エンジニアリングデザイン演習 (データマイニング班)
Twitter の bot 判定

195706G 崎濱秀一郎, 195718A 屋嘉比朝温
195744K 金城盛宇, 195767J 大城夏輝

提出日：2022/1/25

目次

1　実験の目的と達成目標

エンジニアリングデザイン演習では、知能情報分野のより専門的な知識の習得や、1,2 年次で習得した知識の理解を一層深めることを目的としている. そこで私たちグループは、Twitter アカウントが人間かボットかを判定をするモデルの構築を通して、機械学習の考え方や精度向上のための技術習得を目標として本実験に取り組んだ.

2　実験方法

2.1　実験目的

昨今の AI 技術の発展に伴い, ボットと人間の区別も難しくなってきた. 特に近年 Twitter では不正ボットの増加が観測されており, 中にはフェイクニュースの拡散やユーザーに対する嫌がらせを働いているボットもある [1]. このようなボットが増え続けると, 今後更なる悪影響を及ぼす可能性がある. そのため, 本実験では不正ボットを判別するための足掛かりとして, 機械学習を用いて,Twitter のアカウントがボットであるか否かを判別することを目標に実験を行なった.

2.2　データセット構築

SIGNATE [2] で公開されているデータセットをダウンロードし, 利用した. 以下の表 1 はデータに含まれているそれぞれの特徴量の説明である. 学習データのサンプル数は 1588 である.

表 1　データに含まれる特徴量と説明

カラム	ヘッダ名称	データ型	説明
0	id	int	インデックスとして使用
1	bot	bit	アカウントの分類 (1=ボット, 0=人間)
2	statuses_count	int	ツイート数
3	default_plofile	bit	プロファイル
4	default_profile_image	bit	プロファイルイメージ
5	friends_count	int	友達数
6	followers_count	int	フォロワー数
7	favourites_count	int	お気に入り数
8	geo_enabled	bit	地理情報の有無 (1=ある, 0=なし)
9	listed_count	int	リスト数
10	account_age_hours	int	年齢
11	diversity	float	多様性
12	mean_mins_between_tweets	float	ツイート間隔の平均時間 (分)
13	mean_tweet_length	float	ツイートの長さの平均値
14	mean_retweets	float	リツイートの平均値
15	reply_rate	float	リプライ率

3　モデル選定

今回使用したデータセットは分類問題であり、分類問題に対して有効であるモデルの精度比較を行うために、ランダムダムフォレスト、ロジスティック回帰、K-nn、LinearSVC の 4 つを選択し実験を行った.

3.1　ロジスティック回帰

3.1.1　前処理

データの基本統計量を確認したところ、値にばらつきがあったため、標準化を行なった. 最初に、特徴量を全て使用し分類を行なったところ、精度が 0.34 だった. 次に、目的変数 (bot) と相関係数の絶対値が 0.1 以上の特徴量を、以下の図 1 のヒートマップで確認し選択したところ、精度が 0.17 と下がってしまった. そこで、目的変数である bot のデータ数に注目した. 図 2 が bot アカウントと bot アカウントでないデータ数の割合を示した図である.bot であるデータの割合は 15.3%、bot でないデータは 84.7% だったことから、このデータセットは不均衡データであると判断した. そこで、不均衡データに対し、有効な手段の一つであるアンダーサンプリングを行い、精度向上を試みた. アンダーサンプリングの方法として、bot のデータ：bot でないデータの割合が 1：2 になるよう、トレーニングデータを選択するといったものである.

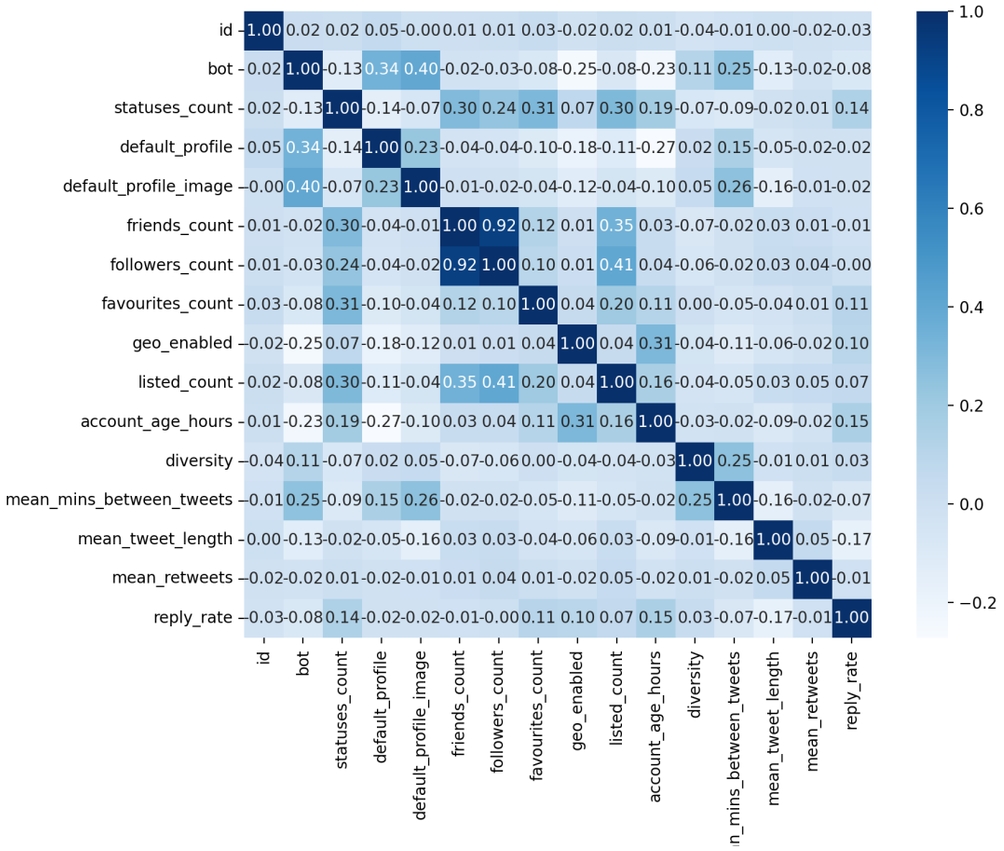

図 1　ヒートマップ

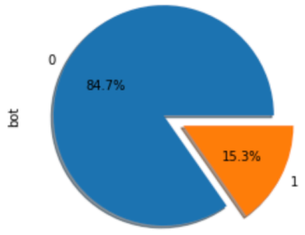

図 2　目的変数 bot(0,1) の割合

3.1.2　特徴量選択

EFS(Exhaustive Feature Search) による特徴量選択を行なった.EFS とは、特徴量の全ての組み合わせを試し、最も性能が高くなった組み合わせの選択を行う Wrapper Method である. 以下が EFS によって選択した特徴量である.

- default_profile_image
- followers_count
- geo_enabled
- mean_mins_between_tweets
- default_plofile
- friends_count
- favourites_count
- listed_count
- mean_tweet_length

3.1.3　パラメーター調整

グリッドサーチによるパラメーター調整を行った. 調整したパラメータは penalty、C、solver の 3 つである.penalty は正則化の方法を指定するもので、C は正則化項の係数を表している.solver は最適解の探索手法を指定するためのパラメータである. 正則化に関するパラメータを選択した理由として、分類の方法に変化があると考えたためである. また、solver を選択した理由は、最適解の探索手法によって、得られる結果に変化があると予想したからである. 調節後のパラメータの値はそれぞれ以下の値となった.

- penalty=l2
- C=1000
- solver=newton-cg

3.2 ランダムフォレスト

3.2.1 前処理

データの前処理として、アンダーサンプリングのみを行い、標準化は行わなかった. その理由として決定木は特徴量の大小関係のみに着目しているため、スケーリングを行う必要がないからである.

3.2.2 特徴量選択

まず、目的変数との相関係数の絶対値が 0.1 以上の特徴量のみを選択し、モデルの構築と精度の評価を行なったが、特徴量選択する前のモデルの精度よりも低い結果となった. そこで、目的変数との相関係数では確認することのできない複数の特徴量との関係があると考え、EFS による特徴量選択と主成分分析を行なった. 結果として、主成分分析よりも EFS で選択した特徴量を使用した方が精度が高くなったため、EFS で選択した以下の特徴量でモデルを構築した.

- default_profile_image
- followers_count
- geo_enabled
- mean_mins_between_tweets
- default_plofile
- friends_count
- favourites_count
- listed_count
- mean_tweet_length

3.2.3 パラメータ調整

グリッドサーチによるパラメータ調整を行なった. 調整したパラメーターは以下の 2 つである.n_estimators は決定木の数であり、数が多くなればなるほど精度が上がるとともに、計算コストが高くなる.max_depth は決定木の深さの制限値を表し、デフォルトでは深さの制限が無く、過学習が起こる可能性があるため調節を行なった. 調節後のパラメータの値はそれぞれ以下の値となった.

- n_estimators=600
- max_depth=7

3.3 K-nn

3.3.1 前処理

前処理として欠損値の確認と標準化、アンダーサンプリング、外れ値処理を行った. 標準化はロジスティック回帰と同様の理由で行った. また、データの分布を見た際に、極端に平均から離れている値を確認したため外れ値処理を行なった. しかし、KNN では未知のデータに対しそこから距離が近い順で取得していくため、外れ値処理による精度向上に影響しなかったため、最終的に標準化とアンダーサンプリングを行なった.

3.3.2 特徴量選択

主成分分析と、目的変数との相関係数の絶対値が 0.1 以上の特徴量のみを選択した 2 つのモデルの構築を行なったが、どちらも精度向上がみられなかったため、EFS を行い特徴量選択を行なった。以下が EFS によって選択した特徴量である.

- statuses_count
- default_plofile
- default_profile_image
- friends_count
- followers_count
- favourites_count
- geo_enabled
- listed_count
- mean_mins_between_tweets
- mean_retweets

3.3.3 パラメーター調整

グリッドサーチによるパラメータ調整を行なった. 調整を行なったパラメータは以下の 6 つである. しかし、調整後のパラメータを用いて構築したモデルの精度が、パタメータ調整前のモデルの精度よりも低くなったため、最終的には以下のデフォルト値でモデル構築を行った.

- n_neighbors=5
- weights='uniform'
- algorithm='auto'
- leaf_size30
- p=2
- metric='minkowski'

3.4 LinearSVC

3.4.1 前処理

標準化とアンダーサンプリングをおこなった.

3.4.2 特徴量選択

相関係数から特徴量の選択を行なったが精度向上につながらなかったため、EFS による特徴量選択を行なった. 以下が EFS によって選択した特徴量である.

- default_plofile
- favourites_count
- listed_count
- mean_tweet_length
- reply_rate
- default_profile_image
- geo_enabled
- diversity
- mean_retweets

3.4.3 パラメーター調整

グリッドサーチによるパラメータ調整を行った. 調整したパラメータは penalty、C、multi_class の 3 つである.penalty はペナルティで使用する基準を指定するパラメータで、C は正則化項の係数を表している.multi_class は y に 3 つ以上のクラスが含まれている場合、マルチクラス戦略を決定するパラメータである. 調節後のパラメータの値はそれぞれ以下の値となった.

- penalty='l2'
- C=0.5
- multi_class='ovr'

4　実験結果

それぞれのモデルの評価結果は以下の表 2 のようになった.

表 2　作成したモデルの評価

モデル	評価値
ランダムフォレスト	0.6909
ロジスティック回帰	0.5677
KNN	0.5575
LinearSVC	0.5285

5　考察

LinearSVC、ロジスティック回帰、k-nn という手法は、境界線や、近くの点によってデータを分類する手法である. 一方、ランダムフォレストは、特徴量を比較する事によってデータを分類する手法で、バギングを用いて決定木モデルを多数作成するアンサンブル学習法である. アンサンブル学習の特徴は、弱い学習器 (決定木) を複数組み合わせる事によって高い精度を得ることができる点である. また、バギングの特徴として、学習データの一部を弱学習器で学習し、最終的な学習器に統合するため、過学習を防ぎ、バリアンスを下げることができる.LinearSVC、ロジスティック回帰、k-nn と比較して、ランダムフォレストはより細かな条件で分類できるため、最も高い精度を得られたと考える.

6　意図していた実験計画との違い

当初の計画では、個人で精度の上がる前処理や特徴量選択の手法を探り、最も精度の高いモデルを選択し実験を行う予定だったが、精度向上までにかなりの時間を要してしまったため、最終的にはそれぞれのモデルで精度向上を図る事となった.

参考文献

[1] Twitter で増加し続ける悪質な不正ボット https://ascii.jp/elem/000/001/743/1743878/

[2] SIGNATE, [練習問題] ボットの判定 https://signate.jp/competitions/124