注：加\*的是我们未考虑的特征，黄色背景标注的是可以轻松实现的特征

1）长度相关

word\_count

sentence\_count

avg\_sentence\_len

\*每个单词的平均字符个数

spelling\_errors

long\_word

2) language model counts

unigrams\_count

bigrams\_count

trigrams\_count

3) POS counts

noun\_count

adj\_count

adv\_count

verb\_count

fw\_count

\*4）occurance features

\*逗号

\*引号

问号个数

正式引用和任务强相关（对给材料的作文）

直接引语和间接引语

转述动词等规则区分是否为引语

（对给材料的作文）核心概念仅给材料时使用

识别核心概念：对材料中高tf.idf的词汇或者短语

\*5）句法变化 Chen and He (2013)

句法变化与任务弱相关

不同的句法树的占比，平均树高

时间和原因状语从句的比例, 时间和原因状语从句的引导词

\*6）Style Features

A= {N, ADJ, PP, DET}, B= {PR, V,ADV, UH}, and n is the number of tokens in thetext

type-token-ratio Chen and He (2013)

用以估计作者词汇量的大小

average word frequency 与任务弱相关

作者词汇量的丰富程度与语料库中词汇的频率关系很大.作者使用的词汇在语料库中的频率越小,作者的语言熟练度越高

We model this idea by calculating the average wordfrequency in the Web1T-corpus (Brants and Franz,2006).

\*7) Cohesion Features

文章结构与任务弱相关

连接词的使用(like therefore or accordingly)表征了文章的一致性

用一个fixed list来计数连接词的使用数, 用文章的token总数来normalize

\*8）Coherence Features与任务弱相关

measuring the topical overlap between adjacent sentences. We use similarity measures based on n-gram overlap and redundancy (e.g. of nouns).

9）Error Features 与任务弱相关

语法和拼写错误是一篇烂文的明显标志, 但被发现在作文评分中影响并不大.

基于规则的简单语法错误检测

10）Readability Features与任务弱相关 可调用textstat!

非常简单或非常复杂的写作方式都是烂文的标志.

rely on normalized counts of words, letters, syllables or other phenomena(like abbreviations) which affect the readability.

high scoring essays might be associated with different levels of readability

2.9 Task-Similarity Features：Pass

\*11）Set-Dependent Features

n-grams features和 POS n-grams features

【挑选特征】

rf.feature\_importances\_.argsort()[::-1][:k]

【study】

利用NLTK进行分句分词

<http://blog.csdn.net/baidu_27438681/article/details/60468848>

拼写检查 enchant

<http://www.jb51.net/article/64895.htm>

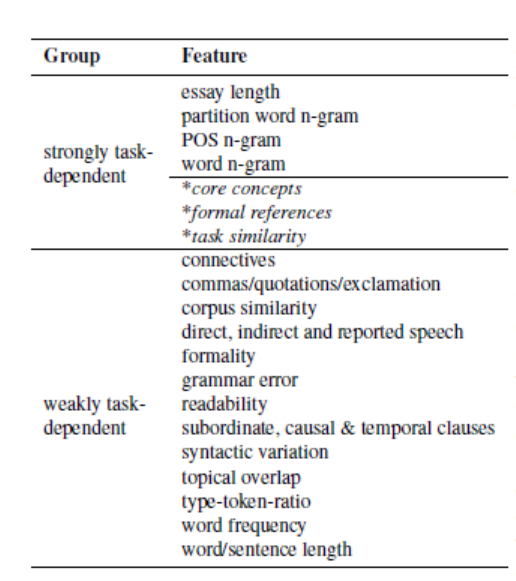
http://norvig.com/spell-correct.html

from textblob import Word

w = Word('falibility')

w.spellcheck()

confidence意思是纠错时可能有很多可能



Lasso > LogisticRegression

33%

|  |  |  |
| --- | --- | --- |
|  | 直接预测score | score = score1+score2 |
| LogisticRegression | 0.6458680291851369 | 0.6644192462674752 |
| Lasso | 0.8140479312552429 | 0.8082751969235769 |
| LassoCV | 0.817965416989607 | 0.8181317310494515 |
| XGBRegressor | 0.8192058190153856 | 0.8210317669184185 |
| RandomForestRegressor | 0.801746350347467 | 0.8031391977172094 |
| GradientBoostingRegressor | 0.8176579049616708 | 0.818029212009945 |
| SVR | 0.09704862677364856 | 0.09626108405035987 |

ensemble

|  |  |  |
| --- | --- | --- |
|  | 直接预测 | 单个预测再求和 |
| 不考虑LassoCV | 0.7869863006119999 | 0.802706583617689 |
| 不考虑XGBRegressor | 0.7912311870478275 | 0. 8030834876906545 |
| 不考虑RandomForestRegressor | 0.8024845184375717 | 0. 7906975745659727 |
| 不考虑GradientBoostingRegressor | 0.7883060868675381 | 0.7837678382867238 |
| All | 0.7935479290045 | 0.7997111385110282 |

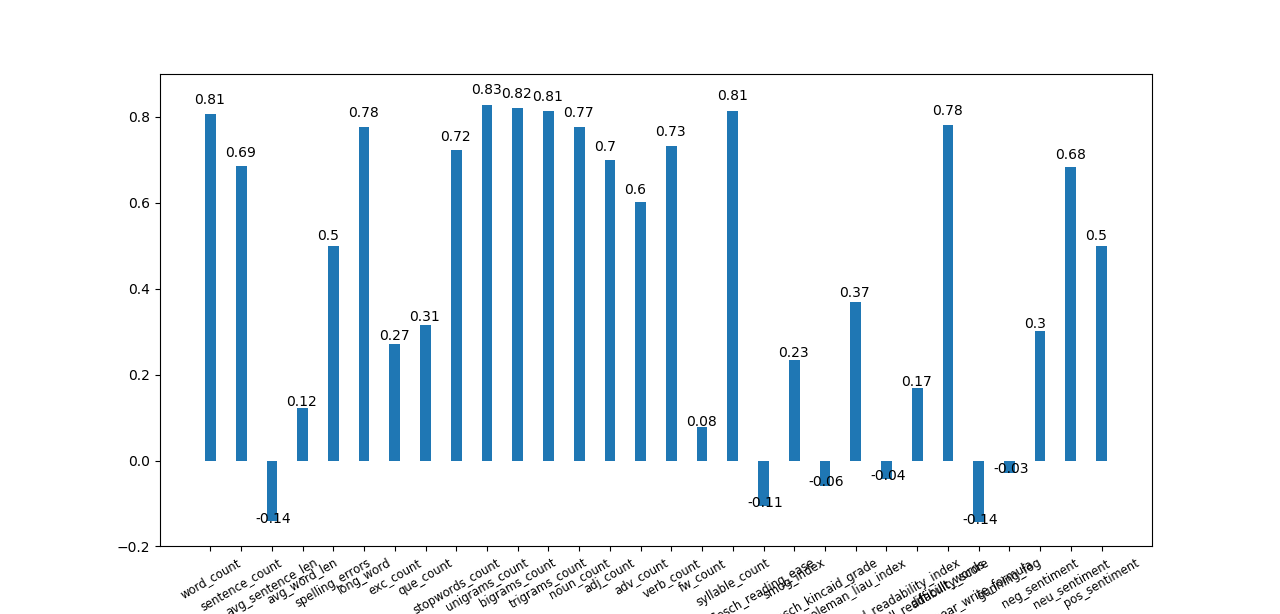
直接求平均

|  |  |  |
| --- | --- | --- |
|  | 直接预测 | 单个预测再求和 |
| 不考虑LassoCV | 0.816939711818324 | 0.8198827120838125 |
| 1.不考虑XGBRegressor | 0.8233919280859795 | 0.8271370514380181 |
| 2.不考虑RandomForestRegressor | 0.826332117630096 | 0.8269909809307967 |
| 不考虑GradientBoostingRegressor | 0.8262123401763879 | 0.8267080280592762 |
| 3.All | 0.8228908107038992 | 0.8267406708157818 |

<https://github.com/RDulepet19/AES>

https://github.com/SahilC/AutomaticEssayGrading

|  |  |
| --- | --- |
| 3 | 0.0071988821020094376 |

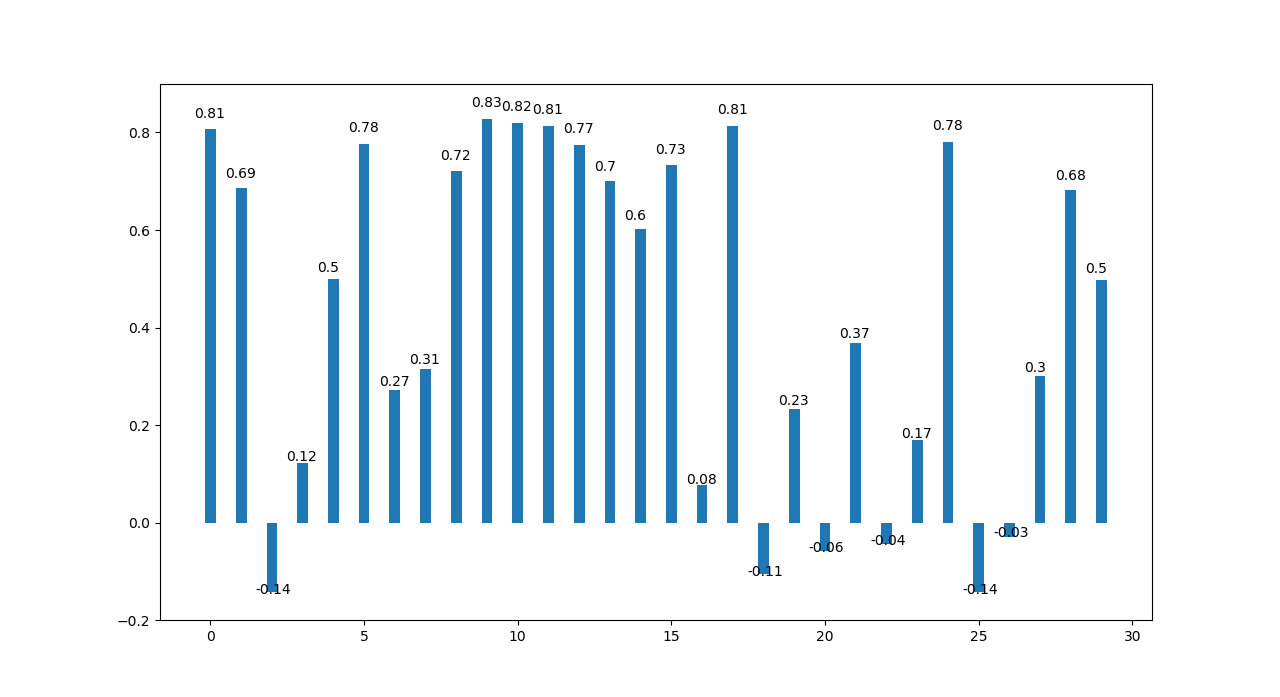


2 3

考虑所有特征是，线下Fit score1 + score2: The pearsonr of test set is 0.8378227751352587

【特征选择】

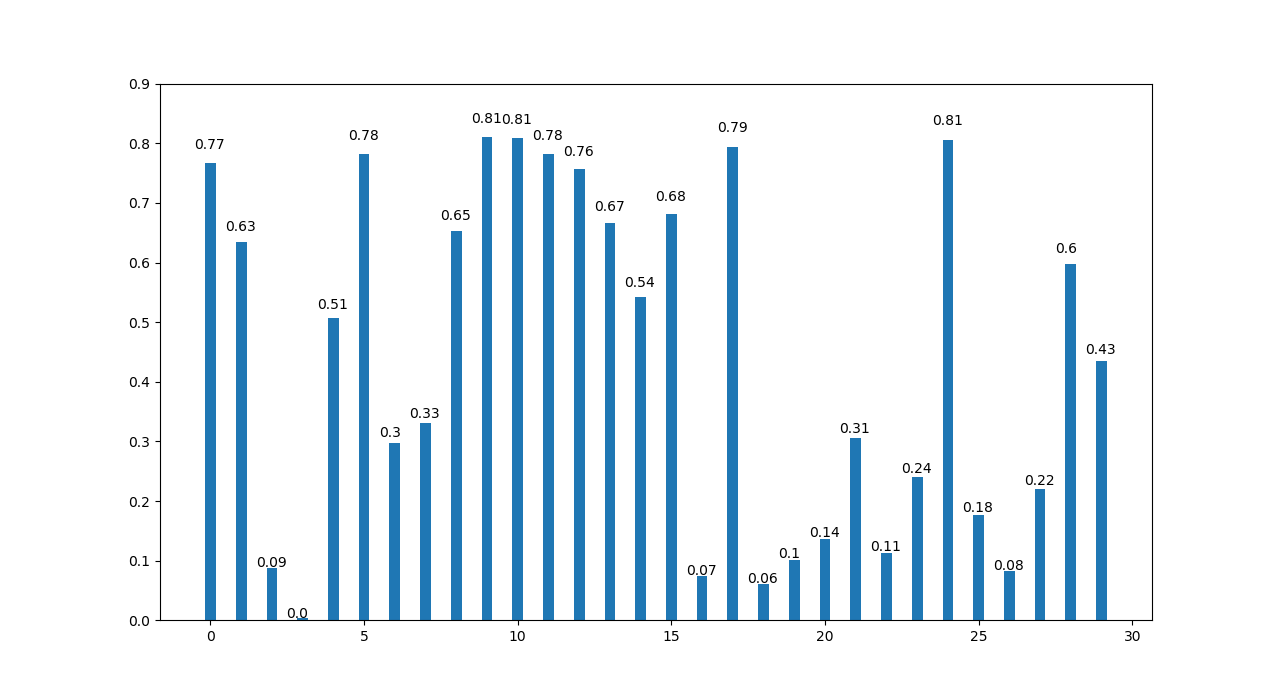
1. 特征与label的相关系数



相关系数小于0.15 —— 2,3,16,18,20,22,25,26

相关系数小于0.2 —— 2,3,16,18,20,21,22,23,25,26

1. 单个特征应用到模型时的皮尔逊系数



皮尔逊系数小于0.15 —— 2,3,16,18,19,20,22,26

皮尔逊系数小于0.2 —— 2,3,16,18,19,20,22,25,26

第一轮：

|  |  |
| --- | --- |
| 相关系数小于0.15 | 2,3,16,18,20,22,25,26 |
| 皮尔逊系数小于0.15 | 2,3,16,18,19,20,22,26 |

所以去除以下特征序号的特征：2,3,16,18,20,22,26

【New】

去除皮尔逊系数小于0.15，只保留23个特征，直接跑线下测是变差的。所以调参后提交，结果：0.865565

不好

30个特征+调过的参数 好~ 0.867077

这是数值：

num\_tokens 0.794060

num\_sentences 0.690165

average\_sent\_len -0.163936

average\_word\_len 0.341625

num\_commas 0.609351

num\_quotations 0.289667

num\_exclamation\_marks 0.271486

type\_token\_ratio 0.151769

avg\_word\_freq 0.252310

cohesion 0.372869

spelling\_error 0.428537

grammar\_error 0.274009

syllable\_count 0.813792

flesch\_reading\_ease -0.105345

flesch\_kincaid\_grade -0.058573

gunning\_fog -0.028699

smog\_index 0.234122

automated\_readability\_index -0.043910

coleman\_liau\_index 0.368479

linsear\_write\_formula -0.142551

dale\_chall\_readability\_score 0.169032

difficult\_words 0.780656

num\_long\_words 0.737200

num\_stopwords 0.729191

score 1.000000

33%训练集

|  |  |
| --- | --- |
|  | 单个预测再求和 |
| 不考虑LassoCV | 0.8396168486293869 |
| 不考虑XGBRegressor | 0.8393801034873545 |
| 不考虑RandomForestRegressor | 0.8406300463534263 |
| 不考虑GradientBoostingRegressor | 0.8397572022619177 |
| All | 0.8409196120409997 |

加了稀疏特征 0.867424

特征归一化0.867450

std = StandardScaler()

train\_std = std.fit\_transform(train\_no\_vec)

test\_std = std.transform(test\_no\_vec)

<https://help.github.com/articles/basic-writing-and-formatting-syntax/>

<https://github.com/SahilC/AutomaticEssayGrading/tree/master/src>

<https://github.com/adamcsvarga/essay_scoring>

https://github.com/RDulepet19/AES

<http://www.cs.cmu.edu/~norii/pub/aes.pdf>

<http://cs229.stanford.edu/proj2012/MahanaJohnsApte-AutomatedEssayGradingUsingMachineLearning.pdf>

http://www.aclweb.org/anthology/D/D13/D13-1180.pdf