機械学習シリーズ：ロジスティック回帰

原文はこちら

ロジスティック回帰とは？

ロジスティック回帰は、バイナリ分類に使用される一般的な機械学習アルゴリズムです。ロジスティック回帰は、サンプルに 2 つの可能なクラスのうちの 1 つを使用してサンプルにラベル付けします。オプションで、出力は、サンプルが指定されたクラスに属する確率とすることができます。

たとえば、研究者が、学生が大学院に受け入れられるか受け入れられないかどうかを決定する要素に興味があるとします。応答はバイナリ、受け入れるか受け入れないとします。研究者は、学部での GPA、GRE のスコア、および職務経験などの要因を検討する可能性があります。

標準のロジスティック関数は、以下のように定義されます。

$$y = \frac{1}{1 + e^{-x}} \qquad (1)$$

ここで、y は従属変数を表し、x は独立変数であり、次のように定義されます。

$$x = a_0 + a_1 x_1 + a_2 x_2 + \ldots \qquad (2)$$

この例では、従属変数はバイナリ応答（受け入れる、あるいは、受け入れない）であり、独立変数は考慮するすべての要素で構成されています。

Vertica におけるロジスティック回帰のトレーニングは、2 段階のプロセスです。

1. モデルの構築 – トレーニングデータセットは、上記の式（2）で与えられた係数の最良の値を見つけるために使用されます。
2. モデルの評価 – ステップ 1 で作成したモデルをテストデータセットに適用して、独立変数に基づいて従属変数を予測します。予想される出力を実際の値と比較して、予測がどれほど良いかを評価することができます。

基本を説明しましたので、ロジスティック回帰をよりよく理解するための例を見ていきましょう。

タイタニックデータセット

1912 年 4 月 15 日の早朝、タイタニック号は氷山と衝突して大西洋に悲劇的に沈みました。この例では、タイタニック号の乗客情報を含むデータセットを使用します。

注意: 使用したデータは Vanderbilt Department of Biostatistics site より入手可能です。

データセットは次の情報を含んでいます。

- pclass: 乗客クラス（1 等、2 等、3 等）
- survival: 0 = いいえ; 1 = はい
- name: 乗客の名前
- sex: 乗客の性別
- age: 年齢
- sibsp: 海外の兄弟/配偶者の数
- parch: 海外の親/子供の乗車人数
- ticket: チケット番号
- fare: 旅客運賃
- cabin: キャビン番号
- embarked: 乗車港, C = Cherbourg; Q = Queenstown, S = Southampton
- boat: 救命艇番号
- body: ボディ番号
- home.dest: ホームまたは目的地

このデータセットを使用して、乗客クラス、年齢、性別などの他の列に基づいて生存率（0 または 1 のいずれか）を予測します。これを行うために、元のデータセットをサンプルの 70%と 30%にそれぞれ「train」と「test」という 2 つのランダムに選択されたテーブルに分割します。

- train: このテーブルは、ロジスティック回帰モデルを生成するために使用されます。
- test: モデルが生成された後、このテーブルを使用してモデルの精度をチェックします。

データの読み込み

次のテーブル定義を使用してデータセットをロードしましょう。

```
=> CREATE TABLE public.titanic (
pclass integer,
survived integer,
name varchar(96),
gender integer,
age numeric(6,3),
sibsp integer,
parch integer,
ticket varchar(24),
fare numeric(7,4),
cabin char(10),
embarked integer,
boat char(4),
body integer,
homedest varchar(64),
split float DEFAULT RANDOM()
) ;
```

元のデータセットのすべての列が、split という名前の追加の列とともにテーブル定義に表示されます。split 列は、データセットを「train」と「test」に分けるために使用されます。

ダウンロードしたファイルの名前が「titanic3.csv」であるとします。COPY コマンドを使用してこのデータをテーブルにロードするには、最初に空の二重引用符をバックスラッシュと二重引用符で置き換える必要があります。これは任意のテキストエディタを使用して行うことができますが、次の Linux コマンドを使用することもできます。

```
sed -i 's/""/\\"/g' titanic3.csv
```

ファイルからテーブルにデータをロードするには、COPY コマンドを使用します。

```
=> \set titanic_file '\' absolute path/titanic3.csv\''
=> COPY public.titanic (
pclass, survived, name,
fgend FILLER varchar(6),
gender AS DECODE (fgend,'female',0,'male,1),
age, sibsp, parch, ticket, fare, cabin,
femb FILLER char(1),
embarked AS DECODE(femb,'C',1,'Q',2,'S',3),
boat, body, homedest)
FROM :titanic_file
DELIMITER ','
ENCLOSED BY '"'
ESCAPE AS '\'
SKIP 1
ABORT ON ERROR
DIRECT ;
```

ロジスティック回帰は、係数を計算する際に数値を入力として使用するだけなので、上の COPY 文でデータ変換を実行しました。この変換により、gender と embarked の両方のテキストが数値に置き換えられました。

欠損値

ロジスティック回帰アルゴリズムは欠損値を受け入れません。予測因子の 1 つが NULL の場合、行全体が破棄されます。

元のデータセットには、age 列にいくつかの NULL 値が欠落しています。行全体の除外を避けるために、同じ class の同じ gender である人の平均年齢に NULL 年齢を置き換えます。

この置換は、「train」と「test」テーブルを作成すると同時に実行されます。

まず、train テーブルを作成しましょう。

```
=> CREATE LOCAL TEMPORARY TABLE train
ON COMMIT PRESERVE ROWS AS /*+DIRECT*/
SELECT
pclass,
survived,
```

```
gender,
COALESCE(age, avg_age) AS age,
(sibsp + parch + 1) AS family_size,
fare, embarked
FROM (
SELECT
pclass,
survived,
gender,
age,
sibsp,
parch,
fare,
embarked,
AVG(age) OVER(
PARTITION BY pclass,gender)
AS avg_age,
split
FROM public.titanic )
x
WHERE x.split < 0.7;
```

続いて、test テーブルを作成します。

```
=> CREATE LOCAL TEMPORARY TABLE test
ON COMMIT PRESERVE ROWS AS /*+DIRECT*/
SELECT
pclass,
survived,
gender,
COALESCE(age, avg_age) AS age,
(sibsp + parch + 1) AS family_size,
COALESCE(fare,avg_fare) AS fare,
COALESCE(embarked,3) AS embarked
FROM (
SELECT
pclass,
survived,
gender,
age,
sibsp,
```

```
parch,
fare,
embarked,
AVG(age) OVER(
PARTITION BY pclass,gender)
AS avg_age,
AVG(fare) OVER(
PARTITION BY pclass, gender)
AS avg_fare,
split
FROM public.titanic )
x
WHERE x.split >= 0.7 ;
```

覚えておくべき重要なポイントは次の通りです。

- ロード段階で生成された split 列を使用して、元のデータセットを分割しました。
- ロジスティック回帰は数値列のみを使用するため、2 つのテーブルを作成するときに非数値列をすべてスキップしました。簡単にするために、この例では数値以外の列は扱いません。
- 家族のサイズが生存率に影響を与えるかどうかを判断するために、sibsp と patch の組み合わせである family_size 列を作成しました。

ロジスティック回帰モデルの作成

さて、Vertica の機械学習関数を使って、モデルを作成しましょう。モデルを作成するには、train テーブルで LOGISTIC_REG 関数を使用します。Survived は従属変数であり、独立変数は pclass、age、gender、family_size、fare、および embarked です。

```
=> SELECT LOGISTIC_REG(
'titanic',
'v_temp_schema.train',
'survived',
'pclass,age,gender,family_size,
fare,embarked') ;
```

モデルの生成後、test テーブルで PREDICT_LOGISTIC_REG 関数と ERROR_RATE 関数を使用して、予測値と実際の値を比較し、エラーレートを出します。

```
=> SELECT ERROR_RATE (
       obs, predict::int USING PARAMETERS num_classes=2)
OVER()
FROM (
SELECT survived AS obs, PREDICT_LOGISTIC_REG (
pclass, age, gender,
family_size,
fare, embarked
USING PARAMETERS MODEL_NAME='titanic',
OWNER='dbadmin')
```

```
AS predict
FROM
test ) AS prediction_output ;
```

結果は 0.17 であり、このモデルにおいて、82%以上のケースで生存の値が正しく予測されたことを意味します。

age や passenger class 別に予測精度をグループ化することもできます。

```
=> SELECT
pclass,
CASE WHEN gender = 1 THEN 'male'
ELSE 'female' END AS gender,
(100 * SUM(
CASE WHEN survived = predict
THEN 1 ELSE 0 END) / COUNT(*)
)::NUMERIC(5,2) as accuracy_perc
FROM (
SELECT
pclass,gender,survived,
PREDICT_LOGISTIC_REG (
pclass,
age,
gender,
family_size,
fare,
embarked
USING PARAMETERS OWNER='dbadmin',
MODEL_NAME='titanic') AS predict
FROM
test) AS prediction_output
GROUP BY pclass, gender
ORDER BY accuracy_perc DESC
;
```

注意: Vertica 8.1 では、OWNER パラメーターは、今後廃止予定となります。

```
 pclass | gender | accuracy_perc
--------+--------+---------------
1       | female | 95.00
3       | male   | 91.78
2       | male   | 86.21
2       | female | 84.38
```

```
1        | male   | 68.09
3        | female | 59.32
```

この結果から、1 等の女性は最も正確に予測され、一方、3 等の女性は最も正確に予測されなかったことがわかります。

参考文献

[1] How to Perform a Logistic Regression in R - datascience+
[2] Titanic: Machine Learning from Disaster - Kaggle
[3] Titanic data set at biostat.mc.vanderbilt.edu

詳細については、Vertica ドキュメントの Machine Learning for Predictive Analytics を参照してください。