Tutorial_titanic

2019 年 4 月 9 日

1　データの前処理

In [1]:
```
# 適当に必要になりそうなモジュールをインポート
import numpy as np
import pandas as pd
```

In [2]:
```
# データの読み込み（kaggle の titanic competition から自分で train.csv test.csv gender_submission.
train=pd.read_csv('train.csv') # 訓練用データ
test=pd.read_csv('test.csv') # テスト用データ
```

In [3]:
```
# 欠損値確認
# train data
train.isnull().any(axis=0)
```

Out[3]:
```
PassengerId    False
Survived       False
Pclass         False
Name           False
Sex            False
Age             True
SibSp          False
Parch          False
Ticket         False
Fare           False
Cabin           True
Embarked        True
dtype: bool
```

In [4]:
```
# test data
test.isnull().any(axis=0)
```

Out[4]:
```
PassengerId    False
Pclass         False
Name           False
Sex            False
Age             True
```

```
SibSp          False
Parch          False
Ticket         False
Fare            True
Cabin           True
Embarked       False
dtype: bool
```

```
In [5]: # 欠損値補完--> Age Embarked Fare
        train['Age']=train['Age'].fillna(train['Age'].mean()) # mean は平均値
        test['Age']=train['Age'].fillna(train['Age'].mean())
        train['Embarked']=train['Embarked'].fillna('S')
        test['Embarked']=train['Embarked'].fillna('S')
        train['Fare']=train['Fare'].fillna(train['Fare'].mean())
        test['Fare']=test['Fare'].fillna(test['Fare'].mean())
```

```
In [6]: # labeling--> embarked
        embarked_mapping={'C':3,'S':2,'Q':1}
        train['Embarked']=train['Embarked'].map(embarked_mapping)
        test['Embarked']=test['Embarked'].map(embarked_mapping)
```

```
In [7]: # labeling --> Sex
        sex_mapping={'male':1,'female':0}
        train['Sex']=train['Sex'].map(sex_mapping)
        test['Sex']=test['Sex'].map(sex_mapping)
```

```
In [8]: # delete columns--> Name Ticket Cabin
        train=train.drop(['Name','Ticket','Cabin'],axis=1)
        test=test.drop(['Name','Ticket','Cabin'],axis=1)
```

```
In [9]: # 整形したデータから、訓練用特徴行列:X_train、訓練用クラスラベル:y_train、テスト用特徴行列:x_test を
        # test.csv には  survived（クラスラベル）がないことに注意
        x_train,x_test=train.iloc[:,2:],test.iloc[:,1:]
        y_train=train.iloc[:,1]
```

2 モデル予測

2.1 複数モデルの作成

```
In [10]: # 使用する予測モデルモジュールのインポート
         # Logistic 回帰  決定木  K 近傍法
         # 標準化を行うのでそれもインポート
         # 訓練用データのなかでさらにデータを分割して交差検定を行うのでそれもインポート
         from sklearn.model_selection import cross_val_score
         from sklearn.linear_model import LogisticRegression
         from sklearn.tree import DecisionTreeClassifier
```

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
```

```
In [11]: # ハイパーパラメータを引数に各モデルのクラスをインスタンス化（各モデルを明示化するってことです）
clf1=LogisticRegression(penalty='l2',C=0.001,random_state=1)
clf2=DecisionTreeClassifier(max_depth=1,criterion='entropy',random_state=0)
clf3=KNeighborsClassifier(n_neighbors=1,p=2,metric='minkowski')
```

```
In [12]: # 標準化する工程をくっつけたモデルを作成（pipeline でできる）
pipe1=Pipeline([['sc',StandardScaler()],['clf',clf1]])
pipe3=Pipeline([['sc',StandardScaler()],['clf',clf3]])
```

2.2 最適モデルの探索

```
In [13]: # 各モデルの精度評価
clf_labels=['Logistic regression','Decision tree','KNN']
print('10-fold cross validation:\n')
for clf, label in zip([pipe1,clf2,pipe3],clf_labels):
    scores=cross_val_score(estimator=clf,X=x_train,y=y_train,cv=10,scoring='roc_auc')
    print("ROC AUC: %0.2f(+/- %0.2f) [%s]"%(scores.mean(),scores.std(),label))
```

```
10-fold cross validation:

ROC AUC: 0.84(+/- 0.03) [Logistic regression]
ROC AUC: 0.77(+/- 0.04) [Decision tree]
ROC AUC: 0.74(+/- 0.05) [KNN]


C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
  FutureWarning)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:401: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
  FutureWarning)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:401: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
```

```
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
  FutureWarning)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:401: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
  FutureWarning)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:401: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
  FutureWarning)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:401: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
  FutureWarning)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:401: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
  FutureWarning)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:401: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
```

```
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
  FutureWarning)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:401: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
  FutureWarning)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:401: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
  FutureWarning)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:401: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:381: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:381: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:381: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
```

```
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:381: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:381: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:381: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:381: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:381: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:381: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:381: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
```

2.3 test data を予測してみる

```
In [14]: # いまのところ精度が一番いい Logistic 回帰モデルでテストデータを予測してみる
         pipe1.fit(x_train,y_train)
         y_pred=pipe1.predict(x_test)
```

```
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\preprocessing\data.py:625: DataConversionWarning:
  return self.partial_fit(X, y)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\base.py:465: DataConversionWarning: Data with inp
  return self.fit(X, y, **fit_params).transform(X)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Defa
  FutureWarning)
C:\Users\taiki\Anaconda3\Lib\site-packages\sklearn\pipeline.py:331: DataConversionWarning: Data with
  Xt = transform.transform(Xt)
```

3 予測データの csv 作成（Kaggle submit 用）

```
In [15]: # submit to kaggle titanic competition
         sub=pd.DataFrame(pd.read_csv('test.csv')['PassengerId']) # test.csv の PassengerID のみ新しいテ
         sub['Survived']=y_pred.tolist() # Survived コラムに先ほど予測したラベルリストを入力 (y_pred は行列
         sub.to_csv('submission.csv',index=False) # submission.csv として保存（titanic competition でし
```

```
In [16]:
```