**实验背景**

本次实验的目的为对现有的传统机器学习方法与深度学习方法在文本分类领域进行测试，并观察两者之间在文本分类任务上的表现差异。除外，本次实验也会针对模型在不同参数下的表现，尝试总结出参数对模型表现的影响规律。

**数据集介绍**

本次实验所使用的数据集为IMDB数据集。IMDB 数据集是一个经典的自然语言处理数据集，广泛应用于情感分析研究领域。该数据集来源于互联网电影数据库（IMDB），包含了从中提取的电影评论，主要用于分析评论者的情感倾向。其应用场景覆盖文本分类、情感分析以及自然语言处理模型的性能评估等任务。

数据集中总共包含50,000条影评，这些评论被平均分为训练集和测试集，每部分各有25,000条。评论被标注为两类情感标签，即正面情感（positive）和负面情感（negative）。数据集的设计确保了类别平衡性，正负评论数量各占50%，为分类模型的训练和测试提供了稳定的样本分布。

除此之外，IMDB数据集也有个显著的特点，那便是其评论文本的长度长短不一。因此对模型的训练存在一定挑战。而本次实验便是以影评为训练文本，并预测其情感标签。而由于数据量巨大可能会导致训练耗时，因此本文仅采用5000条数据作为训练集，1000条数据作为测试集。

**机器学习算法简介**

为了避免单一机器学习模型造成的对比偏见，因此本文应用了四种主流的分类机器学习模型来和深度学习模型进行对比。以下为本次实验所用模型。

1. **Naive Bayes（朴素贝叶斯）**

朴素贝叶斯是一种基于贝叶斯定理的简单而强大的概率分类算法，广泛用于文本分类、垃圾邮件检测等任务。它假设特征之间是条件独立的，这种“朴素”的独立性假设使得算法在实际应用中计算高效且鲁棒。尽管独立性假设不总是成立，朴素贝叶斯在高维数据上表现良好，尤其是文本分类任务中，其简单性和较低的计算成本使其成为许多应用中的首选。

1. **Linear SVM（支持向量机）**

支持向量机是一种线性分类算法，旨在通过寻找超平面将数据集的不同类别最大程度地分开。它以最大化分类边界（即支持向量到超平面的最小距离）为目标，从而提高模型对新样本的泛化能力。支持向量机对于线性可分的数据非常有效，且通过核函数扩展，可以处理非线性数据。

1. **Logistic Regression（逻辑回归）**

逻辑回归是一种回归分析技术，用于建模因变量（目标变量）和一个或多个自变量（特征）之间的线性关系。它的基本形式假设目标变量是所有输入变量的加权线性组合，并通过最小化误差平方和（OLS）来拟合模型。在简单逻辑回归中，只有一个自变量；在多元线性回归中，可以处理多个输入变量。线性回归的优点在于模型简单且易于解释，适用于对数据中变量关系的研究。尽管在面对非线性数据时表现有限，但它作为一种基线模型，仍在经济学、统计学和机器学习任务中广泛使用。

1. **Random Forest（随机森林)**

随机森林是一种集成学习算法，通过构建多个决策树并利用多数投票（分类任务）或平均值（回归任务）来做出最终预测。其核心思想是通过引入随机性（如随机选择特征和数据子集）来构建多样化的决策树，降低单棵树过拟合的风险。随机森林具有较高的准确性和鲁棒性，能够处理大规模数据和高维特征，且对缺失值和非线性数据有良好的处理能力。

**深度学习算法简介**

本次实验所采用的深度学习模型为Facebook（现为Meta）AI团队在2019年基于BERT模型优化后提出的一种改进版本---- RoBERTa，全称为Robustly Optimized BERT Approach。RoBERTa 的设计目标是通过对预训练过程的调整和优化，进一步提升模型在自然语言理解任务中的性能。

相比于 BERT，RoBERTa 在预训练过程中使用了更大规模的训练数据，总计约 160GB，是 BERT 数据量的十倍。训练数据来源包括 BookCorpus、English Wikipedia、CommonCrawl News、OpenWebText 和 Stories，这些语料库为模型提供了多样且丰富的语言上下文。在训练策略上，RoBERTa 移除了 BERT 中的下一句预测（NSP）任务，专注于掩码语言模型（Masked Language Model, MLM）的训练。研究表明，NSP 任务对模型性能的提升并无显著作用，而通过移除 NSP，可以更专注于语言模型核心能力的提升。

在文本表示方法上，RoBERTa 采用了字节级的 Byte-Pair Encoding (BPE) 分词方法。相比于传统的分词方法，这种方法能更有效地处理未登录词和罕见词，提高了词汇的覆盖率。通过这一改进，RoBERTa 在处理复杂语言模式时具有更强的表现力。

**实验方法**

**数据预处理**

本次实验通过tensorflow的接口加载数据集后，由于低频单词往往对文本分类贡献较小，且还会增加模型的复杂度，于是将词汇表的大小限制为10000，同时也减少分词数据的稀疏性和模型过拟合的风险。

在传统机器学习方法中，一般会需要获取TF-IDF（Term Frequency-Inverse Document Frequency）特征，这是为了衡量一个词在文档中的重要性，以此让机器学习模型能够更好的理解和处理文本数据。因此，本次实验也通过获取文本数据的索引并进行存储。

在深度学习方法中，文本数据需要固定的长度输入，因此需要进行分词、编码、填充等步骤，以此确保数据输入时的长度一样。编码的目的是因为模型不能直接理解自然语言，因此需要将模型进行编码形成模型能够理解的形式。确保数据的长度相同是由于模型的输入需要固定形状的tensor，因为模型在并行计算过程中如矩阵乘法都依赖输入维度的一致性。

**实验过程**

本次实验针对不同的机器学习模型和深度学习模型采用了不同参数进行对比，以下为各个模型的参数。

|  |  |  |  |
| --- | --- | --- | --- |
| **模型类型** | **参数名称** | **说明** | **默认值/范围** |
| **机器学习 (ML)** | max\_features | 最大考虑的特征数量 | [500, 1000, 2000, 3000] |
|  | train\_size | 训练数据大小 | [5000, 10000,40000] |
|  | test\_size | 测试数据大小 | [1000. 2000, 10000] |
| **深度学习 (DL)** | batch\_size | 批处理大小 | [16, 32, 64] |
|  | max\_length | 文本的最大长度 | [64, 128, 256] |
|  | learning\_rate | 学习率 | [1e-5, 2e-5, 3e-5] |
|  | train\_size | 训练数据大小 | [5000, 10000,40000] |
|  | test\_size | 测试数据大小 | [1000. 2000, 10000] |

调整各个参数的目的是为了观察模型在不同参数下对数据的适应性如何，以下为调整各个参数的意义：

|  |  |  |
| --- | --- | --- |
| 参数名称 | 模型类型 | 意义 |
| max\_features | 机器学习 (ML) | 最大考虑的特征数量，决定使用哪些重要特征。 |
| train\_size | ML / DL | 训练数据的数量，决定模型学习能力。 |
| test\_size | ML / DL | 测试数据的数量，决定评估精度。 |
| batch\_size | 深度学习 (DL) | 每次训练用于更新模型的样本数量。 |
| max\_length | 深度学习 (DL) | 输入文本的最大长度，截断或填充以满足长度要求。 |
| learning\_rate | 深度学习 (DL) | 模型参数更新的步长大小。 |

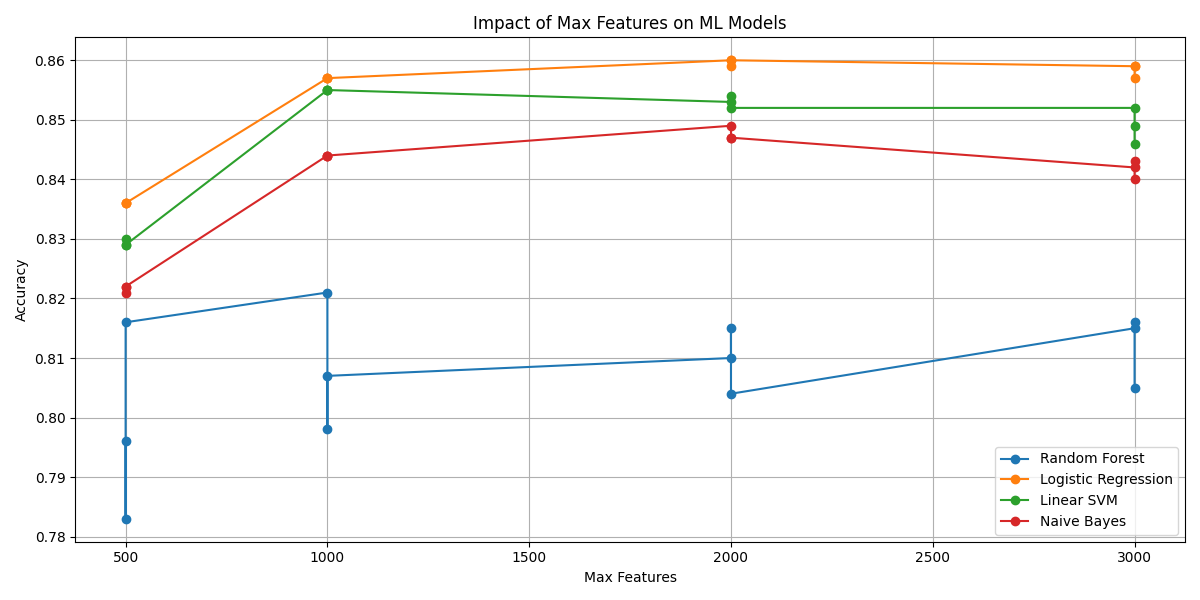
模型训练后，将模型的各个参数结果记录起来并以绘图形式进行比对，最终得出机器学习模型或是深度学习模型在文本分类上的效果好坏。

**实验结果**

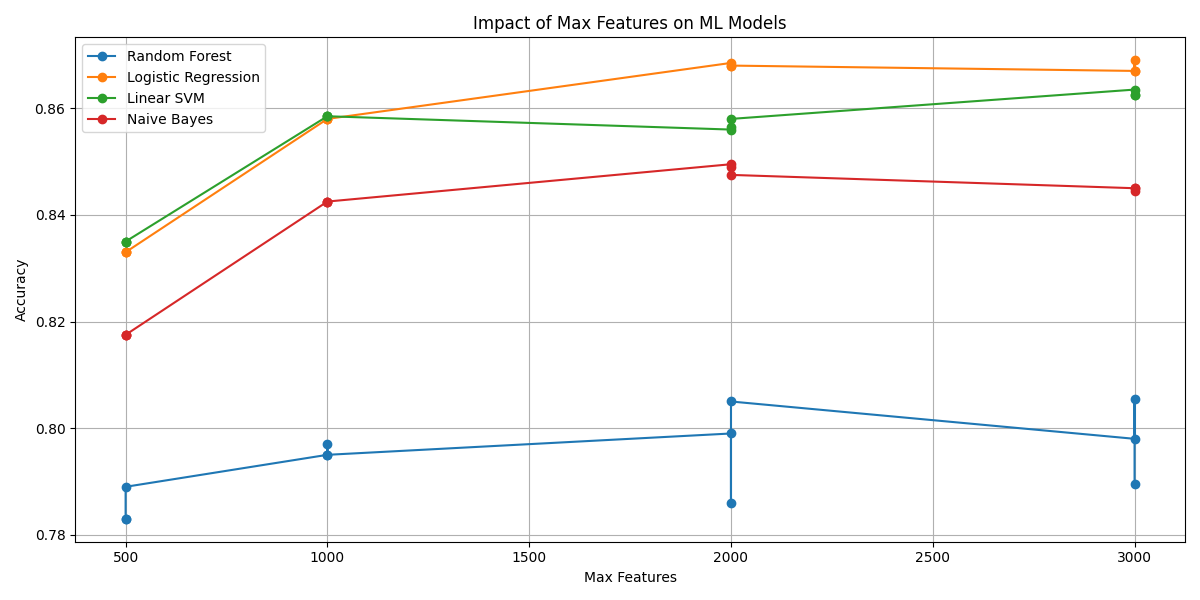
**机器模型间各个参数对比**

**Max Features**

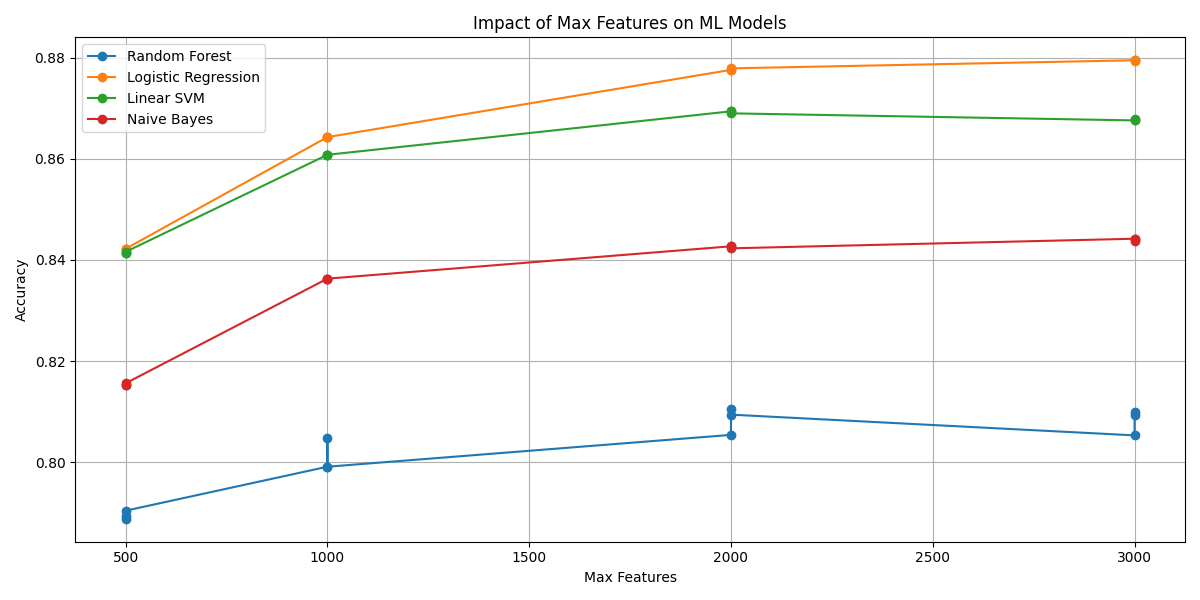
5000条数据训练集， 1000条数据测试集



10000条数据训练集， 2000条数据测试集



40000条数据训练集， 10000条数据测试集



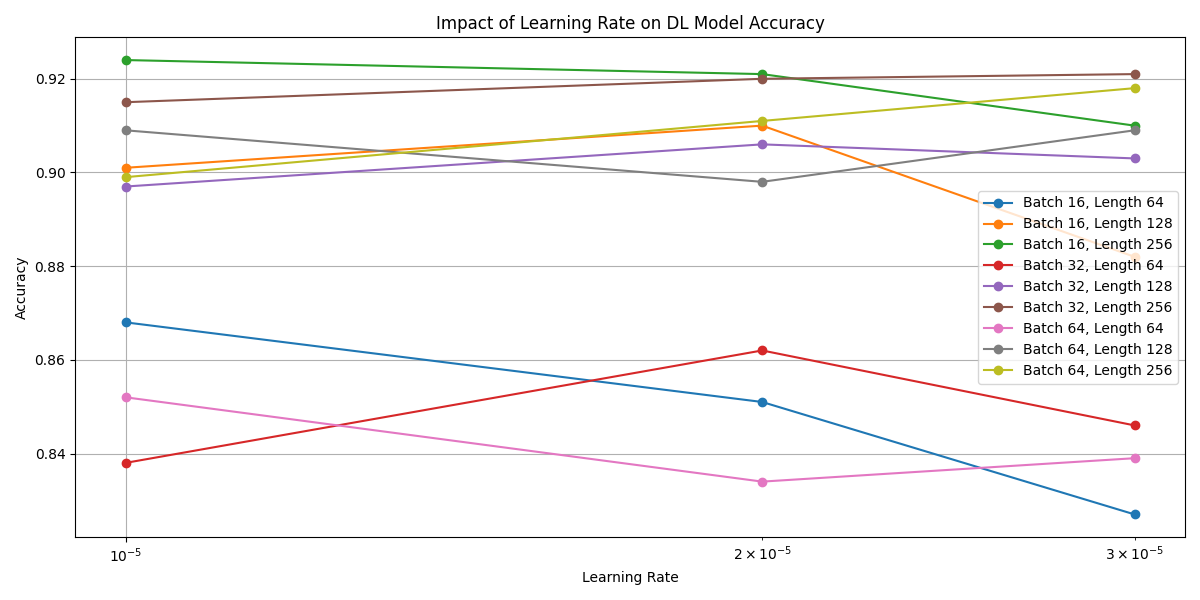
特征数的增加在初期（500 到 2000）对所有模型都有显著提升作用。在超过2000后模型性能趋于平缓甚至有所下滑，尤其是Random Forest和Naïve Bayes。因此可以判断的是，保留过多的词汇可能会造成过拟合的现象，2000个特征数是本次实验中最适合的参数。

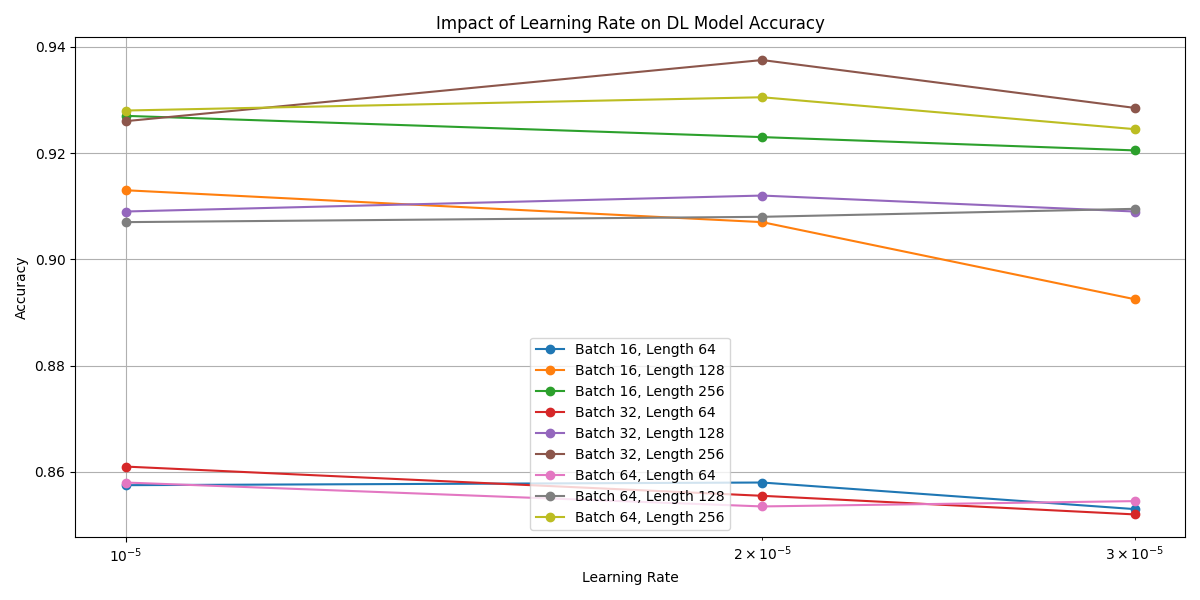
当中不管数据集的变化如何，Logistic Regression和SVM都是效果最好的模型。而在数据集不断增大的时候，这两个模型表现也都持续增长，反之Random Forest和Naïve Bayes的表现停滞甚至下滑。

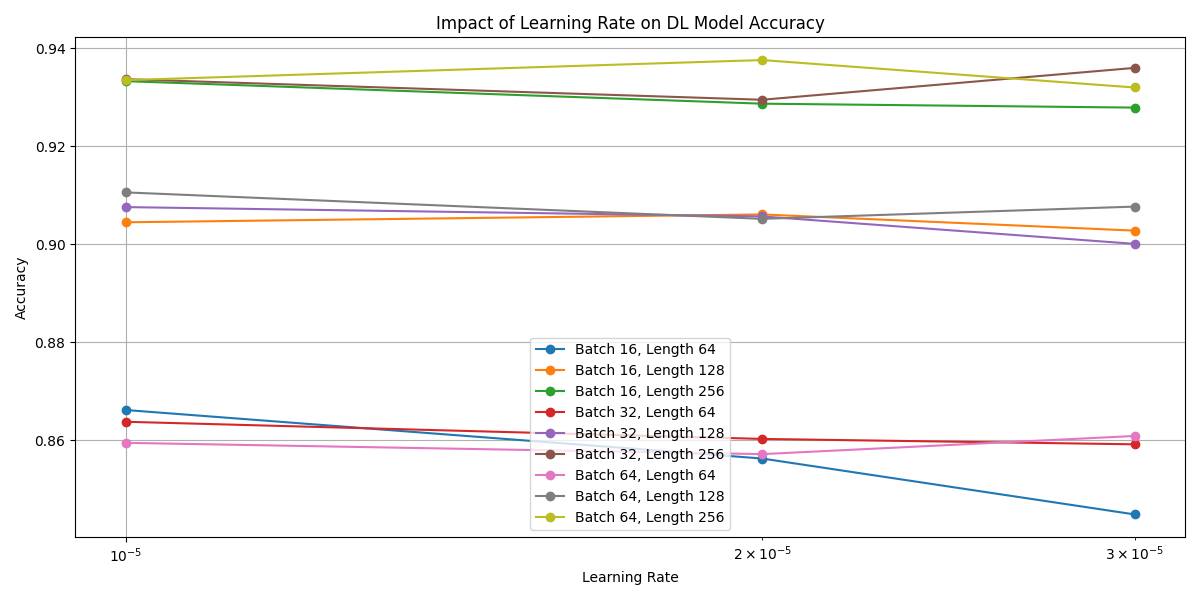
**深度学习间各个参数对比**

**Learning-rate, Batch Size, Max Length间对比**

5000条数据训练集，1000条数据测试集



10000条数据训练集，2000条数据测试集****

40000条数据训练集，10000条数据测试集****

从上图能够发现，在数据集大小递增的情况下，batch\_size的调整对性能影响是越来越小的。而max\_length在不同数据集的大小变化下，都依然是影响准确率的关键，因此可以推断文本在错误的地方阶段会很大影响语义的表达。

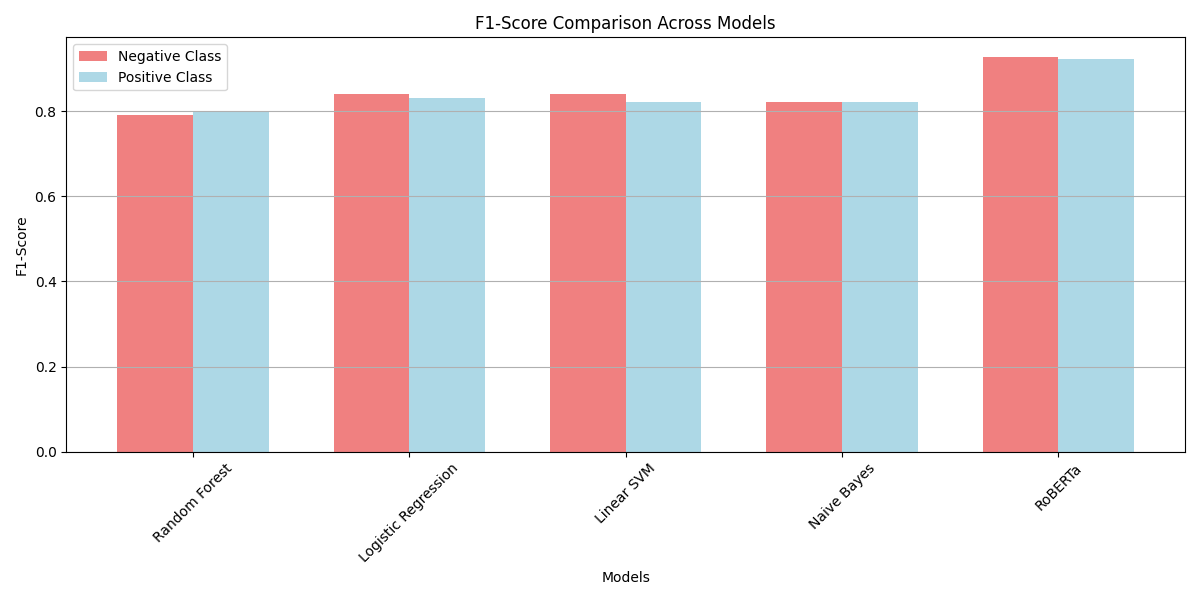
在学习率的选择上，咦2\*10^-5效果为最佳。再大一些的3\*10^-5学习率明显让大多数模型的表现都下降了，但是在小规模数据集上，较低学习率和较小的batch\_size能够带来更大的正向影响，适合在有算力限制的情况下进行训练。

但是在数据集的增长情况下，10000条训练集和40000条训练集的最佳模型表现差距不大，都在0.92-0.94的区间，因此能判断10000条数据集、2\*10^-5的学习率、32 batch\_size、256 max\_length会是该任务的最佳解答。

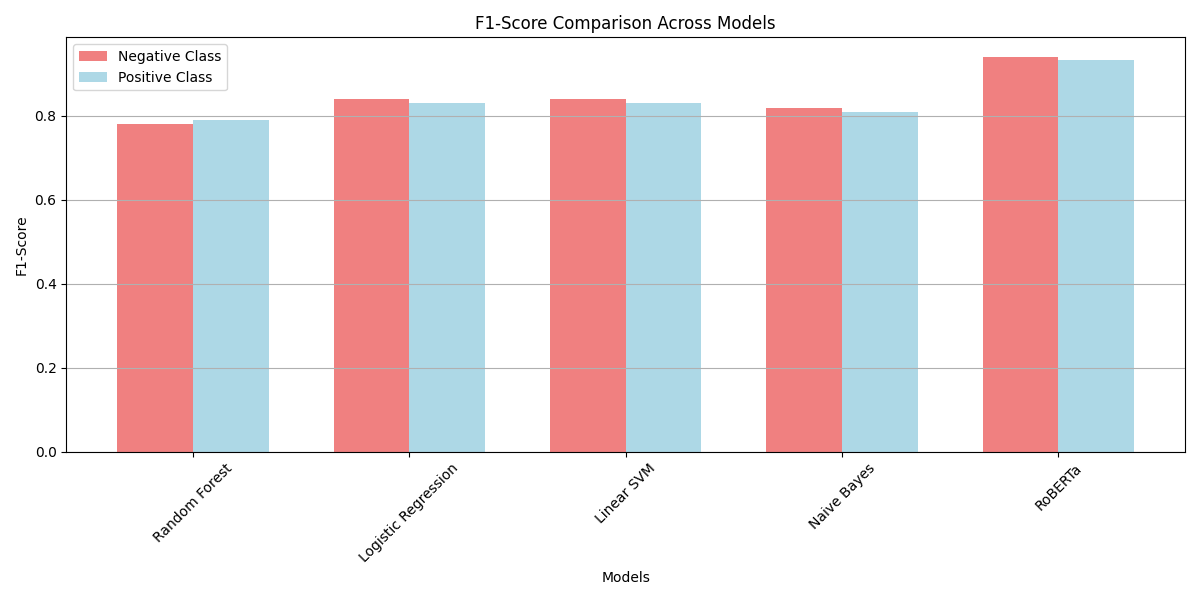
**深度学习与机器学习模型对比**

以下为将所有模型的最佳表现进行F1-Score上的对比，以此观察各个模型的表现差异。

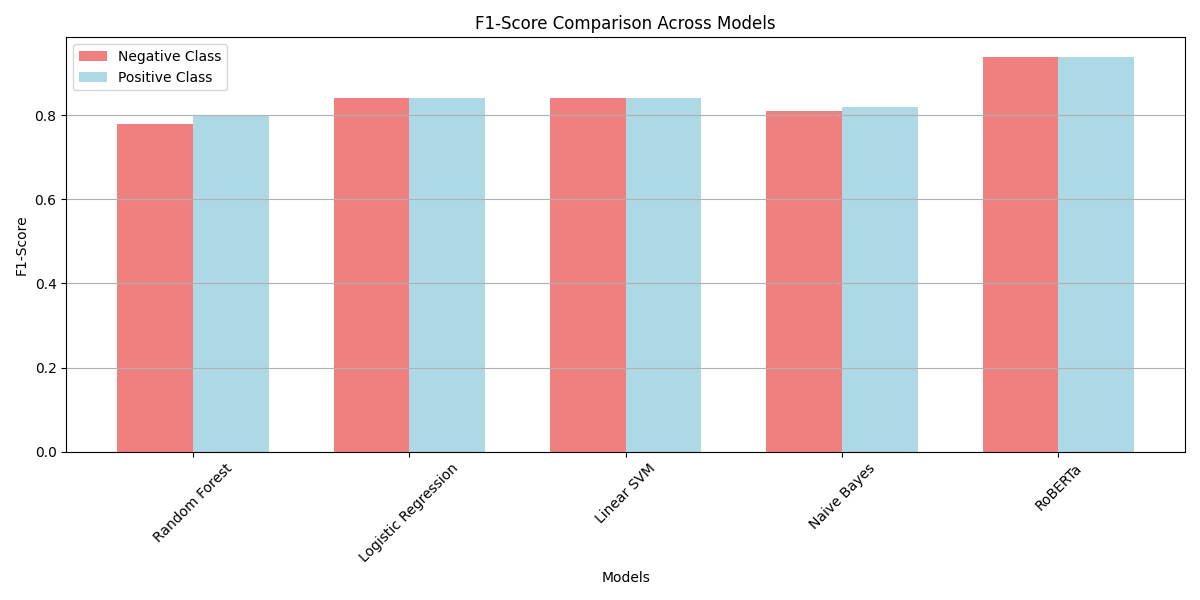
5000条数据训练集，1000条数据测试集



10000条数据训练集，2000条数据测试集



40000条数据训练集，10000条数据测试集



深度学习模型 RoBERTa 对正负类别的 F1-Score 明显在所有规模的数据集表现均高于所有传统机器学习模型。Logistic Regression 和 Linear SVM 的 F1-Score 次之，表现较为接近。Random Forest 和 Naive Bayes 的 F1-Score 最低，尤其对负类别的预测较差。可以推测Random Forest和Naïve Bayes的表现会更差的原因很大程度上是因为受负类别的表现所有影响。

在数据规模不断增大的情况，深度学习模型的F1-Score也肉眼可见的在增长，进一步拉开与机器学习模型的差距，几乎块到达0.90的区间。因此能够判断深度学习模型能依靠数据集规模的增长来不断提升预测表现。机器学习模型增长率则较小，甚至有停滞和倒退的情况。

整体而言深度学习模型的表现完全优于传统机器学习模型，但是两者之间的资源耗费也有着巨大的差距。深度学习模型需要大量的显卡算力来进行训练，耗费的时间也非常的多，相比机器模型只需要几秒钟就能学习好，两者之间的资源耗费区别可见一斑。因此在实际应用上依然需要考量性价比上的区别。

**总结**

本次实验对比了传统机器学习方法（Naive Bayes、Logistic Regression、Linear SVM、Random Forest）和深度学习方法（RoBERTa）在文本分类任务上的表现。通过使用不同规模的数据集（5000、10000、40000 条训练数据）和调整多项参数（如特征数量、学习率、文本长度等），评估了模型的准确率和 F1-Score。

实验结果表明，深度学习模型 RoBERTa 在所有数据集规模和评估指标（准确率、F1-Score）上均优于传统机器学习模型，尤其在数据规模增大的情况下，性能提升显著。相比之下，Logistic Regression 和 Linear SVM 在小数据集下表现接近 RoBERTa，但随着数据量增加，与深度学习模型的差距逐渐拉大。Random Forest 和 Naive Bayes 表现相对较差，尤其在负类别的预测上有明显不足。

参数调整的结果显示，机器学习模型的性能在特征数量超过一定阈值（如 2000 个）后趋于平稳甚至下降，而深度学习模型对文本长度（max\_length）和学习率（learning\_rate）更为敏感。综合分析表明，RoBERTa 适合大规模数据场景，并能够通过丰富的特征表示和上下文理解优势显著提升性能。

总体而言，深度学习模型在准确率和分类能力上有压倒性优势，但其训练耗时和资源需求远高于传统机器学习模型。实际应用中，需要结合任务规模和计算资源综合权衡模型选择。