知能情報実験 III（データマイニング班）
人工知能を用いて那覇の天気を予測する

195757B 口石祥聖, 195752A 上原蒼矢, 195345B 黒澤一希

提出日：2021 年 8 月 24 日

目次

1　概要

本文書は, 知能情報実験 III(データマイニング班) のグループ 3 が行った実験をまとめた最終レポートである. 本グループでは, 沖縄県那覇市における 3 時間後の天気を高い精度で予測することを目標に実験を行ってきた. 本実験では sikit-learn のチート紙とを利用して Linear-SVC と AR モデルを学習モデルとして選んだ. 実験を通して, Linear-SVC は精度を求めてコードを改善することができたが, AR モデルに関しては途中からの実験ということで精度を求めるところで終わってしまった. 今後は AR モデルに関してコードの見直し, 修正を行なっていきたい.

2　はじめに

2.1　実験の目的と達成目標

本グループでは, 機械学習を用いて那覇における 3 時間後の天気を予測することを対象問題として設定した. これまでの天気予測は, 予報官の経験や知識に基づいて過去のデータを照らし合わせることで行ってきた [1]. しかし, より高い精度で天気の予測を行う場合, 人間だけの能力で行うには限界がある. よって, 本グループは機械学習を用いて精度の高い天気予報を行えるのではないかと考えた.

3　実験方法

実験のために作成したコードは github で, 公開している.
`https://github.com/e195345/stulab_dm`

3.1　実験目的

Python を用いて天気や気候に関するデータを分析することで, 沖縄県那覇市における天気予報の精度上昇を目的として実験に取り組んだ.

3.2　データセット構築

気象庁 [2] に保管されているデータを元にデータセットを作成した. 沖縄県那覇市について 2020 年 1 月 1 日から 2021 年 1 月 1 日までの年月日時, 気温, 天気, 降水量, 雲量, 湿度の 1 時間ごとのデータをダウンロードした. 図 1 は, 実際に作成したデータセットである. また, 前半 80% をトレーニングデータとし, 後半 20% をテストデータとした.

- 説明変数
  - 気温　:flort 型,1 時間毎の気温が記録されている.
  - 降水量:flort 型,1 時間毎の降水量が記録されている.
  - 雲量　:flort 型,3 時間毎の雲の量が 12 段階で記録されている.
    12 段階の中に 0+ と 10-とあるのだが, それぞれを 0.5 と 9.5 に変更している.
  - 湿度　:flort 型,1 時間毎の湿度が記録されている.
- 目的変数
  - 天気　:flort 型,3 時間毎にそれぞれの天気に対応した値が記録されている.
    データをダウンロードした時から, 天気は値で表示されていたため, そのまま使用する.
    値に対応している天気は, 気象庁のデータについて [3] で確認できる.

```
2020/1/1 1:00:00 ~ 2021/1/1 24:00:00までのデータ
年月日時,気温,降水量,雲量,湿度,天気
2020/1/1 1:00:00,16.4,0.0,,59.0,
2020/1/1 2:00:00,16.2,0.0,,56.0,
2020/1/1 3:00:00,16.7,0.0,9.5,63.0,4.0
2020/1/1 4:00:00,16.4,0.0,,58.0,
2020/1/1 5:00:00,16.0,0.0,,57.0,
2020/1/1 6:00:00,16.1,0.0,9.5,61.0,4.0
2020/1/1 7:00:00,16.1,0.0,,58.0,
2020/1/1 8:00:00,15.9,0.0,,56.0,
2020/1/1 9:00:00,16.1,0.0,9.5,57.0,4.0
2020/1/1 10:00:00,16.5,0.0,,57.0,
2020/1/1 11:00:00,17.6,0.0,,51.0,
2020/1/1 12:00:00,17.7,0.0,9.5,53.0,4.0
```

図 1　3 時間後の天気を予測させた場合の混合行列

3.3　データの前処理

前処理は 3 種類行った. 1 つ目は nan の入っている行を削除した. 2 つ目は時刻 t における気温を 気温$_t$ とすると,気温$_t$ を (気温$_t$ − 気温$_{(t-1)}$) に書き換えた. 3 つ目は天気以外の特徴量に対して標準化を行なった.

3.4　モデル選定

本実験は天気を予測することが目的のため, 分類問題にあたる. よって, 学習アルゴリズムは Linear-SVC を使用した. 選定方法は, scikit-learn チートシート [5] を利用した. また, 気象の変化は突然起こるのではなく, 連続的に変化するという点が, 回帰モデルを使用するのに適していると考え, 一つ前のデータとの差を利用して学習を行う自己回帰モデルも使用することにした.

3.4.1 自己回帰モデルについてのアプローチ

自己回帰モデルを実装する際に, scikit-learn には自己回帰モデルがなかったため, 自己回帰モデルが多変数の線形回帰モデルと同一視できることを活かし, 今回は scikit-learn 内の LinearRegression モデルを使用して実験を行った.[6] しかし, 結果が思わしくなかったため, 他のモデルの利用について考えた. この際, StatsModels というライブラリに自己回帰モデルが用意されていることを知ったため StatsModels 内の線形回帰モデルである OLS モデルと AR モデルを利用してプログラムを作成することを考えた. AR モデルを使用したものについては, 作成が間に合わなかったため結果は省略する.

3.5 パラメータ調整

3.5.1 Linear-SVC

パラメータの調整は, GridSearchCV でパラメータの最適化を行った. max_iter の値をデフォルトの値 1,000 から 100,000,000 に調整した. dual をデフォルトの Ture から False に調整した. penalty をデフォルトの l2 から l1 に調整した. C をデフォルトの 1.0 から 3.6 に調整した.

3.5.2 LinearRegression

fit_intercept は目的変数が原点を必ず通る性質のデータのときに利用するパラメータである. 今回の実験では, 切片の計算結果を必要としないため False とした. その他はデフォルト値で使用した.

copy_X=True, n_jobs=None, positive=False

3.5.3 OSL

OSL については全てデフォルト値である.

exog = None, missing = 'none', hasconst=None, **kwargs

3.6 モデルの評価方法

3.6.1 Linear-SVC

Linear-SVC モデルの精度は, 全てのテストデータについて正確に予想ができたかどうかで判断し, 数字が 1 に近いほど評価が高くなる.

$$\text{正答率} = \frac{\text{正しく分類できたテストデータの数}}{\text{テストデータ全体の数}}$$

3.6.2 LinearRegression

LinearRegression モデルの評価は, 決定係数を用いた. 以下の式の y に目的変数の平均値, y' に予測値が入る. これは 0 から 1 の範囲の値をとり, この値が 1 に近づくほどモデルの性能が高いことになる.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(yi - y'i)^2}{\sum_{i=1}^{n}(yi - y)^2}$$

3.6.3 OSL

OLS については, LinearRegression と同様であるため省略する.

4 実験結果

4.1 Linear-SVC

2.3 で示した 3 種類の前処理を全て行って学習させた.
今の天気を予測させた場合の正答率は, 81.712% であった.
混合行列は図 2 の通りである. 晴天と晴れは正しく予測できているが, 雨を曇と予測している割合が高いことがわかった.

	晴天	晴れ	薄曇	曇	雨
晴天	47	0	0	0	0
晴れ	0	150	0	0	0
薄曇	0	0	0	8	0
雲	0	0	0	205	0
雨	0	0	0	86	18

図 2　今の天気を予測させた場合の混合行列

次に, 教師データを 1 つ前にずらして,3 時間後の天気を予測させた場合の正答率は,55.75% であった. 混合行列は図 3 の通りである. ほとんどのテストデータを晴れもしくは曇と分類していることがわかった.

	晴天	晴れ	薄曇	曇	雨
晴天	0	39	0	8	0
晴れ	0	102	0	48	0
薄曇	0	1	0	6	0
雲	0	23	0	181	1
雨	0	5	0	96	3

図 3　3 時間後の天気を予測させた場合の混合行列

4.2　LinearRegression

2.3 節で示した 3 種類の前処理を適用し, LinearRgression を用いて実験を行った結果は表 1 のようになっている. 結果の信憑性を示すために平均二乗誤差の結果も掲載している.
説明変数の係数：-0.28294

	テストデータ	トレーニングデータ
平均二乗誤差	4.2654	4.8581
決定係数	0.42558	0.42110

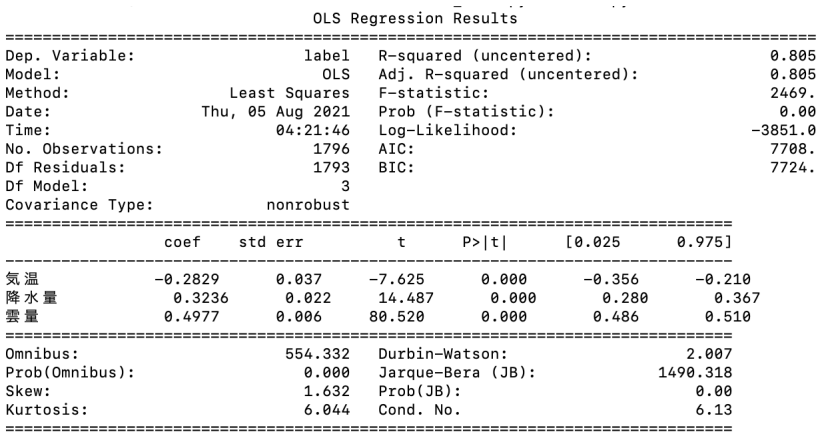
表 1　LinearRegression の結果

4.3　OLS

2.3 節で示した 3 種類の前処理を適用し, OLS を用いて実験を行った結果は表 2 と図 4 のようになっている.OLS については平均二乗誤差を求めることができなかったので結果の記入は省略する.

	テストデータ	トレーニングデータ
前処理なし	0.79811	0.79851
前処理あり	0.80170	0.80512

表 2　OLS を用いた場合の決定係数



OLS Regression Results

Dep. Variable:	label	R-squared (uncentered):	0.805
Model:	OLS	Adj. R-squared (uncentered):	0.805
Method:	Least Squares	F-statistic:	2469.
Date:	Thu, 05 Aug 2021	Prob (F-statistic):	0.00
Time:	04:21:46	Log-Likelihood:	-3851.0
No. Observations:	1796	AIC:	7708.
Df Residuals:	1793	BIC:	7724.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
気温	-0.2829	0.037	-7.625	0.000	-0.356	-0.210
降水量	0.3236	0.022	14.487	0.000	0.280	0.367
雲量	0.4977	0.006	80.520	0.000	0.486	0.510

Omnibus:	554.332	Durbin-Watson:	2.007
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1490.318
Skew:	1.632	Prob(JB):	0.00
Kurtosis:	6.044	Cond. No.	6.13

図 4　OLS の結果

5　考察

5.1　Linear-SVC

まず, 今の天気を予測させた場合の考察である. 実験結果から, 雨を曇と分類している割合が 8 割を超えており, かなり多いことがわかった. データセットを確認すると天気が雨の時は 343 回あり, その中で降水量が 0.0mm の場合は 178 回, 降水量が 1.0mm の場合は 257 回となっており, 7 割以上が降水量 1.0mm 以下ということがわかった. 反対に, 天気が曇の時は 904 回あり, その中で降水量が 0.0mm より多い場合は 28 個で, ほとんどの場合で降水量 0.0mm ということがわかる. よって, 雨を曇と分類してしまう原因は, 降水量にあると考える.

次に,3 時間後の天気を予測させた場合の考察である. 精度がかなり悪くなったため, 正しく教師データをずらせているかを確認したが, 問題なかった. よって, データセットの数が足りなかったと思われる.

5.2　LinearRegression

今回の結果から, 平均二乗誤差をテストデータとトレーニングデータについて比較すると, トレーニングデータのほうが大きくなっていて過学習している可能性が考えられる.

また, 同じ最小二乗法を利用した OLS モデルと比較すると結果が著しく低くなっていることからコード上の誤りがある可能性も考えられる.

5.3　OLS

OLS モデルについては前処理の前後に関わらずトレーニングデータのほうが大きい値を示している. これより, OLS モデルについても過学習が起こっている可能性が考えられる.

また, 決定係数を比較してみるとトレーニングデータのほうが決定係数が小さくなっていることからも過学習の可能性を示している. これより, 過学習を防ぐようなトレーニングデータの割合をみつけることによって結果の向上を図ることができると考える.

今回は 1 年分のデータで学習を行ったが, 回帰モデルとして扱う場合は気象の季節性を考慮すると複数年のデータを利用したほうが精度が向上する可能性がある.

6 今後の課題

LinearSVC では, 最も良い結果が 81% 程度だったため, データセットの数を増やして学習を行う. また, 他の分類学習モデルを使用して, 精度の比較を調査していきたい.

OLS と LinearRegression はどちらも同じ平均に情報を用いた線形回帰モデルであるため性能は同程度になるはずである. しかし, OLS での結果に比べて, LinearRegression モデルで行なった場合の結果が極端に低くなっているため, コード上の誤りがある可能性を考え LinearRegression モデルのコードを見直すことが必要がある.

OLS モデルを用いた場合では, データ数を増やしたり, 特徴量を関連度によって選別したりして更なる精度の向上をしていきたい. また, 今回完成させることのできなかった StatsModels の AR モデルを用いた場合と今回の結果を比較してどちらの方が精度が高くできるのか調査していきたい.

参考文献

[1] 名古屋気象台 天気予報の仕組み, https://www.data.jma.go.jp/nagoya/shosai/info/shikumi.html, 2021/06/03

[2] 気象庁 各種データ・資料, https://www.data.jma.go.jp/gmd/risk/obsdl/, 2021/05/27.

[3] 気象庁 各種データ・資料 データについて, https://www.data.jma.go.jp/gmd/risk/obsdl/top/help4.html, 2021/05/27.

[4] ガントチャート, https://ja.wikipedia.org/wiki/ガントチャート, 2020/07/02.

[5] scikit-learn チートシート, https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html, 2021/05/27.

[6] 基礎からはじめる時系列解析入門, https://techblog.nhn-techorus.com/archives/14093, 2021/06/24.