

课程信息（由任课老师填写后发布给学生）：

学年学期： 2021至2022学年第1学期1

课程名称： 大数据管理与应用 1

课程代码： 0630126 1

任课教师： 海沫 1

学生信息(由学生填写):

姓 名： 秦旭昆 1

学 号： 2018312284 1

班 级： 信息管理与信息系统18 1

评分区域（由阅卷老师填写）：

总 分： 1

评 分 人： 1

真假新闻特征提取与分类—基于spark ML

学生姓名： 秦旭昆

学 号： 2018312284

学 院： 信息学院

专 业：信息管理与信息系统

指导教师： 海沫

论文成绩：

**内 容 摘 要**

随着互联网行业的发展，人们对存储的需求越来越大，集中式的存储渐渐不能满足大规模存储应用的需要。受益于服务器技术的发展和成熟，分布式存储开始被广泛的应用起来。

针对传统的大数据处理框架Hadoop在执行计算任务时抽象层次低、本文使用Spark作为计算引擎。引用Spark平台的ML机器学习库，使用Word2Vec处理文本数据并构建词向量，采用逻辑回归、决策树、随机森林、支持向量机和多层感知机作为分类器，对新闻数据进行处理。通过查准率、查全率等指标判断模型的优劣。最终结果表明，多层感知机在此分类问题上效果较好。

**关键词**：Spark 机器学习 自然语言处理 多层感知机

**ABSTRACT**

With the development of the Internet industry, people's demand for storage is increasing. Centralized storage can not meet the needs of large-scale storage applications. Benefiting from the development and maturity of server technology, distributed storage has been widely used.

In view of the low abstraction level of Hadoop, the traditional big data processing framework, when executing computing tasks, this paper uses spark as the computing engine. Using the ML machine learning library of spark platform, word2vec is used to process text data and construct word vectors. Logistic regression, decision tree, random forest, support vector machine and multi-layer perceptron are used as classifiers to process news data. Judge the advantages and disadvantages of the model through the indexes such as precision and recall. The final results show that the multi-layer perceptron is effective in this classification problem.

**KEY WORDS：**Spark Machine Learning NLP MLP

**目 录**

**[一、大数据平台 1](#_Toc29068)**

[（一）Hadoop 1](#_Toc2176)

[（二）Spark 2](#_Toc28816)

[（三）Hadoop与Spark的结合使用 2](#_Toc20999)

**[二、大数据与真假新闻定义 3](#_Toc16220)**

**[三、模型理论介绍 3](#_Toc26147)**

[（一）Spark ML库简介 3](#_Toc31145)

[（二）机器学习理论 4](#_Toc27384)

**[四、模型设计与训练 12](#_Toc2327)**

[（一）环境配置 12](#_Toc14217)

[（二）数据介绍 12](#_Toc13557)

[（三）指标分析 13](#_Toc30936)

[（四）特征构建 13](#_Toc13673)

[（五）模型选择 15](#_Toc28921)

**[五、结果分析与不足 15](#_Toc3249)**

**[六、总结与展望 16](#_Toc10178)**

[（一）实验总结 16](#_Toc16275)

[（二）实验不足 16](#_Toc25090)

[（三）展望 16](#_Toc3645)

**[参考文献 17](#_Toc7012)**

**真假新闻特征提取与分类——基于spark ML**

信息科技的发展引发数据量的爆炸式增长，导致传统数据存储与处理方式已经无法应用于大数据，分布式存储与处理技术应运而生。本文通过hadoop集群与spark平台处理大样本的新闻数据，构建真假新闻的分类器，帮助判别新闻的真假。

# 一、大数据平台

**（一）Hadoop**

Hadoop是一个由Apache基金会所开发的分布式系统基础架构, 是一个存储系统+计算框架的软件框架。主要解决海量数据存储与计算的问题，是大数据技术中的基石。Hadoop以一种可靠、高效、可伸缩的方式进行数据处理，用户可以在不了解分布式底层细节的情况下，开发分布式程序，用户可以轻松地在Hadoop上开发和运行处理海量数据的应用程序。

Hadoop能够解决海量数据存储以及资源管理，调度和分配的问题。前者使用HDFS实现，后者通过Apache Hadoop YARN进行资源管理与分配。

1.HDFS

传统的文件系统是单机的，不能横跨不同的机器。HDFS(Hadoop Distributed FileSystem)的设计本质上是为了大量的数据能横跨成百上千台机器，但是你看到的是一个文件系统而不是很多文件系统。

HDFS是Hadoop体系中数据存储管理的基础，它是一个高度容错的系统，能检测和应对硬件故障，在低成本的通用硬件上运行。HDFS简化了文件的一次性模型，通过流式数据访问，提供高吞吐量应用程序数据访问功能，适用带有数据集的应用程序。HDFS提供一次写入多次读取的机制，数据以块的形式，同时分布存储在不同的物理机器上。

比如你说我要获取/hdfs/tmp/file1的数据，你引用的是一个文件路径，但是实际的数据存放在很多不同的机器上。你作为用户，不需要知道这些，就好比在单机上你不关心文件分散在什么磁道什么扇区一样。HDFS 为你管理这些数据。

2.YARN

Apache Hadoop YARN （Yet Another Resource Negotiator）是一种新的Hadoop[资源管理器](https://baike.baidu.com/item/%E8%B5%84%E6%BA%90%E7%AE%A1%E7%90%86%E5%99%A8/1951545" \t "https://baike.baidu.com/item/yarn/_blank)，它是一个通用资源管理系统，可为上层应用提供统一的[资源管理](https://baike.baidu.com/item/%E8%B5%84%E6%BA%90%E7%AE%A1%E7%90%86/7069136" \t "https://baike.baidu.com/item/yarn/_blank)和调度，它的引入为集群在利用率、资源统一管理和数据共享等方面带来了巨大好处，相当于一个分布式的操作系统平台，而MapReduce等运算程序则相当于运行于操作系统之上的应用程序。

YARN的基本思想是将JobTracker的两个主要功能（资源管理和作业调度/监控）分离，主要方法是创建一个全局的ResourceManager（RM）和若干个针对应用程序的ApplicationMaster（AM）。这里的应用程序是指传统的MapReduce作业或作业的[DAG](https://baike.baidu.com/item/DAG/10765028" \t "https://baike.baidu.com/item/yarn/_blank)（有向无环图）。

ResourceManager和 NodeManager构成了数据计算框架。 ResourceManager 是在系统中的所有应用程序之间仲裁资源的最终权限。NodeManager 是每台机器框架代理，负责 Containers，监视其资源使用情况（CPU，内存，磁盘，网络）并将其报告给 ResourceManager。[1]

3.Hadoop缺陷

Hadoop作为一个处理大数据的软件框架，虽然受到众多商业公司的青睐，但是其自身的技术特点也决定了它不能完全解决大数据问题。在当前Hadoop的设计中，所有的metadata操作都要通过集中式的NameNode来进行，因此不适用于低延迟数据访问，不能高效存储大量小文件并且不支持多用户写入并任意修改文件。

**（二）Spark**

Spark是[加州大学伯克利分校](https://baike.baidu.com/item/%E5%8A%A0%E5%B7%9E%E5%A4%A7%E5%AD%A6%E4%BC%AF%E5%85%8B%E5%88%A9%E5%88%86%E6%A0%A1/3755024" \t "https://baike.baidu.com/item/SPARK/_blank)的AMP实验室所开源的Hadoop MapReduce的通用并行框架，拥有Hadoop MapReduce所具有的优点。不同于MapReduce的是，使用Spark时，job中间输出结果可以保存在内存中，从而不再需要读写HDFS，因此Spark能更好地处理实时数据流，适用于数据挖掘与机器学习等需要迭代的MapReduce的算法。

Pyspark是Spark在Python语言中提供的功能库，方便用户调用Spark。由于Spark内核是在Scala语言上进行的，而Scala又是建立在Java语言上，因此在使用Spark时需要配置Java环境。在使用Python时，Pyspark通过py4j调用Scala与Java语言的方法来运行并实现程序。[2]

**（三）Hadoop与Spark的结合使用**

在使用Spark时，可以不适用HDFS运行。但是Spark on YARN必须要运行HDFS，

Hadoop和Spark都是当今热门的开源大数据框架，但两者并不是互斥的，可以一起工作。Spark没有用于分布式存储的文件系统，需要依靠Hadoop的HDFS与YARN实现文件的存储，而Hadoop无法低延迟处理大量数据，需要解读Spark的分布式数据集（RDD）实现数据计算与处理。

# 二、大数据与真假新闻定义

大数据(big data)，或称巨量资料，广义上讲指的是所涉及的资料量规模巨大到无法透过主流软件工具，在合理时间内达到撷取、管理、处理、并整理成为帮助企业经营决策更积极目的的资讯。大数据中数据丰富而信息贫乏，因此需要从中进行挖掘，找到可用的指标与特征，提取有用信息，帮助进行分析与决策。从特征论方面，大数据具有5V属性。即数据规模大（[Volume](https://baike.baidu.com/item/Volume/17610592" \t "https://baike.baidu.com/item/%E5%A4%A7%E6%95%B0%E6%8D%AE/_blank)），数据聚集快（[Velocity](https://baike.baidu.com/item/Velocity/1398152" \t "https://baike.baidu.com/item/%E5%A4%A7%E6%95%B0%E6%8D%AE/_blank)），数据类型多（[Variety](https://baike.baidu.com/item/Variety/191328" \t "https://baike.baidu.com/item/%E5%A4%A7%E6%95%B0%E6%8D%AE/_blank)），数据价值大（[Value](https://baike.baidu.com/item/Value/2285610" \t "https://baike.baidu.com/item/%E5%A4%A7%E6%95%B0%E6%8D%AE/_blank)）以及数据真实性（[Veracity](https://baike.baidu.com/item/Veracity/19362178" \t "https://baike.baidu.com/item/%E5%A4%A7%E6%95%B0%E6%8D%AE/_blank)）。

在技术推动下的社交渠道与信息传播渠道高度融合的大数据时代，新媒体技术释放出了海量信息。随着互联网时代社交媒体的持续发展，假新闻以前所未有的速度覆盖了人们日常生活的各个方面。例如，2016年美国总统大选期间，在南欧小城韦莱斯，这个人口仅5.5万的小镇拥有至少100个支持特朗普的网站，其中很多充斥着各种耸人听闻的假新闻，靠吸引流量来赚取广告费。[3]

在本数据集中，共有新闻数据四万多条，勉强符合大数据的定义。因此，笔者使用Hadoop+Spark+Python实现真假新闻的分类器。

# 三、模型理论介绍

**（一）Spark ML库简介**

分布式数据集（RDD）是一个抽象的弹性数据集合，可以灵活的处理数据。与之不同，DataFrame（Spark SQL中的DataFrame）数据组以列的形式组织起来，类似于关系数据库中的表。它是一个不可变的分布式数据集合，允许开发人员将数据结构(类型)加到分布式数据集合上，从而实现更高级别的抽象。

Spark平台提供了两个用于机器学习的库，主要面向分布式训练，针对大量数据，分别是操作DataFrame的ML和操作RDD的MLlib。Spark官方推荐使用ML，MLlib在spark2.0以后进入维护状态，只修复BUG不增加新功能。ML库中最主要的就是转换器（Transformer）、模型（Pipeline）以及评估器（Estimator）三个部分。模型中涵盖了分类、聚类与回归算法，提供了目前常用的机器学习功能的API接口。

综上，笔者选择使用DataFrame处理数据，使用ML库进行特征选取、分类器的构建和模型评估。

**（二）机器学习理论**

在本案例中，在文本特征提取方面使用了波特词干还原和Word2Vec方法，在模型上使用了逻辑回归、决策树、随机森林、支持向量机和多层感知机，在评价指标中使用了查准率、查全率、F1值、准确率和AUC。本部分介绍上述所有方法模型和指标。

1.波特词干还原

在英语中，一个单词常常由另一个单词增加或更改后缀形成，例如单词“fast”和单词“fastly”，两个单词所表达的意思相近。因此在文本处理中，可以考虑将同类词提取词干，将单词规范化，方便后续的处理与统计。词干提取方法有许多种，波特词干提取算法（Porter Stemmer）是其中具有重要代表意义的一种[4]。

波特词干提取的原理分六步。1）将复数，ed以及ing结尾的单词还原；2）如果单词中含元音并以y结尾，将y改为i；3）将双后缀单词映射为单后缀；4）处理如ness等特殊后缀；5）去除特定情况下的ant、ence等后缀；6）若单词长度大于1，去除末尾的字符e。

2.Word2Vec

2013年，Google开源了一款用于词向量计算的工具——Word2Vec，引起了工业界和学术界的关注。Word2Vec是词嵌入模型（word embedding）的一种。词嵌入，顾名思义，就是将一些特征给嵌入到词汇里面，让它能表现出可用的信息。Word2Vec是一种基于浅层神经网络的词嵌入模型。它主要由CBOW（Continues bags of words）和Skip-gram模型构建而成。[5]

首先，Word2Vec可以在百万数量级的词典和上亿的数据集上进行高效地训练；其次，该工具得到的训练结果是词向量，可以很好地度量词与词之间的相似性。两模型具体如图所示。

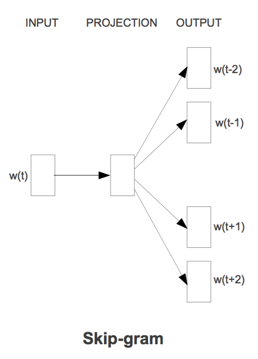
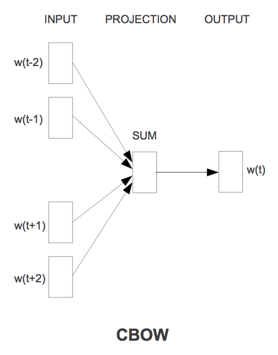


图1 Word2Vec两模型图示

CBOW模型改造子NNML模型，移除了前向反馈网络中非线性的隐藏层，将中间层与输出层的softmax连接，忽略上下文环境的序列信息，输入的所有词向量均汇总到同一个编码层。总的而言，CBOW是在通过文本对目标词的预测中得到词向量的表达。与之相反，Skip-gram是从目标词对文本的预测中得到词向量的表达。

3.逻辑回归

逻辑回归是一种广义的线性回归分析模型，是一个非常经典的算法，被称为回归但实际上是一个分类模型，属于监督学习（Supervised Learning）的一种。其本质是在线性模型的基础上增加了Sigmoid函数，假设数据服从这个分布，然后使用极大似然估计做参数的估计。逻辑回归的整个过程就是面对一个回归或者分类问题，建立代价函数，然后通过优化方法（如极大似然）迭代求解出最优的模型参数，然后测试验证我们这个求解的模型的好坏。

在实际操作中，由于逻辑回归中可以直接看到各个特征的权重，因此其可解释性较强，受到人们的青睐。

逻辑回归是一种线性分类器，而线性分类器是文本分类中一种重要的分类器。在二维平面上，线性分类器就是一条直线，将两组样本数据通过一条直线分隔开来。我们可以将二维平面上的线性分类器推广到高维空间中，此时分类器就变为了一个超平面，样本被该超平面分割为两部分，这个代表线性分类器的超平面也被称为决策超平面（Decision Hyperplane）。

线性分类器简单易懂，但在处理一些特定数据中存在较大困难。一个典型的例子就是在数据集中存在一些噪声样本，他们与类别的总体分布不太吻合，这种噪声的存在导致线性分类器无法使用一个超平面将类别分开，这也是造成新型分类器训练困难的原因之一。

4.决策树

决策树（Decision Tree）是监督学习的一种，是一种常见的，逻辑简单的机器学习算法，它是一种树形结构，所以叫决策树。

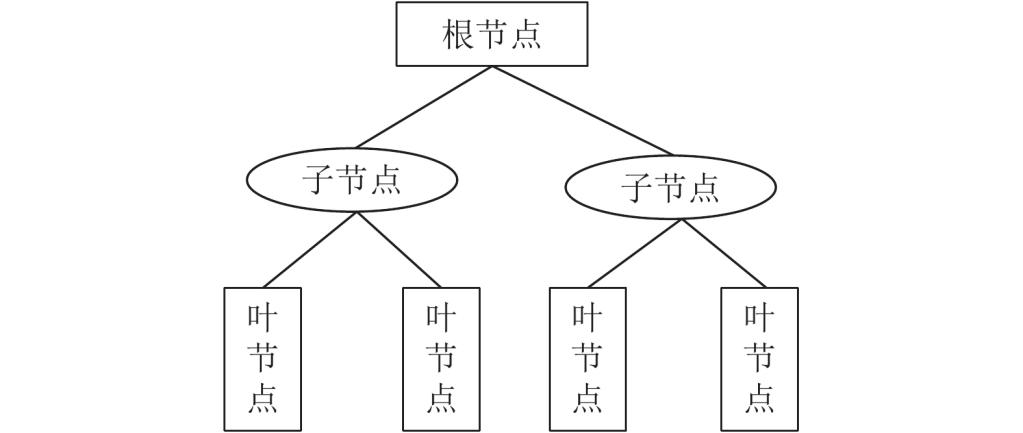


图2 决策树树形结构图示

决策树的每个内部节点表示在一个属性上的测试，每个分枝代表一个测试输出，而每个树叶结点代表类或类分布。树的最顶层结点是根结点。决策树分类方法采用自顶向下的递归方式，在决策树的内部结点进行属性值的比较并根据不同的属性值判断从该结点向下的分枝，在决策树的叶结点得到结论。所以从决策树的根到叶结点的一条路径就对应着一条合取规则，整棵决策树就对应着一组析取表达式规则。

决策树模型使用信息熵（Information Entropy）作为根节点与叶子节点的衡量标准，代表某节点下样本种类的混乱程度。决策数通过最小化熵获得最大的信息增益（Information Gain），实现决策树的构建，基本思路就是随着决策树深度的增加快速降低模型的熵。

一般而言，信息增益越大，意味着使用该节点来划分所获得的熵减越大。著名的ID3决策树学习算法九十一信息增益为准则来选择划分属性。

除了基础的训练方法，决策树还有一些实用技巧。在发生过拟合时，可以采用剪枝（Pruning）的方法对决策树进行处理。剪枝的基本策略包括预剪枝和后剪枝两种，前者在决策树生成过程中，在划分前对每个节点进行估计，若当前节点划分不能带来泛化性能的提升，则停止划分并将该节点定位叶节点；后者则是先从训练集中生成一棵完整的树，然后自底向上地对非叶子节点考察与替换，提高树的泛化能力。

5.随机森林

集成方法一般包括Bagging，Stacking和Boosting三种。Bagging通过多个弱分类器投票的方式得到最终结果；Stacking通过将多个弱分类器的结果放在一个神经网络中，得到最终结果；Boosting方法通过弱分类器预测减少前一个弱分类器的错误实现，随着分类器越多其增强程度越大（可能过拟合）。

随机森林（Random Forest）是一种由决策树构建的集成方法。ML中的随机森林方法属于Bagging方法的一种扩展变体，引入随机属性选择，通过自助法（bootstrap）重采样技术，从原始训练样本集N中有放回地重复随机抽取k个样本生成新的训练样本集合，然后根据自助样本集生成k个分类树组成随机森林，新数据的分类结果按分类树投票多少形成的分数而定。其实质是对决策树算法的一种改进，将多个决策树合并在一起，每棵树的建立依赖于一个独立抽取的样品，森林中的每棵树具有相同的分布，分类误差取决于每一棵树的分类能力和它们之间的相关性。特征选择采用随机的方法去分裂每一个节点，然后比较不同情况下产生的误差。能够检测到的内在估计误差、分类能力和相关性决定选择特征的数目。[6]

随机森林简单、容易实现、计算开销小，此外，它在很多现实任务中展现出了强大的性能。随机森林的收敛性与Bagging相似，然而随着个体学习器数目的增加，随机森林通常会收敛到耕地的泛化误差。

6.支持向量机

支持向量机（support vector machines, SVM）最早是由 Vladimir N. Vapnik 和 Alexey Ya. Chervonenkis 在1963年提出，目前的版本（soft margin）是由 Corinna Cortes 和 Vapnik 在1993年提出，并在1995年发表。深度学习（2012）出现之前，SVM以其超前的数学逻辑以及优秀的表现声名远扬。[7]

SVM是一种二分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机。所谓线性分类，简而言之，一条线可以按照数据可分的原理将需要分类的数据划分开来。SVM还包括核技巧，这使它能够成为非线性分类器。适合非线性且高维的分类问题。SVM的的学习策略就是间隔最大化，可形式化为一个求解凸二次规划的问题，也等价于正则化的合页损失函数的最小化问题。

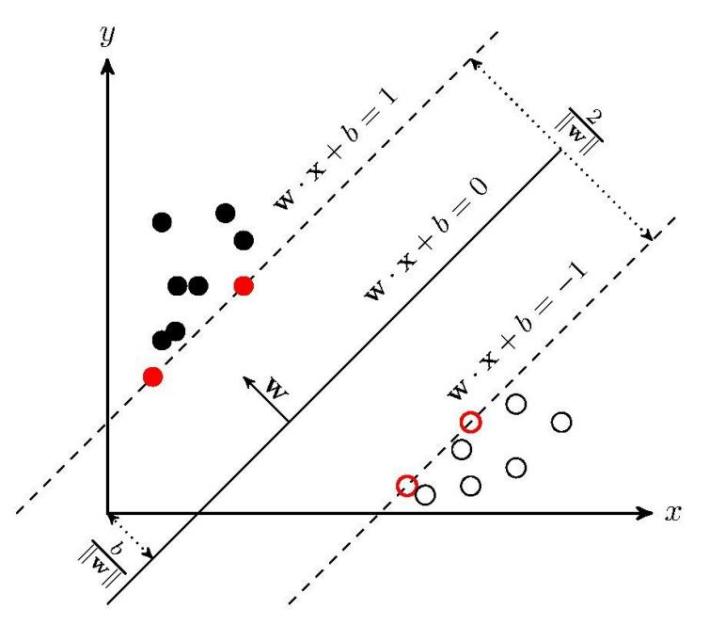


图3 支持向量机SVM计算图示

SVM所做的事情如图所示，找到两样本间一条直线，使这条直线将两样本分割开并里两样本中最近的点的距离（间隔，margin）最大。直观地看，一个处于中间空白出地决策面比那些靠近某个类的决策面更好。

对于在文本分类中很普遍存在的高维空间问题来说，有时数据是线性可分的，但是一般下情况下很难成立。即时线性可分，也应该优先考虑那些能够将大部分数据分开而忽略一些奇异噪声的解决方案。为实现这一目的，引入松弛变量作为允许模型在未满足间隔需求下的惩罚量与代价。这种方法即为软间隔分类，属于传统支持向量机的扩展。实践证明，软间隔分类在数据分类任务中效果更好。

7.多层感知机

在多层感知机的每一层中，都存在着数个神经元（Neuron）。神经元接收上一层神经元的输出，通过内置的激活函数给出输出，实现非线性的数据转变。

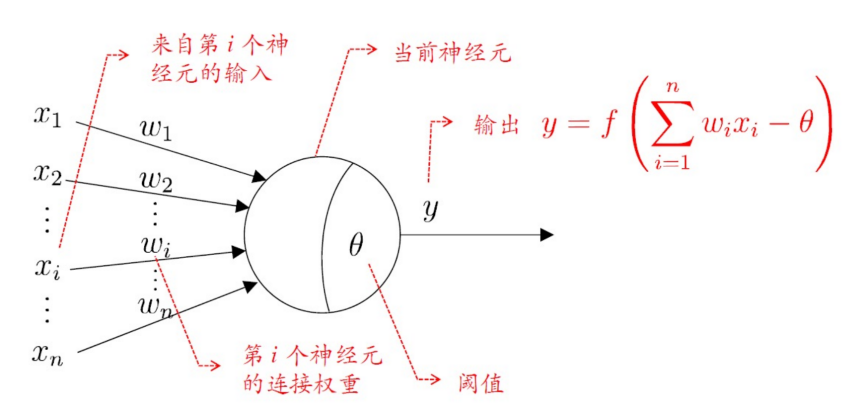


图4 神经网络中神经元模型

在训练神经网络的过程中，神经元的连接权重被实时的更新，不断收敛直到达到阈值。

感知机由两层神经元组成，能够容易地实现逻辑与、或、非运算。然而由于感知机只用有一层功能神经元，其学习能力十分有限，处理非线性问题存在困难。

多层感知机（MLP，Multilayer Perceptron）是一种前向结构的人工神经网络，包含输入层、输出层及多个隐藏层，最简单的MLP只含一个隐层，即三层的结构。多层感知机的学习能力比单层感知机强得多。但是简单的感知机学习规则在多层感知机地学习中明显不够用。误差逆传播算法（Error BackPropagation，简称BP）应运而生。误差逆传播算法是最成功的训练多层前馈神经网络的学习算法之一，其基于梯度下降策略，以目标的负梯度方向对参数进行调整，不断更新连接权值与阈值，直到达到停止条件。

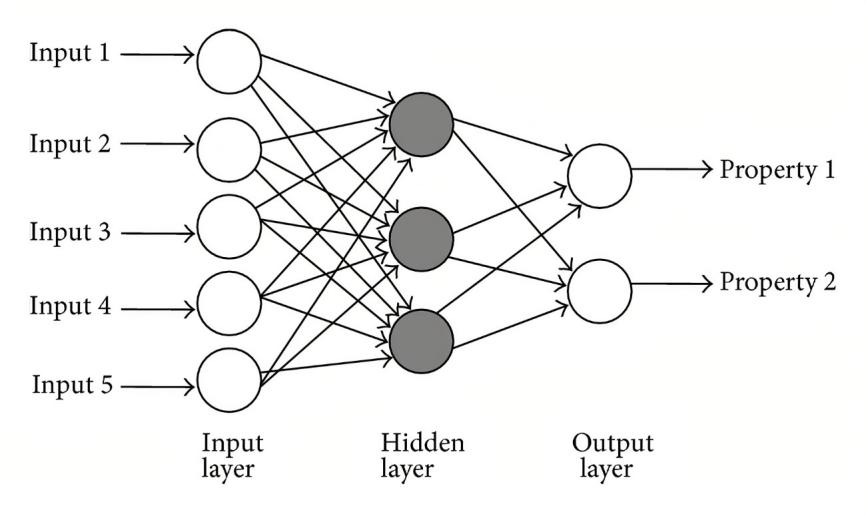


图5 多层感知机示意图

在使用多层感知机时，首先需要指定网络中包含神经元的总层数，并指定每层神经元的个数。一般来说，隐藏层中神经元数量越多，其过拟合的可能性也就越大。如何设置隐层神经元的个数仍是未解决的问题，实际应用中常用“试错法”调整。[8]

8.查准率、查全率、F1值和准确率

首先，在模型训练完毕后，我们可以使用模型对测试集进行判断。对测试集中的每个样本预测后，可以通过预测标签与真实标签作比较，得到混淆矩阵（Confusion Matrix）。

混淆矩阵的列代表了预测类别，行代表数据的真实归属类别，使用P和N表示。因此，混淆矩阵可以被里三成四个部分，如图表所示。

表1 混淆矩阵示意图

|  |  |  |
| --- | --- | --- |
| **混淆矩阵** | **P** | **N** |
| **P** | TP | FP |
| **N** | FN | TN |

查准率（Precision，简记为P）又称正确率，是预测正例的所有结果中真实正例所占的比例，定义为

查全率（Recall，简记为R）又称召回率，亦或称为灵敏度（TPR），指的是所有真实正例中，被预测为正例样本所占比例，定义为

在一些场景下，查准率与查全率均无法很好地反映模型的效果。例如在一个正负样本不均衡的模型中，正例与负例的比例为9：1，那么分类器只要将所有的样本都分为正例，就能获得0.9的查准率。此外，在另一些情况下，查全率也无法很好的度量模型的真实效果。因此，同时采用查准率和查全率两个指标来度量模型的表现。F值是一种融合了查准率和查全率的指标，定义为两者的调和平均值。

平衡F值（balanced F measure）中查准率和查全率的权重相等，即β=1，用或表示。

准确率（Accuracy）是分类模型所有判断正确的结果占总样本数的比重，定义为

9.AUC

提到AUC就不得不提两个概念，ROC曲线和PR曲线。

ROC曲线的全称是Receiver Operating Characteristic Curve，中文名字叫“受试者工作特征曲线”，又被称为感受性曲线（sensitivity curve）。

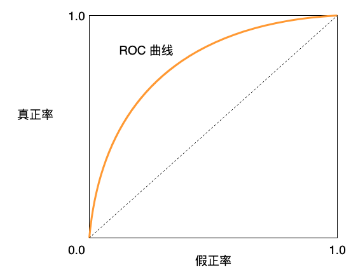


图6 ROC曲线

该曲线的横坐标为假阳性率（False Positive Rate，FPR），纵坐标为真阳性率（True Positive Rate，TPR）。ROC曲线的AUC值（Area Under Curve）定义为这条曲线与下方坐标轴围成的面积。显然，由于横坐标与纵坐标的设定，这个面积小于一，又由于ROC曲线一般都处于y=x这条直线的上方，因此AUC的取值范围一般在0.5到1之间，AUC的值越接近1，说明模型的效果越好，越接近0.5就说明效果越差。ROC曲线能很容易的查出任意[阈值](https://baike.baidu.com/item/%E9%98%88%E5%80%BC/7442398" \t "https://baike.baidu.com/item/AUC/_blank)对学习器的泛化性能影响，有助于选择最佳的阈值。ROC曲线越靠近左上角，模型的准确性就越高。最靠近左上角的ROC曲线上的点是分类错误最少的最好阈值，其假正例和假反例总数最少。此外，ROC曲线可以对不同的学习器比较性能。将各个学习器的ROC曲线绘制到同一坐标中，直观地鉴别优劣，靠近左上角的ROC曲所代表的学习器准确性最高。因此使用ROC-AUC作为衡量模型表现的一个指标。

PR曲线全名为Precision-Recall Curve，常被用在信息提取领域。在数据集中类别分布不均匀时，可以使用PR曲线代替ROC曲线衡量模型的表现效果。

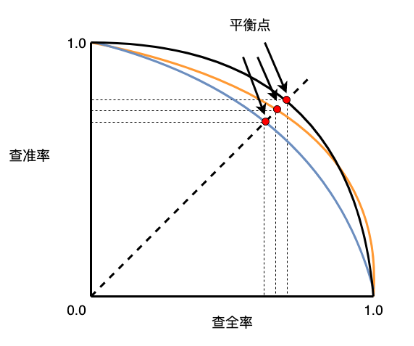


图7 PR曲线

PR曲线的横轴表示查全率，纵轴表示查准率，与ROC曲线类似，可以对曲线下方与坐标轴形成图像的面积进行计算，得到PR-AUC值用以判断模型的表现。从图像上来看，PR曲线越靠近右上角说明模型的性能越好。因此使用PR-AUC作为衡量模型表现的一个指标。

# 四、模型设计与训练

**（一）环境配置**

本项目在Linux虚拟机上进行虚拟机搭建在Oracle VM VirtualBox上，使用系统为ubuntu。使用jdk8（jdk1.8）、hadoop-3.3.1与spark3.2.0作为平台工具，通过Anaconda装载jupyter-notebook，用作编码的IDE。此外，本实验需要工具包有 pandas、numpy、nltk、pyspark以及sklearn。环境的具体配置流程就不再做介绍和展示。

**（二）数据介绍**

本项目使用的数据集源自Kaggle用户Ajay Vamsi上传的自2015年3月31日至2018年2月19日的新闻数据。数据包含新闻标题（title）、新闻文本（text）、新闻类别（subject）以及新闻发布时间（date）四个指标以及真假新闻的标签。

数据情况如图所示，共包括各类别新闻44898条，其中正例21417条，负例23481条。

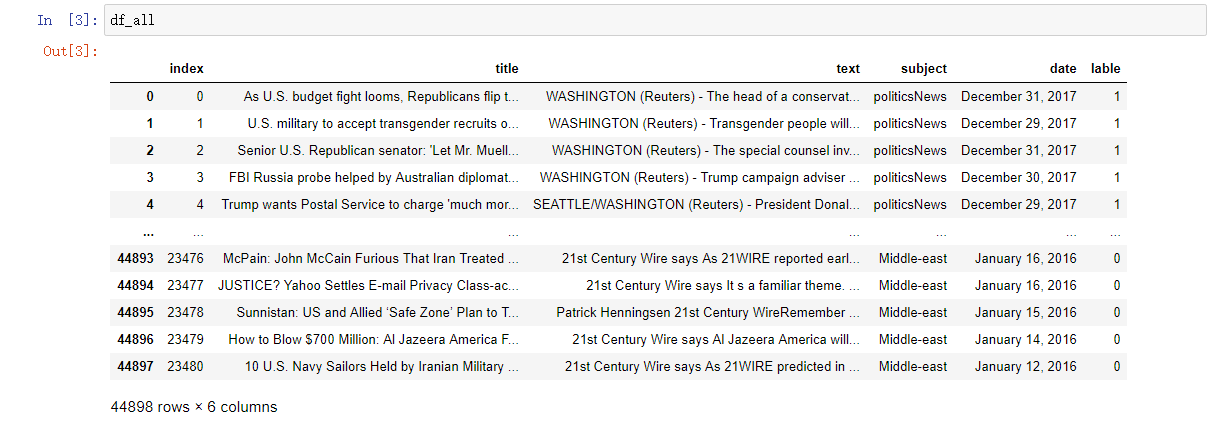


图8 数据基本情况

去脏去重去空后，数据共包括各类别新闻44888条，其中正例21417条，负例23471条，正负例数量占比合适。

**（三）指标分析**

首先分析各指标。在数据中，title指新闻标题，text指新闻内容，subject值新闻种类，date指新闻发布日期。首先，毋庸置疑的是title、text是判断新闻真假的根本依据，subject应该也能一定程度上的反应出新闻的某些特征。

关于date数据，我最开始认为，可以用格林威治时间戳转成标量放入模型中，但后来一想，模型做的目的就是为了预测以后发的新闻是真是假，新发新闻的时间一定不在训练范围内，因此使用date不仅不能能带来性能的提升，反而可能会造成有偏。因此不使用date作为输入。

在仔细观察完数据后，我发现这样一种现象：数据集中真假新闻的类别是不相交的，即属于politicsNews和worldnews类的都是真新闻，属于其他六个类别的都是假新闻，因此新闻的真假与类别具有强相关性。实际使用过程中，我先拿带类别数据的指标进行训练，得到了0.986以上的准确度（Accuracy），显然如果使用类别数据做为模型的输入特征并不合适，因此本项目不使用新闻类别subject作为输入，只考虑新闻标题title和新闻文本text作为模型的输入训练和测试模型。

**（四）特征构建**

首先，我使用波特词干器对新闻标题和文本的单词进行词干提取与还原，同时使用nltk自带的停用词库将新闻中的停用词删除并删除一些特殊符号。下图展示了对标题做词干还原和去停用词后的结果。其中深色代表原标题，浅色代表波特词干还原后的结果。

表2 波特词干提取结果

|  |  |
| --- | --- |
|  | **新闻标题** |
| **1** | As U.S. budget fight looms, Republicans flip their fiscal script |
|  | u.s. budget fight loom republican flip fiscal script |
| **2** | U.S. military to accept transgender recruits on Monday: Pentagon |
|  | u.s. militari accept transgend recruit monday pentagon |
| **3** | FBI Russia probe helped by Australian diplomat tip-off: NYT |
|  | fbi russia probe help australian diplomat tip-off nyt |
| **4** | Trump wants Postal Service to charge 'much more' for Amazon shipments |
|  | trump want postal servic charg 'much ' amazon shipment |
| **5** | White House, Congress prepare for talks on spending, immigration |
|  | white hous congress prepar talk spend immigr |

在做完词干提取后，通过pyspark中的Tokenizer方法实现切词。在对新闻标题进行切分时未遇到问题，但在切分新闻内容时遇到了问题。由于pyspark中的Tokenizer方法会错误的切分一些较长的单词或字符串，或是直接返回空值，因此最终切分结果在word2vec中会发生错误，无法正常调用fit方法进行训练。错误类型为Py4JJavaError。因此，在对新闻内容做切词处理时，我直接使用了split方法进行切割，将切分好的词组以list的形式存放在列中，实现与Tokenizer相同的效果。

下面对切好的单词做处理。首先，我考虑使用Tf-idf构建词向量。Tf-idf是一种结合词频以及你逆档频率的词向量构建方法，实践证明，Tf-idf在处理文本数据中效果较好。虽然Pyspark中并未提供统一化的Tf-idf方法，但提供了分散的Hashingtf和IDF方法，可以用来构建词向量。然而由于虚拟机内存有限，再加上Tf-idf方法构建的词向量为一个稀疏矩阵，结果为SparseVector格式，若使用低纬度构建词向量会造成大量信息遗漏，使用高纬度构建词向量又可能会导致在训练过程中内存溢出。因此最后舍弃了使用Tf-idf的方法。

使用word2vec将单词打做词向量的形式。此处我将title和text均通过w2v打成20维的词向量。合并加总为40维的词向量。此处我使用sklearn中的TSNE方法将w2v生成的词向量降至二维，观察词向量在二维空间中的分布情况。左侧为训练集，右侧为测试集。注意，这里边由于训练数据量的问题，所以两图的横纵坐标表示并不是一个量值。

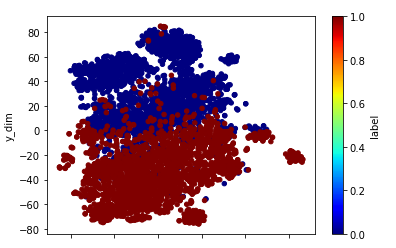
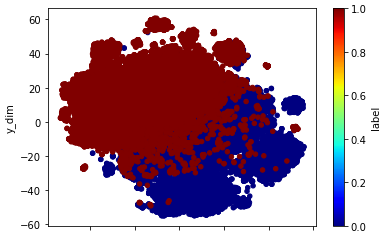


图9 TSNE降维处理结果（左侧为训练集，右侧为测试集）

可以观察到，正例和负例经过word2vec被很好的分隔开了。

**（五）模型选择**

由于word2vec构建的词向量中存在负值，因此不能使用朴素贝叶斯进行分类。所以我使用了Pyspark自带的ML机器学习库中分类器模块中逻辑回归LogisticRegression，决策树DecisionTreeClassifier，随机森林RandomForestClassifier，支持向量机LinearSVC和多层感知机MultilayerPerceptronClassifier共五种不同的分类器。

前四种分类器使用默认参数，多层感知机使用两个隐藏层，输入层神经元数为40（词向量维度），两隐藏层神经元数为16、2。

# 五、结果分析与不足

模型训练结果如下所示。

表3 各分类器表现

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **precession** | **recall** | **f1** | **accuracy** | **roc-auc** | **pr-auc** |
| **逻辑**  **回归** | 0.944668 | 0.944310 | 0.944291 | 0.944310 | 0.944169 | 0.940780 |
| **决策树** | 0.888427 | 0.887525 | 0.887429 | 0.887525 | 0.887281 | 0.878294 |
| **随机**  **森林** | 0.916027 | 0.916027 | 0.916027 | 0.916027 | 0.916014 | 0.897817 |
| **LinearSVC** | 0.943058 | 0.942776 | 0.942759 | 0.942776 | 0.942649 | 0.937843 |
| **MLP** | 0.948850 | 0.948476 | 0.948458 | 0.948476 | 0.948333 | 0.946091 |

结果表明，各分类器在此数据集上都有较好的表现。多层感知机在处理本问题上表现最佳，但是决策树、随机森林和支持向量机效果甚至不如逻辑回归。

# 六、总结与展望

**（一）实验总结**

由于本实验是使用pyspark对文本数据进行处理并构建分词器，因此较纯数值数据的分类器构建更加困难。在实验过程中，我多次遇到了py4j调用的错误，其中既有转换类型问题，又有内存溢出的问题，但报错类型都是同一种，因此比较让人头疼。

此外在实验过程中，我也深刻地感受到了RDD与Spark.SQL.DataFrame之间的差别与联系。DataFrame确实比RDD更容易操作和处理，但没有RDD用起来灵活变通。

无论如何，在本次实验中，我系统的掌握了Spark环境配置，Pyspark ML库和SQL库的使用，让我更深入的掌握了不同的分类器与数据处理与清洗方法，也让我对自然语言处理与文本挖掘有了更加清晰的认识。在实验过程中，我记忆最深的地方就是学到了Pyspark中自带的两种向量类型，一种为SparseVector，表示稀疏向量；另一种为DenseVector，表示稠密向量。在使用过程中，前者只展示存储元素的个数以及非零元素的标号和值，后者则是将所有元素全罗列出来。不同的向量类型有助于对模型的训练以及结果的展示。

**（二）实验不足**

本实验不足之处有三，一是仅考虑了word2vec对词向量的构建。之后的实验可以同时比较如Glove模型，tfidf等模型对词向量的构建。二是本实验并没有对各分类器调参，因此最终结果可能并不能很好的代表分类器的真是性能。比如SVM实际上是一种性能极佳的分类器，但在不调参的情况下效果甚至不如逻辑回归。第三点是本实验对实验数据的切分为9：1切分训练集和测试集。后续可以使用k折交叉验证等方法，排除样本切分造成的实验偏差。

**（三）展望**

Spark因其简单性、通用性、容错性和高性能而获得了工业界和学术界的极大兴趣和贡献。目前，Spark的正式版本得到了部分Hadoop主流厂商的支持，这说明业界已经认可了Spark，Spark也被许多企业尤其是互联网企业广泛应用到商业项目中。然而Spark在内存资源管理与RDD操作等方面仍然存在问题。

在内存资源管理中，作为用Scala构建的内存处理平台，Spark的性能对其内存配置和jvm的使用非常敏感。内存资源分为两部分，一个是用于RDD缓存，另一个用于任务的工作内存，用来存储任务执行期间创建的对象。这种内存分配方式导致性能会因为错误的配置而大幅下降。

在对RDD的操作中，当前Spark的RDD有几个未解决的问题。首先，它仅允许对RDD执行粗粒度操作。第二，当前的RDD是不可变的。除了对现有的RDD进行修改之外，任何更新操作都将生成新的RDD，从而导致存储资源的浪费。

无论如何，Spark平台具有旺盛的生命力以及社区活跃度。随着Spark的逐渐成熟，相信越来越多的人会接触并使用Spark，Spark会越来越受到欢迎。

**参考文献**

[1] White T . Hadoop: The Definitive Guide[J]. O'rlly Media Inc Gravenstn Highway North, 2012, 215(11):1 - 4.

[2] Zaharia M , Chowdhury M , Franklin M J , et al. Spark: Cluster computing with working sets. 2010.

[3]佚名. 马其顿韦莱斯 靠炮制假新闻获利的小城[J]. 看世界, 2017(1):1.

[4] Porter M F . An algorithm for suffix stripping[J]. Program Electronic Library & Information Systems, 2006, 14(3):130 - 137.

[5] Mikolov T , Chen K , Corrado G , et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.

[6]V.F. Rodriguez-Galiano and M. Chica-Olmo and F. Abarca-Hernandez and P.M. Atkinson and C. Jeganathan. Random Forest classification of Mediterranean land cover using multi-seasonal imagery and multi-seasonal texture[J]. Remote Sensing of Environment, 2012.

[7]张学工. 关于统计学习理论与支持向量机[J]. 自动化学报, 2000, 26(1):11.

[8]韦岗, 贺前华. 关于多层感知器的函数逼近能力[J]. 信息与控制, 1996, 25(6):4.