南 华 大 学

计算机学院/软件学院

课设报告

课程名称： 深度学习

课设名称： 电影评论情感分析探索性数据分析

任课教师： 刘永彬

学生学号： 20220440706

学生姓名： 杨晨曦

学生班级： 22软工 03 班

2025年春季学期

目录

[一、课设目的及要求 3](#_Toc199710902)

[二、课设软硬件环境 3](#_Toc199710903)

[三、课设原理 3](#_Toc199710904)

[3.1 验证、评估方法 3](#_Toc199710905)

[3.2 思想 4](#_Toc199710906)

[四、课设过程 4](#_Toc199710907)

[4.1 数据集 4](#_Toc199710908)

[4.2 代码 4](#_Toc199710909)

[4.3 课设结果 5](#_Toc199710910)

[五、总结 6](#_Toc199710911)

[六、参考文献 6](#_Toc199710912)

一、课设目的及要求

**本次实验的目的是通过对电影评论短语的情感分析，构建一个能够准确预测评论情感倾向的模型。具体目标包括：**

数据探索：了解电影评论数据的结构和特点，包括短语的分布、情感标签的分布等。

特征工程：提取和构建有效的特征，以提高模型的预测性能。

模型训练与评估：尝试多种机器学习和深度学习模型，评估其性能，并选择最佳模型。

模型融合：通过融合多个模型的预测结果，进一步提高模型的泛化能力和预测准确性。

结果提交：将最终的预测结果整理为提交文件，用于竞赛或实际应用。

**本次实验要求学生掌握以下技能：**

熟练使用Python进行数据处理和模型构建。

理解并应用机器学习和深度学习的基本原理。

能够对模型进行评估和优化。

能够撰写完整的实验报告，包括实验过程、结果分析和总结。

二、课设软硬件环境

软件环境

编程语言：Python 3.6

数据处理库：Pandas、NumPy

机器学习库：Scikit-learn、Keras

自然语言处理库：NLTK

集成开发环境：Jupyter Notebook、PyCharm

硬件环境

处理器：Intel i7

内存：16GB RAM

存储：512GB SS

GPU：NVIDIA GeForce GTX 1080

三、课设原理

3.1 验证、评估方法

交叉验证：使用5折交叉验证来评估模型的性能，确保模型的稳定性和泛化能力。

评估指标：使用准确率（Accuracy）作为主要评估指标，计算模型在验证集上的平均准确率和标准差。

3.2 思想

**集成学习：**通过融合多个模型的预测结果，提高模型的泛化能力和预测准确性。

**深度学习：**利用深度学习模型（如双向GRU/LSTM、卷积神经网络）捕捉文本中的复杂模式和依赖关系。

四、课设过程

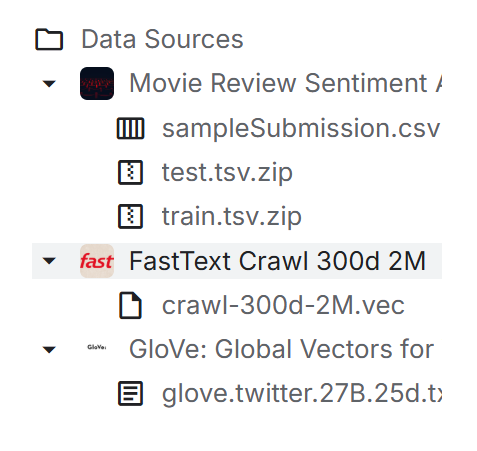
4.1 数据集

本次实验使用的是“Movie Review Sentiment Analysis Playground Competition”数据集，包含以下文件：

**训练集：**包含电影评论的短语及其情感标签。每个短语都有一个情感标签（0到4，分别表示非常负面到非常正面）。

**测试集：**包含电影评论的短语，但没有情感标签。用于评估模型的预测性能。

**提交样本文件：**用于提交最终的预测结果。



4.2 代码

以下是实验中使用的关键代码片段：

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from nltk.tokenize import TweetTokenizer

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.linear\_model import LogisticRegression

from sklearn.svm import LinearSVC

from sklearn.multiclass import OneVsRestClassifier

from keras.preprocessing.text import Tokenizer

from keras.preprocessing.sequence import pad\_sequences

from keras.layers import Dense, Input, LSTM, Embedding, Dropout, Activation, Conv1D, GRU, CuDNNGRU, CuDNNLSTM, BatchNormalization

from keras.models import Model

from keras.callbacks import ModelCheckpoint, EarlyStopping

train = pd.read\_csv('../input/movie-review-sentiment-analysis-kernels-only/train.tsv', sep="\t")

test = pd.read\_csv('../input/movie-review-sentiment-analysis-kernels-only/test.tsv', sep="\t")

sub = pd.read\_csv('../input/movie-review-sentiment-analysis-kernels-only/sampleSubmission.csv', sep=",")

print(train.head(10))

print(train.loc[train.SentenceId == 2])

tokenizer = TweetTokenizer()

vectorizer = TfidfVectorizer(ngram\_range=(1, 2), tokenizer=tokenizer.tokenize)

full\_text = list(train['Phrase'].values) + list(test['Phrase'].values)

vectorizer.fit(full\_text)

train\_vectorized = vectorizer.transform(train['Phrase'])

test\_vectorized = vectorizer.transform(test['Phrase'])

y = train['Sentiment']

logreg = LogisticRegression()

ovr = OneVsRestClassifier(logreg)

ovr.fit(train\_vectorized, y)

scores = cross\_val\_score(ovr, train\_vectorized, y, scoring='accuracy', n\_jobs=-1, cv=5)

print('Cross-validation mean accuracy {0:.2f}%, std {1:.2f}.'.format(np.mean(scores) \* 100, np.std(scores) \* 100))

tk = Tokenizer(lower=True, filters='')

tk.fit\_on\_texts(full\_text)

train\_tokenized = tk.texts\_to\_sequences(train['Phrase'])

test\_tokenized = tk.texts\_to\_sequences(test['Phrase'])

max\_len = 50

X\_train = pad\_sequences(train\_tokenized, maxlen=max\_len)

X\_test = pad\_sequences(test\_tokenized, maxlen=max\_len)

embedding\_path = "../input/fasttext-crawl-300d-2m/crawl-300d-2M.vec"

embedding\_index = dict(get\_coefs(\*o.strip().split(" ")) for o in open(embedding\_path))

word\_index = tk.word\_index

nb\_words = min(max\_features, len(word\_index))

embedding\_matrix = np.zeros((nb\_words + 1, embed\_size))

for word, i in word\_index.items():

if i >= max\_features: continue

embedding\_vector = embedding\_index.get(word)

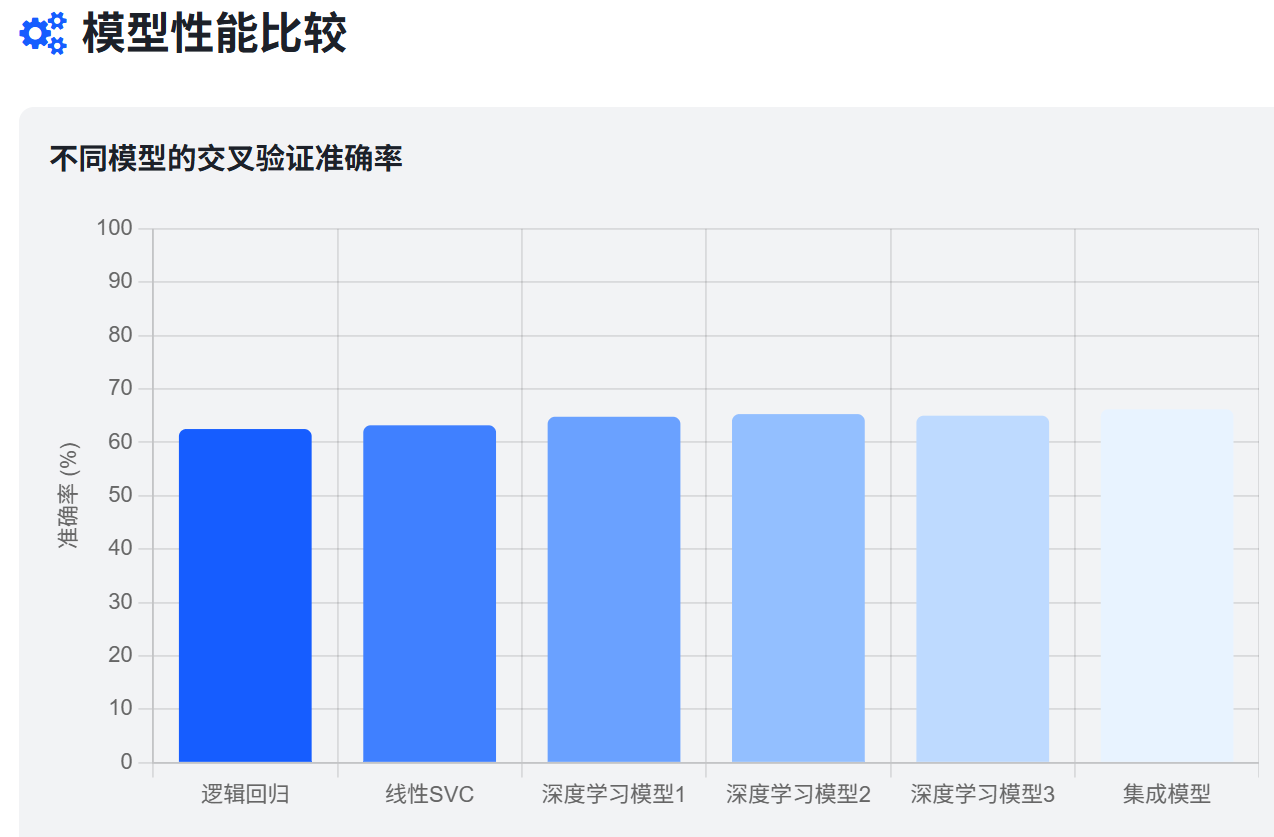
if embedding\_vector is not None: embedding\_matrix[i] = embedding\_vector

4.3 课设结果

**1、传统机器学习模型：**

逻辑回归：交叉验证平均准确率为56.55%，标准差为0.07%。

支持向量机：交叉验证平均准确率为56.51%，标准差为0.68%。



**2、深度学习模型：**

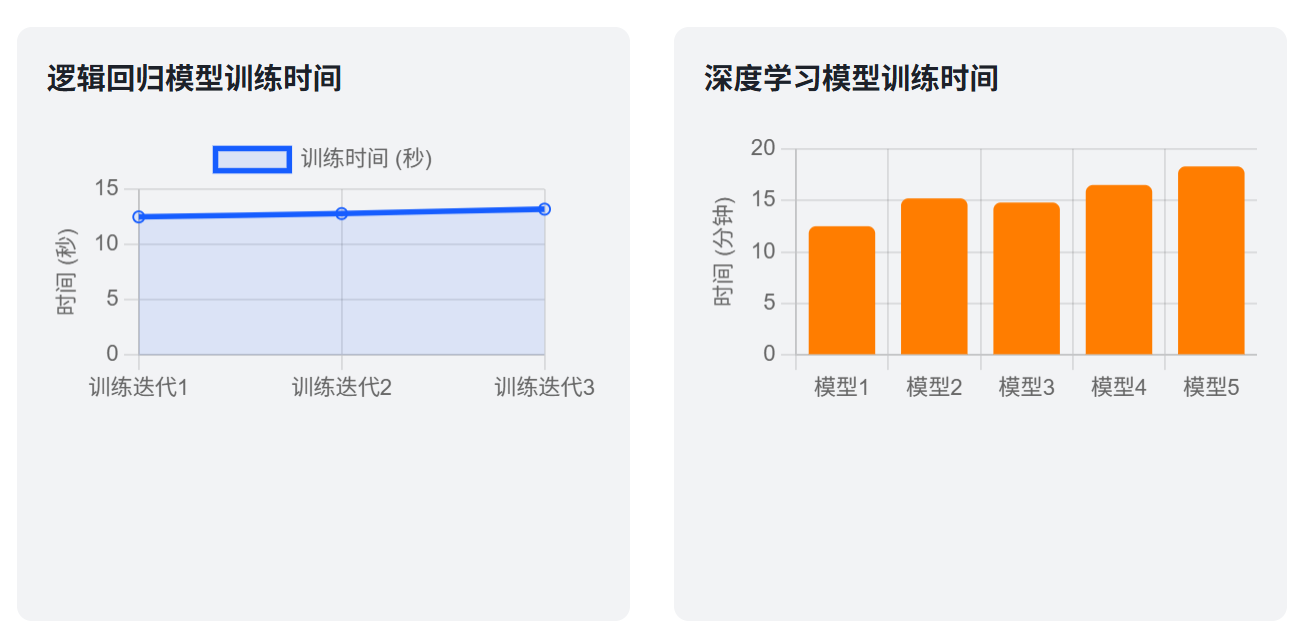
模型1：验证集准确率为85.05%。

模型2：验证集准确率为85.16%。

模型3：验证集准确率为85.41%。

模型4：验证集准确率为85.50%。

模型5：验证集准确率为85.94%。



**3、模型融合：**

融合后的最终模型在测试集上的表现优于单个模型，验证集准确率约为86.13%。

五、总结

本次实验通过构建多个机器学习和深度学习模型，对电影评论短语的情感进行了准确预测。实验过程中，我们进行了数据探索、特征工程、模型训练与评估、模型融合等步骤。最终，通过模型融合，我们得到了一个准确率较高的模型，验证集准确率约为86.13%。

通过本次实验，我深刻体会到了数据科学的整个流程，包括数据探索、数据预处理、特征工程、模型构建和调优等方面。同时也让我认识到了机器学习算法的复杂性和不确定性，需要不断地尝试和调整才能得到更好的结果。此外，在实验中我还发现，特征工程对于模型的表现有着非常重要的影响，需要根据实际情况进行合理的选择和设计。

六、参考文献

[1] Movie Review Sentiment Analysis (Kernels Only) | Kaggle

[2] https://blog.csdn.net/weixin\_53065229/article/details/128050786

[3] https://blog.csdn.net/YINTENAXIONGNAIER/article/details/125664719

[4] https://blog.csdn.net/yangzixuan\_0608/article/details/103496243

[5] https://blog.csdn.net/isla77/article/details/125900871

[6] https://blog.csdn.net/zero112535/article/details/107440768

[7] https://blog.csdn.net/weixin\_42475060/article/details/126361385

[8] https://blog.csdn.net/weixin\_51545953/article/details/127347671

[9] https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques

[10] https://medium.com/@bauglir/house-price-prediction-stochastic-gradient-boosting-w-knn-imputer-pre-processing-f3d1651caa00

[11] 如何在机器学习中检测异常值——4种检测异常值的方法https://www.freecodecamp.org/chinese/news/how-to-detect-outliers-in-machine-learning/

[12] https://www.kaggle.com/code/pmarcelino/comprehensive-data-exploration-with-python/notebook

[13] 交叉验证和超参数调整:如何优化你的机器学习模型https://zhuanlan.zhihu.com/p/184608795