


图 9:对 Ratio 的线性回归

f 统计量为 1.69，小于 2.397 ($\alpha = 0.05$)。因此，在 5%显著性水平下，单词不会影响困难模式中报告的分数的百分比。此外，如果我们转而分析这四个 t 统计量，结论也是一样的。

5 任务 2:预测百分比分布的模型

5.1 用于衡量报告结果的性能的特征

注意，估计模型主要有两个方面，时间和词的难度。在这种情况下，我们首先验证时间和百分比分布之间的相关性。我们使用加权平均值(wt)来表示报告结果中的性能，其中 $w_t = \sum_{i=1}^7 i * i_{try}$ 。然后我们对 W t 进行了线性回归，发现 p 值为 0.457，远远大于 0.05。结果显示成绩和时间之间没有显著的相关性，所以我们考虑成绩和单词难度之间的关系。我们将使用 4.3 章中提到的那四个属性。此外，我们增加了两个新的词汇属性 :深度和重复字母的最大数量。

深度:正如文献综述中提到的，深度可以显示在最优决策树的帮助下找到正确答案的预期时间——深度越大，回答单词就越具有挑战性。

最大同字母数(charNum):基于《worldle》游戏机制，当得到绿色或黄色方块响应时，很难想到同一个字母。我们倾向于尝试一个还没有得到回应的字母。重复的单词也会减少提供给玩家的信息量。例如，对于目标单词“wheel”，如果玩家已经知道正确答案包含“e”、“h”、“l”和“w”，他们会更愿意猜测“whale”而不是“wheel”。因为“鲸鱼”比“轮子”能提供更多的信息。而且，相同字母的数量越多，问题就越复杂。

5.2 特征工程

在对数据进行扫描后发现，对于“日频次”这一特征，列表中有些单词的排名非常大。这些排名非常庞大，应该被认为是异常值。为了减少模型中异常值的影响，使用 RobustScaler 对数据进行归一化。

到目前为止，衡量单词难易程度的特征都是由人类直觉来定义的。这些特征应该进一步分析，确定它们对单词难度的贡献。因此，使用递归特征消除(RFE)算法对这 6 个特征进行排序。最后，选取“日频率”、“CharNum”和“字母频率”作为三个主导特征。

5.3 XGBoost 模型训练

如图 10 所示，我们使用七个 XGBoost 模型来训练固定单词的每个相关百分比(1,2,3,4,5,6,x)，使用“Daily frequency”，“CharNum”和“Initial letter”。在训练过程中，通过交叉验证确定每个 XGBoost 模型对应的超参数。使用确定的超参数，模型的确定系数(R2)为 0.68，表明该模型确实具有预测能力。

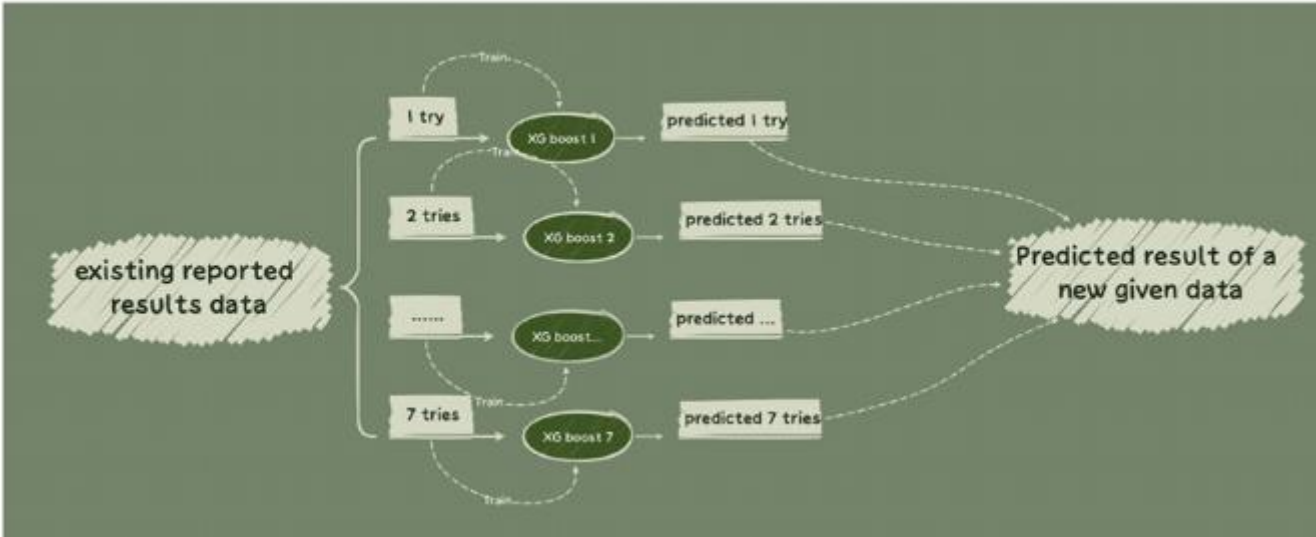


图 10:XGBoost 训练过程

5.4 模型中涉及的不确定性

由于我们的模型是基于 XGBoost 的，所以我们需要使用灵敏度分析来衡量模型的不确定性。我们选择了三个难度相似的单词:“Royal”、“Sport”和 “Prove”，因为它们的日常频率排名和平均字母频率之间的差异都小于 2%。另外，“Royal”是 2022 年 4 月 12 日发布的字谜，这意味着它的百分比分布是已知的，而剩下的两个仍然是未知的。我们使用我们的模型来预测这两个未知的百分比模型，并比较它们与“Royal”分布的差异。

使用分布的加权平均是比较分布之间差异的一种直接的方法，但同样的加权平均会造成信息的损失。例如，3 次尝试和 4 次尝试之间的差距小于 6 次尝试和 x 之间的差距，经过深思熟虑，我们最终确定了一个不均匀加权平均 E_w 每个单词，它是:

$$E_w = 1_{try} + 3 * 2_{try} + 5 * 3_{try} + 6 * 4_{try} + 7 * 5_{try} + 9 * 6_{try} + 11 * X_{try} \tag{3}$$

比较不同单词上分布的差异。

表 3:敏感性分析

	$Freqrank_{scale}$	charNum	letterFreq	E_w
royal	-0.3638	1	6.10164	605
sport	-0.3624 (0.38%)	1 (0%)	6.1195 (0.29%)	601.3349 (0.66%)
prove	-0.3620 (0.50%)	1 (0%)	6.01592 (-1.4%)	607.6483 (0.44%)

如表 3 所示， E_w 的变化范围在 1%以内，因此存在不确定性的型号低。

不确定性也可能由预测损失引起。当我们训练 7 个独立的 XGBoost 模型时，7 次尝试的总和百分比不可能完全等于 100%。25 个随机选择的测试数据的预测值如图所示。

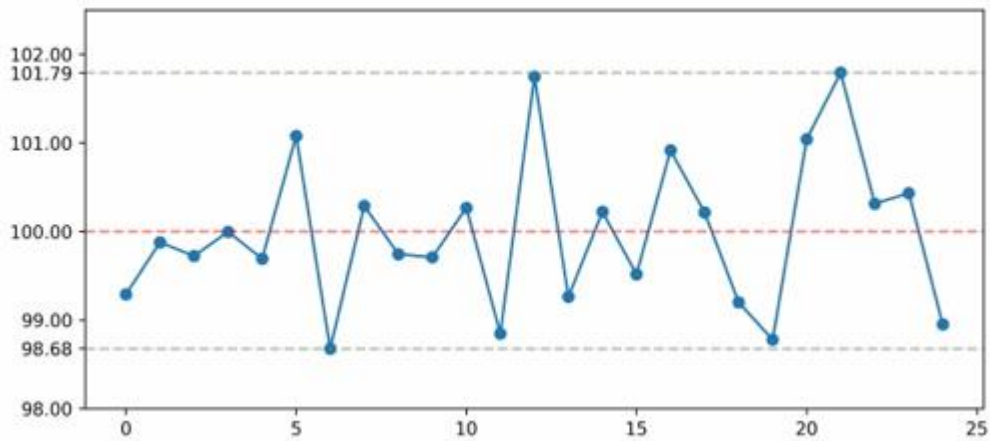


图 11:演出中百分比的总和

25 个预测值的总和落在(98,102)区间内，对 100 的方差为 0.71。因此，不确定性相对较小，说明不确定性较低。

5.5EERIE 预测分布

将“EERIE”带入所建立的模型，得到如下结果(图 12)。

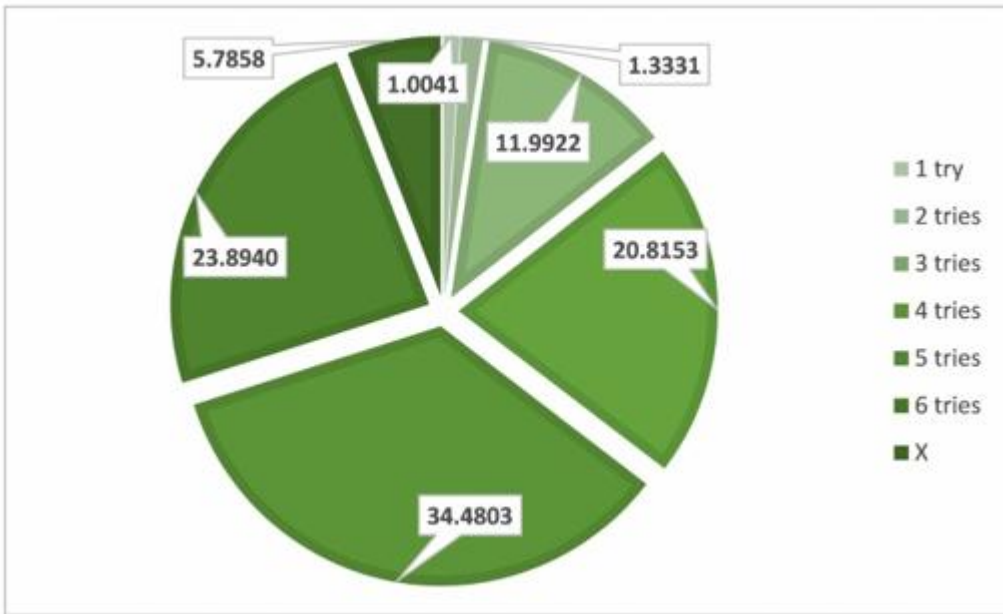


图 12:EERIE 报告分数预期分布的饼状图

这个预测结果显示了一个比较大的 Ew 值，为 709.9076，也就是说这个词是初步困难的。难度分析将在任务 3 中进行。如上所述， $R2$ 相对较高，不确定性较小，我们对“EERIE”的预测有较高的置信度

6 任务 3:单词分类

6.1 Word 难度的定义

在这个任务中，需要根据难度对解词进行分类。由于报告结果的分布主要取决于单词的难易程度，因此我们决定使用第 5.4 章提到的不均匀加权平均值作为单词难易程度的度量。

为了标注所提供的数据，我们首先计算了每个单词的不均匀加权平均值。我们发现，近 60%的点落在 600 到 700 之间，而另外 40%的点几乎均匀地分布在两边(图 13)。因此，我们将分类标签确定为“容易”、“中等”和“困难”(表 4)。不均等加权平均值代表的越小， i 小 i 的百分比就 try 越高，换句话说，这个词更简单。

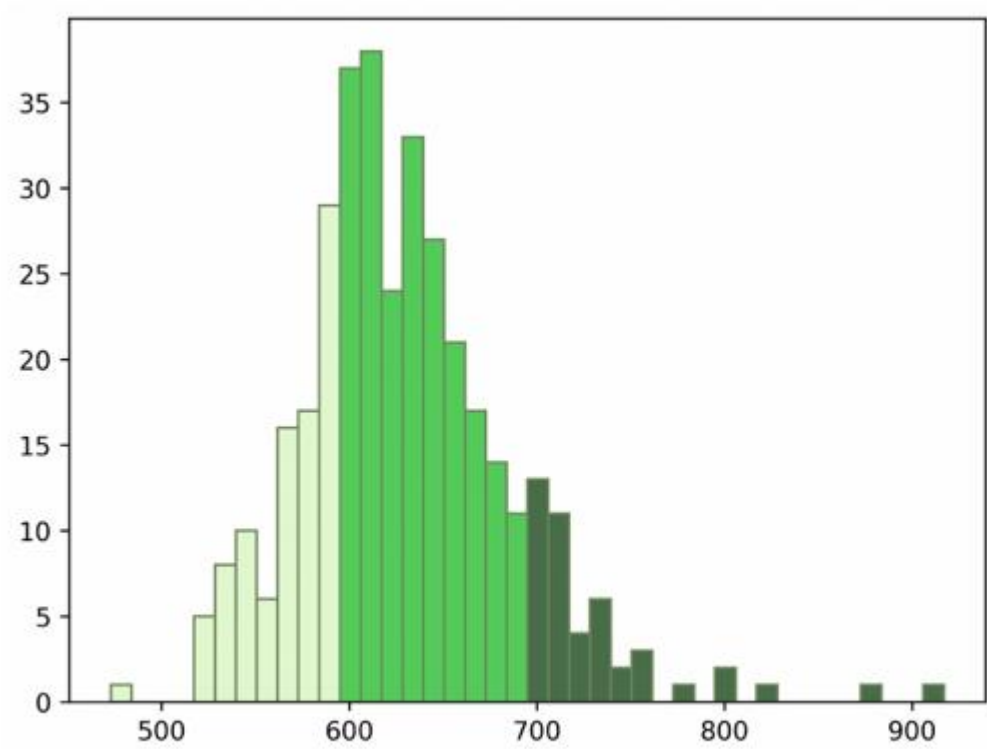


图 13:加权平均性能分布

表 4:难度标签标准

w_t	<600	600-700	>700
label	easy	medium	hard

6.1 特征工程

由于单词难度的分类与性能的分布有关，所以我们在第 5 章中仍然会使用 RFE 算法选择的三个特征。根据这三个特征和分配的标签，359 个提供数据的分类图如图 14(a)所示:

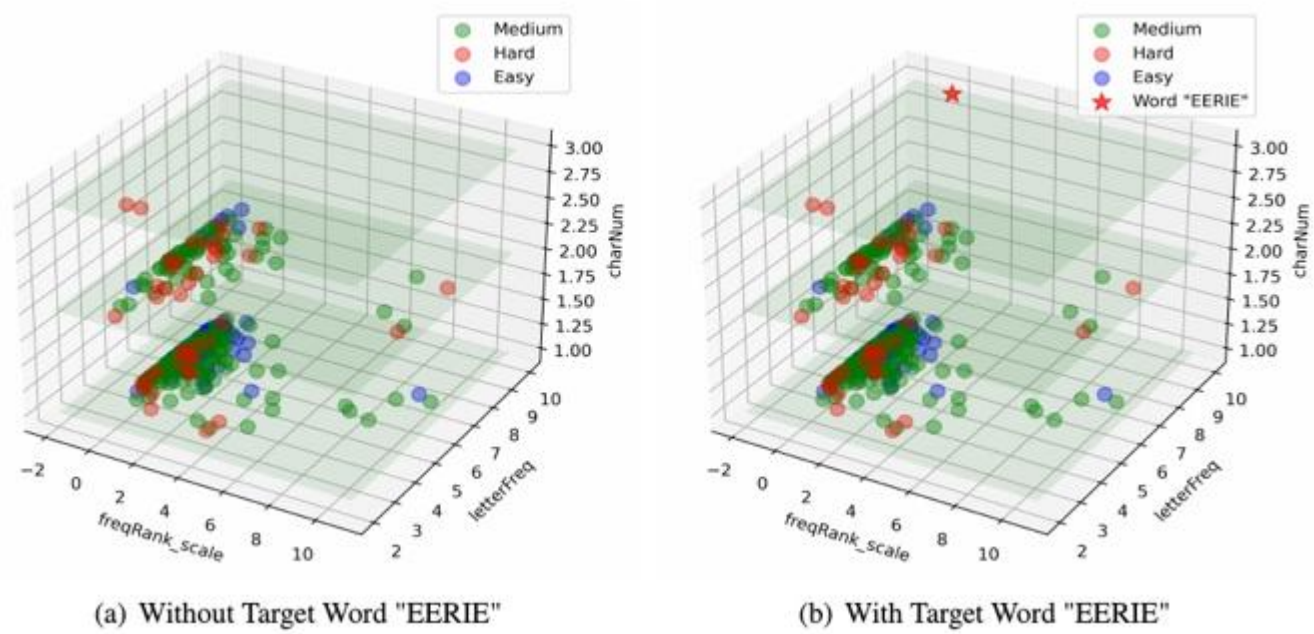


图 14:加权性能分类图

从图中可以看出，蓝点(容易)集中在 freqRankisscale 低(生活中常见)、letterFreq 高(写作中更常见)、charNum 低(重复字母更少)的区域。红点(难)分布在相反的区域，而绿点(中位数)大多在它们之间的区域。

6.3 模型构建与预测

我们将样本数据按 3:1 的比例分成训练集和测试集，训练 RBF 核支持向量机模型。EERIE 的三维特征为(0.0226,9.7215,3)，将其带入模型后，我们得到“EERIE”的预测标签为“hard”，这也与第五章 (709.9076>700)对其性能分布的不均匀加权平均结果一致。上图为“EERIE”在样本数据空间中的位置(图 14(b))

6.4 模型评价

下表 5 显示了分类模型评估的四个重要指标。它们都在 0.6 以上，反映了模型的高准确率(可以描述为高于 60%)。

Table 5: Four Criterias for Evaluating Model

Precision score	Recall score	F1 score	Accuracy score
0.6762	0.6556	0.6634	0.6556

混淆矩阵(图 15(a))表明，“容易”和“中等”的分类很好，但“硬”的分类相对较弱。

ROC 曲线(图 15(b))表示三种类型的平均曲线。ROC 曲线越接近左上角，其灵敏度越高，误诊率越低，因此该模型具有较好的预测性能。

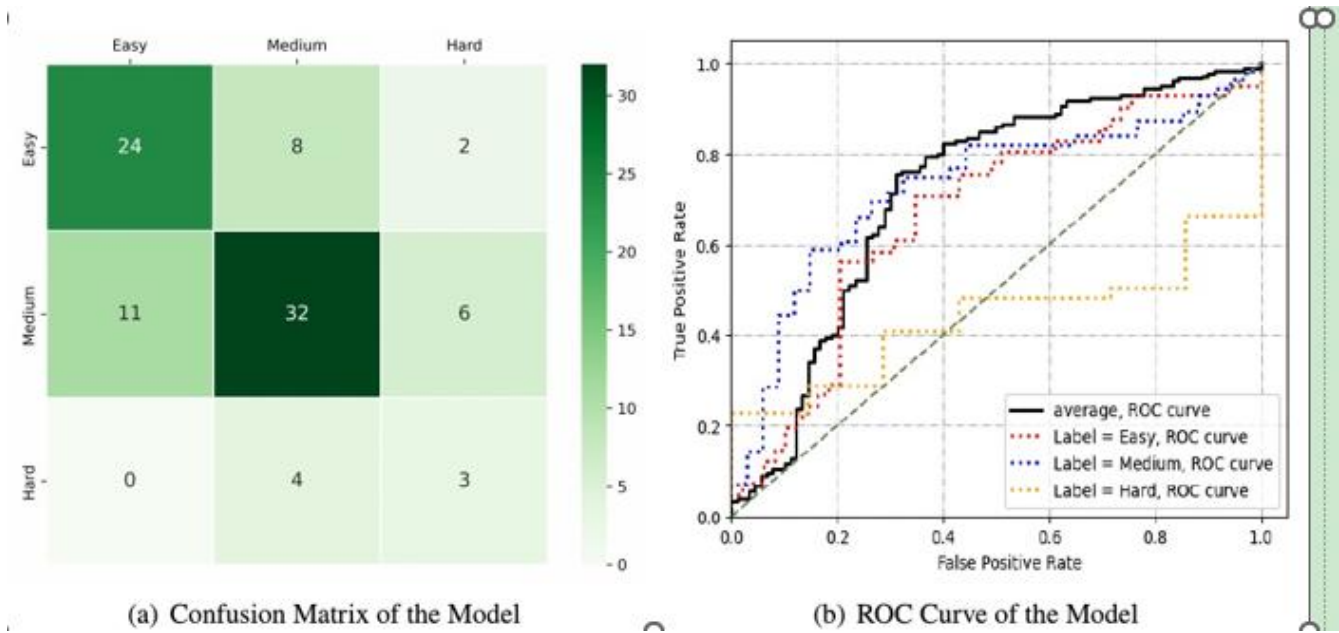


图 15:加权性能的分类图

7 任务 4:其他有趣的发现

7.1 人脑与机器学习的根本差异

正如文献综述中提到的，有一种由决策树算法开发的最优策略，其中唯一影响每个单词使用的尝试次数的是树中的单词深度。然而，当我们分析报告结果的百分比分布时(图 16)，我们发现 Depth 与 wintt 数据集之间的相关性非常低，甚至小于 charNum 与 w t 之间的相关性。

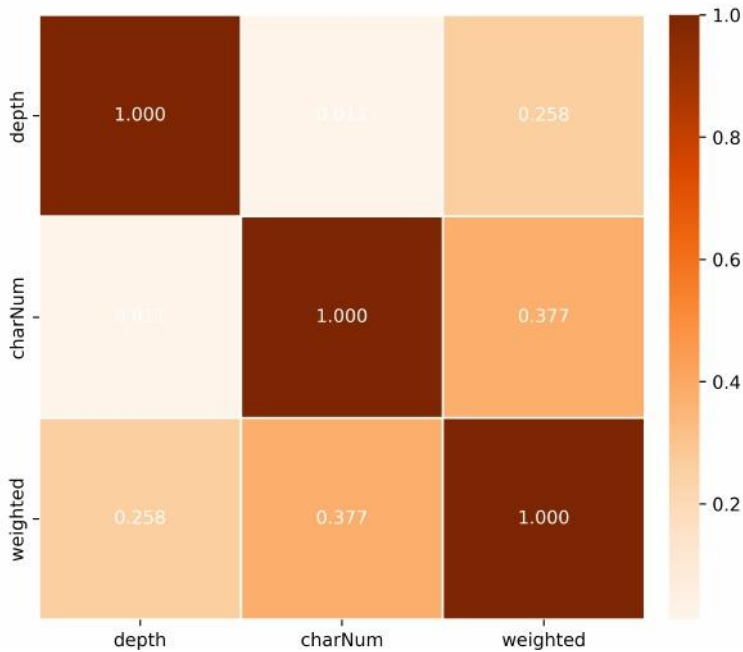


图 16:wt、CharNum 和 Depth 之间的相关性热图

这表明深度特征对《世界大战》玩家的表现贡献不大。

这种不相关性表明，当人们玩《世界大战》时，他们不太可能借助机器开发的最佳策略来解决谜题。相反，人类有自己的策略和思维来处理谜题。给定的数据说明了机器“思维”和人类思维的区别，这是一个可以进一步研究的发人深省的方面。

7.2 其他有趣的特性

随着时间的推移，玩家在这个问题上的表现几乎没有提高。这是一个反直觉的特征。因为从直觉上来说，游戏发布的时间越长，游戏玩家对游戏的理解和表现就越好。可能是样本数据不足的问题造成的。

假期也会在一定程度上影响球员上报的分数和表现。例如，在圣诞节当天，无论是报告结果的数量还是硬模式的数量，都明显低于相邻日子的数据。

我们估计，在硬模式下的百分比与第 4 章中单词的属性没有相关性。但是从图 17 可以看出，未来难模式的百分比有持续增长的趋势。



图 17:难度模式百分比

根据这些信息，我们可以推测，难度模式的百分比与时间有关，受玩家对游戏机制熟悉程度的影响。随着时间的增加，玩家的难度往往会越来越高。

图 18 显示了(1,2,3,4,5,6,X)百分比分布的散点图，黄色的点表示 4 次尝试的百分比，总是在图的顶部。无论单词的时间和难度如何变化，最大占用的尝试次数总是 4 次。验证了最优决策树的平均最优解为 4 的结论。

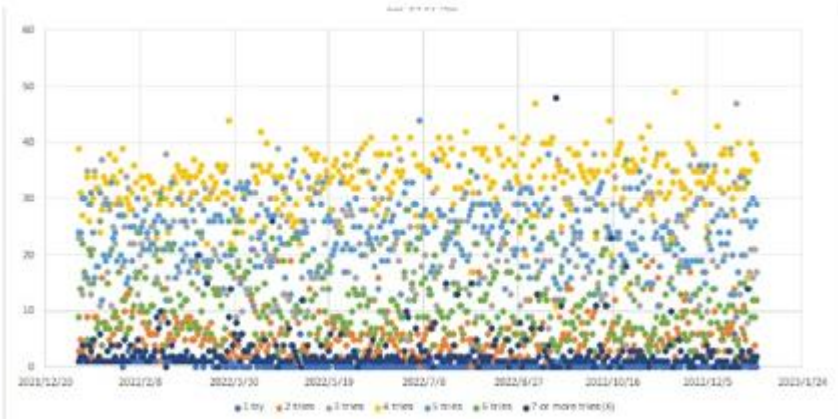


图 18:The Scatter Diagram of Distribution in performance

8 强项和弱项

8.1 优势

我们使用的模型(LSTM 神经网络)比经典方法(如 ARIMA)具有更高的精度和更好的性能。此外，它既可以处理单个数据点，也可以处理整个数据序列，这使得它适合处理具有时间序列的数据集，并处理任务 1 中的主要问题。

我们在 task 2 和 task 3 中得出的结果是完美匹配的，这表明我们的模型具有很高的准确性和高度的置信度。更具体地说，任务 2 的结果表明，人们在处理世界上的“EERIE”时表现不佳。同时，任务 3 的结果显示，“EERIE”是“hard”标签中的一个难词。

8.2 缺点

没有考虑特殊日子(如假期或 MCM 比赛)的影响。

数据由玩家报告的推文收集，可能存在信息差距并影响最终的预期结果。

由于样本给出的百分比数据是四舍五入的，当我们将数据代入模型时，可能会出现数据之间的精度误差，并且表现中的百分比总数可能不是 100。SVM 分类模型在识别“硬”词方面很弱。

References

- [1] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [2] Siami-Namini, S., Tavakoli, N., & Siami Namin, A. (2018). A comparison of Arima and LSTM in forecasting time series. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). <https://doi.org/10.1109/icmla.2018.00227>
- [3] Poirrier, L. (2022, January 23). Laurent's Notes. Mathematical optimization over Wordle decision trees – Laurent's notes. Retrieved February 18, 2023, from <https://www.poirrier.ca/notes/wordle/#decision-trees>
- [4] Selby, A. (2022, January 19). The best strategies for wordle. The best strategies for Wordle - Things (various). Retrieved February 18, 2023, from https://sonorouschocolate.com/notes/index.php?title=The_best_strategies_for_Wordle
- [5] Olah, C. (2015, August 27). Understanding LSTM networks. Understanding LSTM Networks – colah's blog. Retrieved February 19, 2023, from <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [6] Corpora collection. Download Corpora English. (n.d.). Retrieved February 18, 2023, from <https://wortschatz.uni-leipzig.de/en/download/English>
- [7] University Communications | University of Notre Dame. (1970, February 23). University of Notre Dame. Retrieved February 18, 2023, from <https://www.nd.edu/>