

We are TECH.C FUKUOKA 2023
ITワールドプレゼンテーション

自転車レンタル数の予測

2年 AIクリエイター専攻
ブリゲッタ ニコル

自己紹介

2年 AIクリエイター専攻
ブリゲッタ ニコル

シンガポール出身

趣味：音楽を聞くこと
ゆったりゲームをすることです



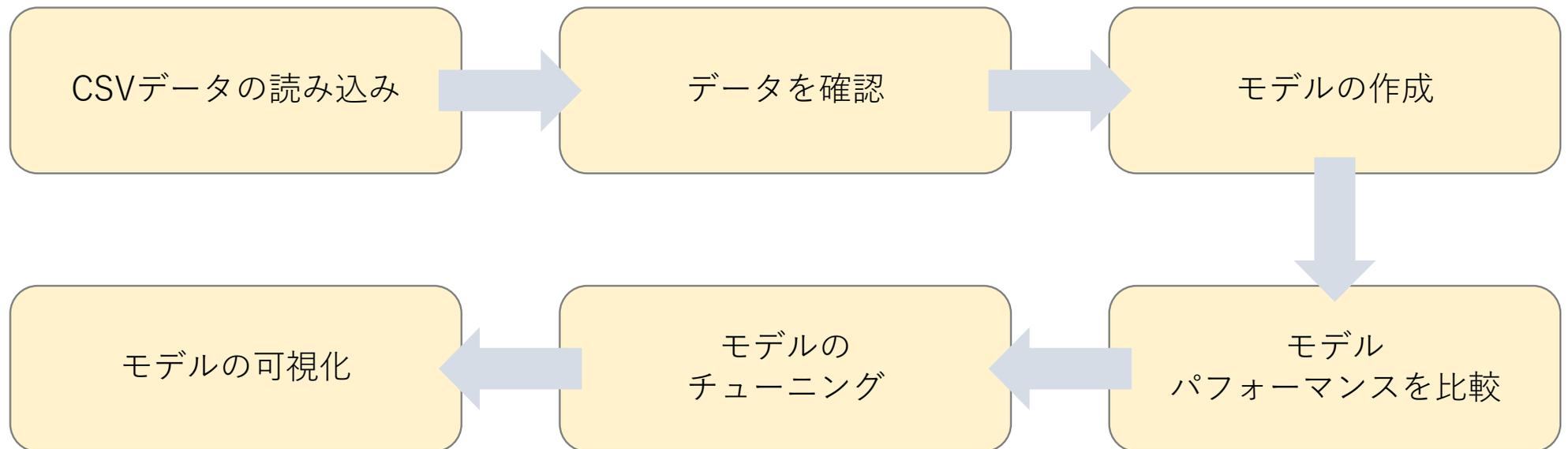
アジェンダ

- 概要の説明
- 開発の目的
- データ収集
- モデルの構築作業
- 完成品
- 気づきや感想

概要の説明

- 回帰モデルを使用して、自転車レンタル数を予測しました

モデル構築の手順：



開発の目的

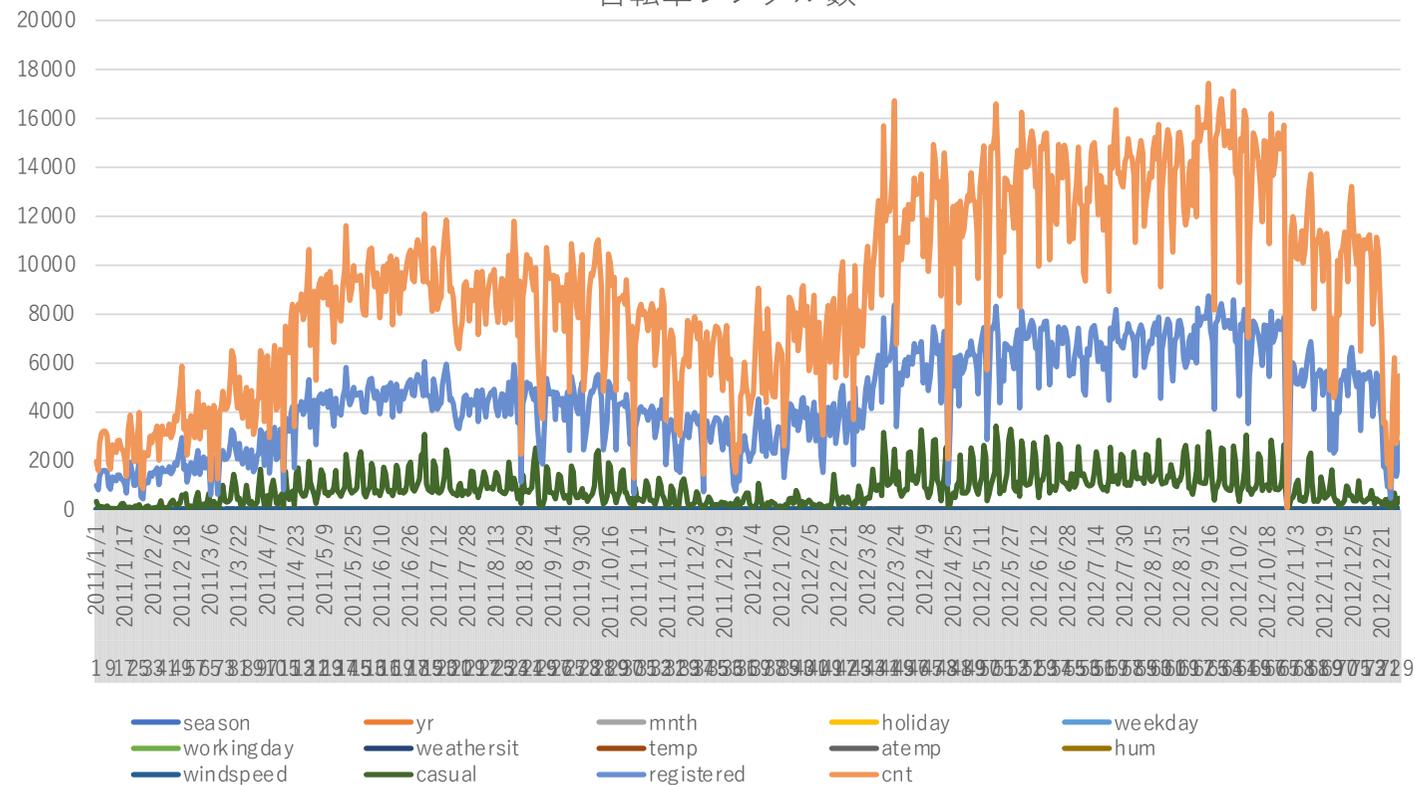
- 天候と季節によって、自転車レンタル数がどう変わるのを知りたい
- 必要なデータだけを使用して、機械を学習させる
- 学習済みのモデルでテスト行う
- 散布図でモデルの可視化にする



データ収集

• 2011~2012年のCSVデータ

自転車レンタル数



<https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset>

モデルの構築作業

データの読み込み作業と内容確認

```
import pandas as pd
import numpy as np
import zipfile

# df = pd.read_csv(r'ダウンロード\data.csv')
df = pd.read_csv('day.csv')
df
```

→ データ読込
→ 内容確認コード

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	ate	
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.3631
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.3531
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.1891
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.2121
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.2291
...
726	727	2012-12-27	1	1	12	0	4	1	2	0.254167	0.2261
727	728	2012-12-28	1	1	12	0	5	1	2	0.253333	0.2551
728	729	2012-12-29	1	1	12	0	6	0	2	0.253333	0.2421
729	730	2012-12-30	1	1	12	0	0	0	1	0.255833	0.2311
730	731	2012-12-31	1	1	12	0	1	1	2	0.215833	0.2231

モデルの作成とパフォーマンス比較

```
from sklearn.model_selection import train_test_split
# df = pd.get_dummies(df, drop_first=True)
# X = df.loc[:, df.columns!='cnt']
# y = df['cnt']
X = df.iloc[:, :-1]
y = df.iloc[:, -1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
```

```
from sklearn import tree
# from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import RandomForestRegressor
model = RandomForestRegressor(random_state=1)
model.fit(X_train, y_train)
# model.predict(X_test)
```

RandomForestRegressor(random_state=1)

```
# trainのr2_scoreを求める
from sklearn.metrics import r2_score
y_pred = model.predict(X_train)
print(r2_score(y_train, y_pred))
```

0.9819674134293361

```
# testのr2_scoreを求める
y_pred = model.predict(X_test)
print(r2_score(y_test, y_pred))
```

0.8921095534295026

モデルの作成

パフォーマンス比較

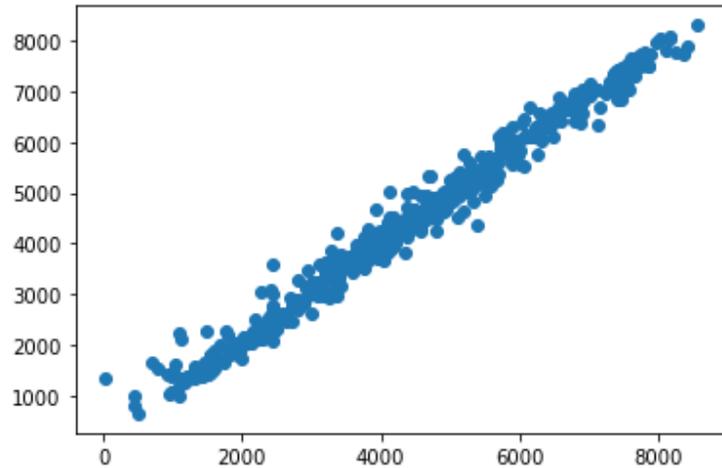
完成品

訓練データのグラフ

```
In [19]: # trainのr2_scoreを求める
y_pred = model2.predict(X_train)
print(r2_score(y_train, y_pred))
```

0.979726269551416

```
In [20]: import matplotlib.pyplot as plt
plt.scatter(y_train, y_pred)
plt.show()
```

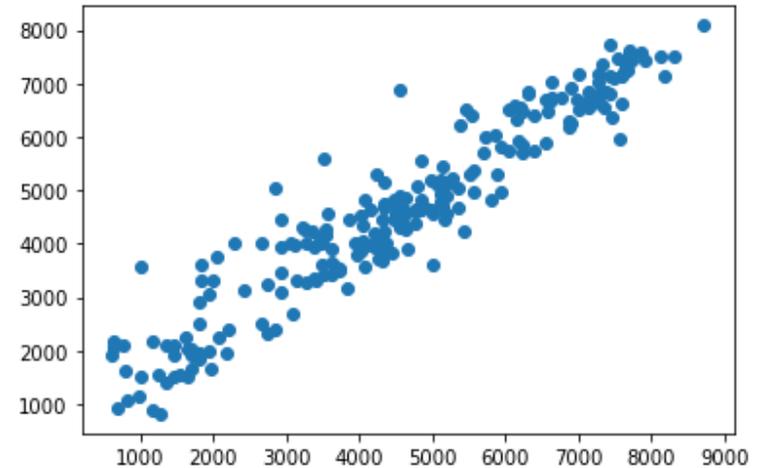


テストデータのグラフ

```
In [21]: # testのr2_scoreを求める
y_pred = model2.predict(X_test)
print(r2_score(y_test, y_pred))
```

0.8903323137035112

```
In [22]: import matplotlib.pyplot as plt
plt.scatter(y_test, y_pred)
plt.show()
```



気づきや感想

- モデルの構築の手順を学び出来ました
- モデル作成した時、講師の先生に色々なアドバイスを頂いて、最後までモデルを完成できました
- 慣れていなかったコードなどは少しずつ慣れてきました
- アルゴリズム定義の大切さ
 - 予測モデルの結果が変わる
- パフォーマンス比較の大切さ
 - モデルのデータによって、グラフの結果が違ってくる

ご清聴ありがとうございました！