メルカリ価格予想チャレンジ

神戸電子専門学校
情報処理学科
WebエンジニアIIコース
大西一輝

課題概要

今回テーマにしたものはKaggleのメルカリ価格予測チャレンジというものです。

販売者が投稿した情報をもとに適正な販売価格を予測するというのを行いました。

利用した技術・環境

- Python
- Pandas
- Numpy
- scikit-learn
- Google Colaboratory

販売商品のデータセット

```
[ ] train.head()
```

	train_id	name	item_condition_id	category_name	brand_name	price	shipping	item_description
0	0	MLB Cincinnati Reds T Shirt Size XL	3	Men/Tops/T-shirts	NaN	10.00000	1	No description yet
1	1	Razer BlackWidow Chroma Keyboard	3	Electronics/Computers & Tablets/Components & P...	Razer	52.00000	0	This keyboard is in great condition and works ...
2	2	AVA-VIV Blouse	1	Women/Tops & Blouses/Blouse	Target	10.00000	1	Adorable top with a hint of lace and a key hol...
3	3	Leather Horse Statues	1	Home/Home Décor/Home Décor Accents	NaN	35.00000	1	New with tags. Leather horses. Retail for [rm]...
4	4	24K GOLD plated rose	1	Women/Jewelry/Necklaces	NaN	44.00000	0	Complete with certificate of authenticity

```
[ ] test.head()
```

	test_id	name	item_condition_id	category_name	brand_name	shipping	item_description
0	0	Breast cancer "I fight like a girl" ring	1	Women/Jewelry/Rings	NaN	1	Size 7
1	1	25 pcs NEW 7.5"x12" Kraft Bubble Mailers	1	Other/Office supplies/Shipping Supplies	NaN	1	25 pcs NEW 7.5"x12" Kraft Bubble Mailers Lined...
2	2	Coach bag	1	Vintage & Collectibles/Bags and Purses/Handbag	Coach	1	Brand new coach bag. Bought for [rm] at a Coac...
3	3	Floral Kimono	2	Women/Sweaters/Cardigan	NaN	0	-floral kimono -never worn -lightweight and pe...
4	4	Life after Death	3	Other/Books/Religion & Spirituality	NaN	1	Rediscovering life after the loss of a loved o...

データの情報

train_id / test _id – ユーザー投稿のID

name – 投稿のタイトル。タイトルに価格に関する情報がある場合（例：$20）はメルカリが事前に削除をして[rm]と置き換えている

item_condition_id – ユーザーが指定した商品の状態

category_name – 投稿カテゴリー

brand_name – ブランドの名前

price – 訓練データのみ。実際に売られた価格。米ドル表示。

shipping – 送料のフラグ。「1」は販売者負担。「0」は購入者負担。

item_description – ユーザーが投稿した商品説明の全文。価格情報がある場合は[rm]と置き換えられている。

testのpriceを予測します。

目標

3ページ目のデータセットを確認すると「価格(Price)」の項目がテストデータに含まれていません。

ランダムフォレストというのを使ってテストデータの「価格(Price)」を予測していきます。

欠損データ確認1

```
# trainの欠損データの個数と%を確認
train.isnull().sum(),train.isnull().sum()/train.shape[0]
```

```
(train_id             0
 name                 0
 item_condition_id       0
 category_name        6327
 brand_name         632682
 price                   0
 shipping                0
 item_description        4
 dtype: int64, train_id          0.00000
 name               0.00000
 item_condition_id   0.00000
 category_name       0.00427
 brand_name         0.42676
 price              0.00000
 shipping           0.00000
 item_description    0.00000
 dtype: float64)
```

```
[ ]  # testの欠損データの個数と%を確認
test.isnull().sum(),test.isnull().sum()/test.shape[0]
```

```
(test_id             0
 name                0
 item_condition_id       0
 category_name        3058
 brand_name         295525
 shipping                0
 item_description        0
 dtype: int64, test_id          0.00000
 name               0.00000
 item_condition_id   0.00000
 category_name       0.00441
 brand_name         0.42622
 shipping           0.00000
 item_description    0.00000
 dtype: float64)
```

欠損データ確認2

６ページ目を確認するとカテゴリ名（category_name）とブランド名（brand_name）の欠損数が大きいことが確認できた。特にブランド名の欠損度合いはかなり大きいと見える。

データ事前処理

ランダムフォレストのモデルを作成しました。そのために以下の手順を踏みました。

1. trainとtestのデータを連結させる
2. 連結させたDataFrameの文字列のデータ形式を「cateogry」へ変換
3. 文字列を数値へ値を変換
4. 訓練用データの「price」をnp.log()で処理
5. ランダムフォレスト用にxとy（ターゲット）で分ける

ランダムフォレストのモデル作成

処理したデータをもとにscikit-learnのRandomForestRegressorでモデルを作りました。

スコアは0.7400987967569919とあまり高くなかった。

実際の予測値

testデータの予測価格は以下の通りです。

	test_id	price
0	0	12.88961
1	1	7.75365
2	2	19.99445
3	3	14.85366
4	4	8.02702

結論

今回はランダムフォレストを使って予測をしたがあまりいいスコアは出なかった。

あまり理解できていないので学習を深めればスコアは伸ばせると思う。

決定木を使っての予測も試してみたい。