Lab5 综合实验

姓名：陈鸿绪　　学号：PB21000224　　日期：2024.2.1

一. 实验目的：

巩固课程中所学的模型，学会应用不同的模型到实际任务当中，考察分析问题到模型构建最后解决问题的综合能力。

二. 实验原理：

1. **多重感知机：**多层感知机（MLP）是一种全连接的前馈神经网络模型，它由多个神经元按照层次结构组成。在 MLP 中，输入层、隐藏层和输出层是逐层连接的，数据从输入层经过多个隐藏层的非线性变换，最后到达输出层进行分类或回归操作。隐藏层的神经元通过线性变换和激活函数引入非线性因素，使得神经网络可以任意逼近任何非线性函数。该算法通过计算损失函数对网络参数的梯度，并根据梯度更新参数，以最小化损失函数。

2. **决策树：**决策树基于树形结构进行决策，从根节点开始，根据数据的特征进行分支，直到达到叶节点并给出最终的决策结果。决策树的生成过程通常包括特征选择、决策树生成和剪枝三个步骤。其中，特征选择是决定哪个特征作为划分标准的关键步骤，常见的特征选择方法包括信息增益、增益率和基尼指数等。

3. **逻辑回归：**在面对一个分类问题时，逻辑回归通过建立代价函数，然后使用优化方法迭代求解出最优的模型参数，最后测试验证求解的模型的好坏。其核心是通过逻辑函数将线性回归的结果映射到(0,1)之间，从而得到样本点属于某一类别的概率。

4. **支持向量机：**支持向量机（SVM）是一种监督学习算法，主要用于分类和回归分析。它的核心思想是将原始数据映射到高维空间，然后在高维空间中找到一个最优超平面，以实现数据的分类或回归预测。SVM 通过找到一个最大间隔的超平面来分割数据，这个超平面由支持向量确定，它们是离决策边界最近的点。SVM 可以处理线性可分和线性不可分的情况，对于线性不可分的情况，可以使用核函数将数据映射到更高维的空间，使其变为线性可分。

5. **贝叶斯分类器：**贝叶斯分类器基于贝叶斯定理，通过计算给定样本属于各个类别的概率，选择概率最大的类别作为分类结果。具体来说，贝叶斯分类器通过构建概率模型，将特征和类别之间的关系表示为概率分布，然后利用这些概率分布计算出给定样本属于各个类别的概率。

6. **Xgboost：**Xgboost 是一种基于梯度提升算法的机器学习模型，其原理是通过迭代地构建一系列决策树来拟合数据，并使用梯度提升方法来最小化损失函数。在 Xgboost 中，每个新的决策树都是基于前一树的残差进行构建的。具体来说，Xgboost 通过最小化损失函数来学习每个决策树的目标函数，该损

失函数由两部分组成：正则化项和损失函数项。正则化项用于防止过拟合，而损失函数项则基于真实值和模型预测值之间的差距来计算。在训练过程中，Xgboost 使用二阶泰勒展开来逼近损失函数，这使得模型能够快速收敛并提高训练效率。

三. 实验步骤：

1. **数据处理：**从 train.json、test.json 中读取数据；首先考虑缺失数据，将缺失数据的属性进行处理；其次考虑冗余属性，如序号 ID、包含 color 或者 url 的属性，冗余属性对预测没有帮助。再根据数据类型和必要处理程度，将一些不必要处理或者难以处理的非数值属性去除，如"description"。最后字符型数据通过字典映射为整型数据，便于后续的处理。为了后续交叉验证法，我们将数据集划分为 5 折。

2. **相关性分析：**将数据处理剩下的 17 个属性与剩余属性"label"进行相关性分析，通过热图刻画属性之间的相关度，根据图中数据，和"label"数据表现出一定相关性的有"statuses_count""geo_enabled","has_extended","profile"等。

3. **模型训练：**我们将分别使用多重感知机、决策树、逻辑回归、支持向量机、Xgboost 以及贝叶斯分类器，通过交叉验证法分别进行模型训练，利用准确率和 F1 分数进行性能评估：

a. **多重感知机：**激活函数分别尝试了 ReLU、Tanh、Logistics 以及线性函数，发现对于 Logistics 函数效果最优，通过对隐层、隐藏层神经元个数以及正则项系数进行调参，发现在隐层、隐藏层神经元个数均为 20，正则项系数取 0 时，模型达到较优效果。特征选择在"listed_count","favourites_count","geo_enabled","verified","statuses_count","lang","has_extended_profile"时，模型的准确率大概会比全部特征都囊括高出 3-4%。由交叉验证法，我们最终得到总共五组准确率以及 F1 分数，得到的两项平均数分别为 0.75732，0.80197。

b. **决策树：**通过比较不同的评价指标，'gini'指数和'entropy'指数,发现使用'gini'指数得到的模型平均预测准确率会比'entropy'指数高出 1-2%。同时决策树模型选择"listed_count","favourites_count","geo_enabled","statuses_count"四个特征，且 min_samples_leaf（在叶节点所需的最小样本数。如果一个节点的样本数少于这个值，那么该节点就会被视为叶节点）为 16, max_depth（树的最大高度）为 6 时，模型准确率也达到较优状态。由交叉验证法，我们最终得到总共五组准确率以及 F1 分数，得到的两项平均数分别为 0.74824，0.79083。

c. **逻辑回归：**通过比较不同的正则形式，发现 L1 正则化效果比 L2 正则化的准确率略好 0.2%, 且正则项倒数系数 C 在取值为 0.8-1.0 之间可以取得最优情况。由交叉验证法，我们最终得到总共五组准确率以及 F1 分数，得到的两项平均数分别为 0.76184，0.80850。

d. **支持向量机：**实验中尝试了"rbf","linear","poly","sigmod"等支持向量机不同的核函数，使用"sigmod"核函数的准确率只有 0.68，使用"linear"核函数并加上适当的正则系数的准确率在 0.750，"rbf"核函数在 gamma 取 0.016 时可以达到极大准确率 0.748，多项

式核"poly"在degree取1时可以达到最优情形0.752的准确率。所以选择多项式核"poly"，且 degree 取 1。特征选择为"followers_count","friends_count","favourites_count","geo_enabled","verified","statuses_count","lang","is_translation_enabled","profile_background_tile","has_extended_profile"。此时由交叉验证法，我们最终得到总共五组准确率以及 F1 分数，得到的两项平均数分别为 0.75227，0.80704。

e. **Xgboost**：由于该问题是二分类问题，所以令 objective 为'binary:logistic'，通过不同参数的对比，确定 colsample_bytree（构建每棵树时的采样比例）在 0.5，learning_rate（学习率）调整为 0.2，max_depth（树的最大深度）调整为 15，alpha（正则化参数）设置为 5，n_estimators（代表决策树弱学习器的个数）设置为 150。特征选择为"followers_count","friends_count","listed_count","favourites_count","geo_enabled","verified","statuses_count","lang","is_translator","is_translation_enabled","profile_background_tile","has_extended_profile","default_profile_image","following","translator_type"。 此时由交叉验证法，我们最终得到总共五组准确率以及 F1 分数，得到的两项平均数分别为 0.787001，0.82075。

f. **贝叶斯分类器**：使用了两个种类的贝叶斯分类器，'GaussianNB'与'BernoulliNB'，根据两者准确率比较，高斯朴素贝叶斯分类器的准确率 70-71%，而伯努利朴素贝叶斯分类器的准确率在 74%-75%，高出至少 3 个百分点，所以选择伯努利朴素分类器，且特征选择为"followers_count","favourites_count","geo_enabled","verified","statuses_count","has_extended_profile","default_profile"。由交叉验证法，我们最终得到总共五组准确率以及 F1 分数，得到的两项平均数分别为 0.74170，0.78474。

4. **模型预测**：最终结合准确率以及 F1 分数认定 Xgboost 为表现最好的模型，将其应用到 test.json 数据集进行预测并将预测结果填入文件。对训练集的选定特征进行 tnse 降维，得到两个主成分，将两个主成分与分类标签进行二维可视化。

四． 实验结果

1. 数据处理结果

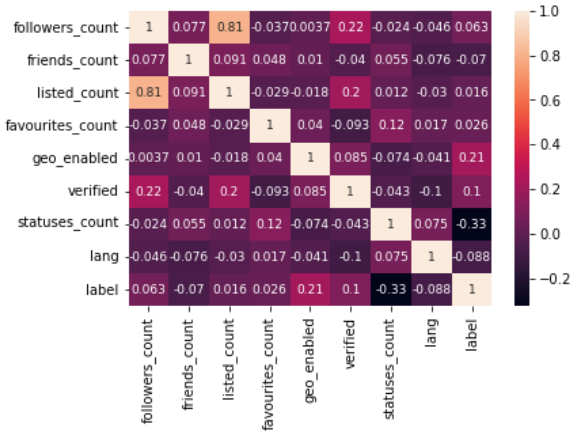
	followers_count	friends_count	listed_count	favourites_count	geo_enabled	verified	statuses_count	lang	is_transl
0	28533.0	1164.0	45.0	6006.0	1.0	0.0	7446.0	8	0.0
1	299192.0	1721.0	1517.0	401.0	1.0	1.0	25344.0	8	0.0
2	89051.0	743.0	307.0	154.0	0.0	0.0	47479.0	4	0.0
3	43211.0	237.0	29.0	358.0	0.0	0.0	16586.0	8	0.0
4	113300.0	16422.0	112.0	37.0	1.0	0.0	70666.0	14	0.0
...	...	...	...	...	...	...	...	...	...
1981	912.0	439.0	0.0	1594.0	1.0	0.0	30501.0	3	0.0
1982	113402.0	85.0	452.0	15349.0	0.0	0.0	35623.0	8	0.0
1983	1285.0	944.0	7.0	5568.0	1.0	0.0	39747.0	14	0.0
1984	48561.0	16802.0	56.0	4425.0	1.0	0.0	6917.0	8	0.0
1985	102646.0	39725.0	213.0	107562.0	0.0	0.0	15448.0	14	0.0

图一． **已处理的训练数据（部分）**

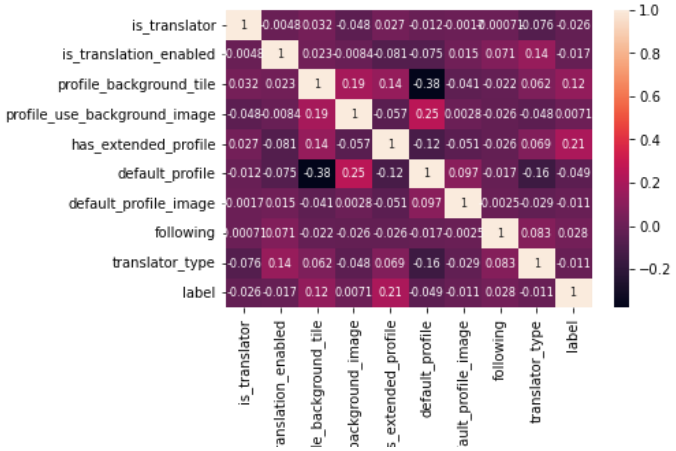
以上是数据处理完成后，训练数据的部分截图。经过上述一系列数据处理结果，

所有冗余属性被删除，包含绝大多数缺失数据的属性亦被删除，且所有的字符属性被字典映射到离散的整数。值得注意的是，实验中数据规范化步骤放入训练步骤之中，所以上图有的属性取值相对别的属性数量级更大。

2. 相关性分析结果



图二. 部分属性之间的热图



图三. 部分属性之间的热图

通过热图的相关性分析，我们可以得到经过处理的特征与"label"属性的相关性大小，理论上而言与"label"相关性越大的属性，则特征选择就越倾向于选择该属性。根据上图，我们发现"statuses_count"，""geo_enabled",
"has_extended_profile","profile_background_tile"是相对于'label'属性相关性较大的四个属性，所以在后续不同模型的特征选择中需要格外关注这几个特征。

3. 各个模型的性能

a. 多层感知机参数：

$$MLPClassifier(activation='logistic', alpha=0, hiddenlayer\_sizes=(20,20), random\_state=1, max\_iter=1000)$$

b. 决策树参数：

$$DecisionTreeClassifier(criterion='gini', min\_samples\_leaf=16, max\_depth=6, random\_state=2024)$$

c. 逻辑回归参数：

$$LogisticRegression(penalty='l1', C=0.9, solver='liblinear')$$

d. 支持向量机参数：

$$svm.SVC(kernel='poly', random_state=32, degree=1)$$

e. Xgboost 参数：

$$xgb.XGBRegressor(objective='binary:logistic', colsample\_bytree=0.5,$$
$$learning\_rate=0.2, max\_depth=15, alpha=5, n\_estimators=150, random\_state=32)$$

f. 贝叶斯分类器参数：

$$BernoulliNB()$$

表一. 各模型五折交叉验证准确率

	1 折	2 折	3 折	4 折	5 折	Average
多层感知机	0.731156	0.770781	0.755668	0.795970	0.740554	0.758826
决策树	0.746231	0.730479	0.745592	0.785894	0.732997	0.748239
逻辑回归	0.748744	0.768262	0.763224	0.780856	0.748111	0.761839
支持向量机	0.738693	0.758186	0.755668	0.768262	0.740554	0.752273
Xgboost	0.791457	0.780856	0.788413	0.826196	0.748111	0.787007
朴素贝叶斯	0.723618	0.758186	0.710327	0.775819	0.740554	0.741701

表二. 各模型五折交叉验证 F1 分数

	1 折	2 折	3 折	4 折	5 折	Average
多层感知机	0.780287	0.810811	0.804829	0.836364	0.782241	0.802906
决策树	0.785563	0.770878	0.792608	0.829659	0.775424	0.790826
逻辑回归	0.793388	0.814516	0.813492	0.827723	0.793388	0.808502
支持向量机	0.793651	0.813230	0.811650	0.823077	0.793587	0.807039
Xgboost	0.823028	0.812903	0.826446	0.855950	0.785408	0.820747
朴素贝叶斯	0.763948	0.799163	0.760915	0.816495	0.783158	0.784736

表三. 各模型五折交叉验证用时

	多层感知机	决策树	逻辑回归	支持向量机	Xgboost	朴素贝叶斯
Time/s	5.77	0.074	0.041	0.232	0.401	0.034

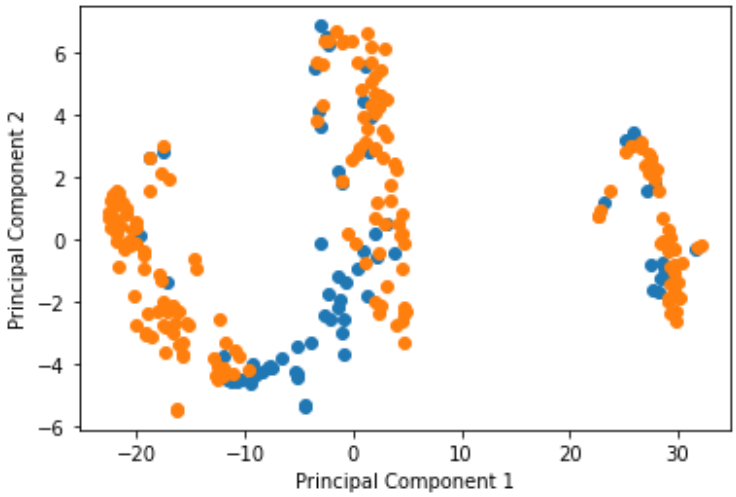
根据以上表格数据，如果只从准确率以及 F1 分数角度出发，则 Xgboost 都是在这些模型中为 top-1 的水平，逻辑回归则排名第二，Xgboost 超过第二名 1-2%。如果考虑训练预测用时，则逻辑回归的训练预测用时为 0.074s，数量级远小于 Xgboost 的 0.401s，所以逻辑回归有能力与 Xgboost 竞争 top-1。

支持向量机、决策树、朴素贝叶斯三种模型总体性能不相上下。但是准确率排名第三的多层感知机并没有占到优势，其相对较长的训练预测时长 5.77s 在数量级上远大于其他任何模型，所以对该模型不予以预测 test 数据集的考虑。综合数据和以上分析，最终决定选择 Xgboost 作为预测模型。

4. Xgboost 进行测试集预测

对于测试集我们并没有 label 标签的具体值，所以在 Xgboost 模型训练后，预测

结果并不能根据定量指标进行评判。在 Xgboost 特征选择为"followers_count","friends_count","listed_count","favourites_count","geo_enabled","verified","statuses_count","lang","is_translator","is_translation_enabled","profile_background_tile","has_extended_profile","default_profile_image","following","translator_type"的情况下，其数据集最重要的两个主成分和 label 属性之间的关系。



图四. T-NSE 降维主成分分析（黄蓝分别代表”human”,”bot”）

五. 实验分析

1. 相关性分析：在第四部分的实验结果可以发现"statuses_count"，“"geo_enabled", "has_extended_profile", "profile_background_tile"都在后续所有模型的特征选择当中无一例外全部被选入。这说明和’label’属性具有相对较强的相关性的特征的选入对模型性能的提升起到关键作用。为了验证这一猜想，我们做出以下实验，以逻辑回归模型为例，将这四个属性分别在所选特征当中剔除，观察其准确率的变化。（详见源码文件第六部分）

表四. 去掉某个属性后准确率的变化

	statues_count	profile_background_tile	has_extended_profile	geo_enalble	NONE
准确率	0.63646	0.74523	0.74724	0.75378	0.761839
相关性	-0.33	0.12	0.21	0.21	—

可以验证，与 label 属性相关性相对很强的属性（如 statues_count 属性），对模型的性能影响起到了至关重要的作用。同样在其他模型中，去掉 statues_count 属性的影响对原准确率的影响大于 10%。

2. 模型性能分析：我们在这一部分着重分析 Xgboost 和多层感知机的性能。

Xgboost 由于是集成模型，一方面可以提高模型的准确率，另一方面其中有并行化处理使得模型训练速度得以大幅度提高，并且鲁棒性好，相对于深度学习不需要精调参数就可以获得与之相媲美的准确率。在该数据集这些特点得到非常好的体现。

然而多层感知机虽然花费开销大，但只要其模型深度和隐藏层的层数赋以合适的值，并对步长、学习率等参数进行微调，理论上应该可以得到准确率是 top-1 的模型，但我们发现并没有，这其实是因为我们的训练数据集不够大，训练集只有不到 2000 的样本量，导致多层感知机的泛化性能并没有得到很好的提升。所以可以采取多轮 epoch 训练的方法，或者过采样的方式缓解这个问题。但是无论如何其过长的训练时间都是无法避免的问题。

3. 训练数据降维分析：在最后利用 Xgboost，再根据特征选择产生数据集，进行训练预测的阶段，我们将该特征选择出的训练集进行了 T-NSE 降维，提取两个主成分，根据图四，实际上此数据的二维降维效果并不好，bot 和 human 两类数据点集高度重叠，这是二维不可线性划分的。说明该数据集即使线性可划分，也只会在更高维度的空间具有线性划分性质。下面我们通 PCA 分析该数据集的主成分：

```
array([9.84680463e-01, 1.89372126e+00, 1.17317456e+01, 6.44318940e+01,
       1.44236493e+02, 1.95435784e+02, 2.75957994e+02, 3.61447481e+02,
       3.89780124e+02, 4.50590668e+02, 4.84220010e+02, 7.31150304e+02,
       2.19364854e+04, 2.41621489e+11, 3.12545336e+12, 1.78706861e+13,
       4.96459428e+13, 9.51543137e+16])
```

图五. PCA 主成分对应的特征值序列

可以认为总共有五个主成分，因为从倒数第五个开始，特征值量级相差 10e7，所以说明只有在五维空间才可以注意到降维成五个主成分的有效性。然而这在低维空间是无法显现出来的。