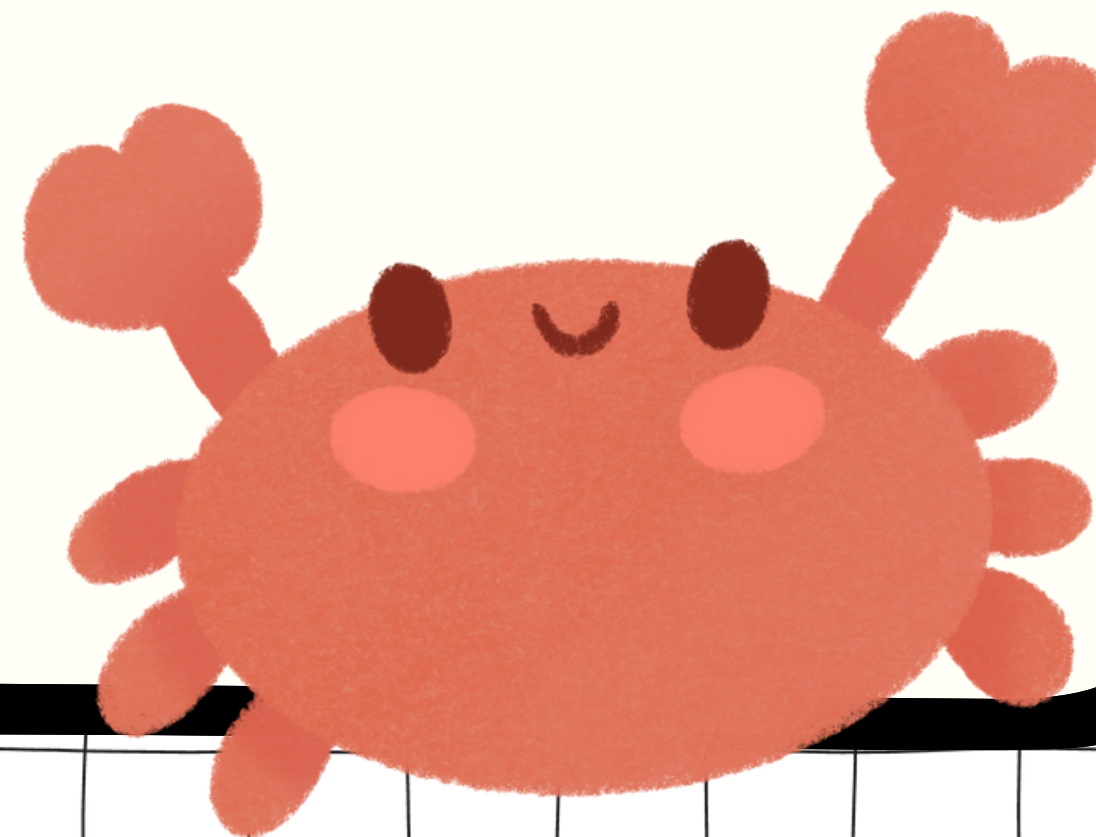


•••

Crab Age

Prediction



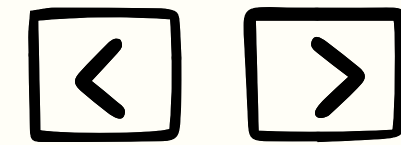
จัดทำโดย



นายภฤชนัส เหล็กดี รหัสนักศึกษา 6404062620036

นายวีรชัย ปฐมสุนทรชัย รหัสนักศึกษา 6404062620117

ที่มาและความสำคัญของโครงการ



ปูเป็นอาหารมีรสชาติอร่อยมาก และหลายประเทศทั่วโลก
นำเข้าปูเพื่อการบริโภคเป็นจำนวนมากในทุกๆปี

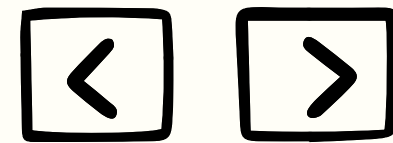
ประโยชน์หลักของการเลี้ยงปู คือ ค่าแรงต่ำมาก ต้นทุน
การผลิตค่อนข้างต่ำและเติบโตเร็วมาก

ธุรกิจการเลี้ยงปูเชิงพาณิชย์ เป็นการพัฒนาวิถีชีวิตของ
ชาวพื้นที่ชายฝั่งทะเลด้วยการดูแลและการจัดการที่เหมาะสม

เราสามารถสร้างรายได้จากธุรกิจการเลี้ยงปูได้มากกว่าการ
เลี้ยงกุ้งอีกด้วย



ประโยชน์ของโครงการ



- สำหรับเกษตรกรผู้เลี้ยงปูเชิงพาณิชย์ ที่ทราบอายุที่เหมาะสมของปูจะช่วยให้พวกเขาตัดสินใจว่า จะเก็บเกี่ยวปูเมื่อใด
- เกษตรกรเลี้ยงปูสามารถวางแผนการดูแลและบำรุงรักษาในระยะเวลาที่เหมาะสม โดยที่ไม่ต้องรอให้ปูโตมากจนแล้วจึงดำเนินการ



ข้อมูลที่นำมาใช้ทำโครงงาน



ที่มาของข้อมูล Kaggle

<https://www.kaggle.com/datasets/sidhus/crab-age-prediction>

Crab Age Prediction

Physical attributes of Crabs found in Boston area - to predict age of Crab



Data Card Code (102) Discussion (2)

ข้อมูลที่นำมาใช้ทำโครงการ



Attribute หรือ Feature

id	Sex	Length	Diameter	Height	Weight	Shucked ...	Viscera W...	Shell Weight
74051	I	1.05	0.7625	0.275	8.618248	3.6570855	1.7293195	2.721552
74052	I	1.1625	0.8875	0.275	15.5071765	7.030676	3.24601775	3.96893
74053	F	1.2875	0.9875	0.325	14.571643	5.556502	3.8838815	4.819415
74054	F	1.55	0.9875	0.3875	28.3778495	13.380964	6.5487345	7.030676
74055	I	1.1125	0.85	0.2625	11.7650425	5.5281525	2.4664065	3.33106625
74056	M	1.425	1.1125	0.35	24.834162	8.731646	5.71242425	8.0796075
74057	M	1.7125	1.325	0.45	46.67745175	21.2337755	11.963489	11.3681495

id - ไอดีของปูแต่ละตัว

Sex - เพศของปู

Length - ความยาวของปู

Diameter - เส้นผ่านศูนย์กลางของปู

Height - ความสูงของปู

Weight - น้ำหนักของปู

Shucked Weight - น้ำหนักที่ชั่งได้ (ไม่รวมเปลือกของปู)

Viscera Weight - น้ำหนักอวัยวะภายใน (ช่องท้องส่วนลึกภายใน)

Shell Weight - น้ำหนักของเปลือก

ข้อมูลที่นำมาใช้ทำโครงงาน



Output หรือ Label

id	Age
74051	10
74052	10
74053	10
74054	10
74055	10
74056	10
74057	10

id - ไอดีของปูแต่ละตัว

Age - เอาท์พุทเป็นตัวเลขซึ่งแสดงถึงอายุของปู

Platform ที่ใช้ทำโครงงาน

Kaggle



The screenshot shows a Kaggle notebook interface with the title "Model Training & Evaluation". The code is written in Python and includes the following sections:

```
LR_parameters = {'fit_intercept': [True, False]}
# LGB_parameters = {
#     'n_estimators': [100, 500, 1000],
#     'learning_rate': [0.01, 0.001]
# }
XGB_parameters = {
    'n_estimators': [100, 500, 1000],
    'learning_rate': [0.01, 0.001]
}
RF_parameters = {
    'n_estimators': [100, 500, 1000],
    'max_depth': [3, 10]
}

mlp = Sequential()
mlp.add(Dense(128, activation='sigmoid', input_shape=(X_Train1.shape[1],)))
mlp.add(Dense(64, activation='sigmoid'))
mlp.add(Dense(1))
mlp.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')

maeLRarr = []
maeLGBarr = []
maeXGBarr = []
maeRFarr = []
maeMLParr = []

cv = KFold(n_splits = 10)
i=0
for train_idx , test_idx in cv.split(X_Train1,Y_Train1):
    xxTrain , xxTest = X_Train1.iloc[train_idx] , X_Train1.iloc[test_idx]
    yyTrain , yyTest = Y_Train1.iloc[train_idx] , Y_Train1.iloc[test_idx]

    lr_grid_search = GridSearchCV(LinearRegression(), LR_parameters)
    lr_grid_search.fit(xxTrain, yyTrain)

    best_params = lr_grid_search.best_params_
```


ขั้นตอนการดำเนินงาน



1

ศึกษาข้อมูลโจทย์และ Dataset

2

เตรียมและคัดกรองข้อมูล

3

Train Model & ปรับพารามิเตอร์

4

ประเมิน Model

ศึกษาข้อมูลโจทย์และ Dataset



การวัดผล

Evaluation

Submissions will be evaluated using Mean Absolute Error (MAE),

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - y_i|$$

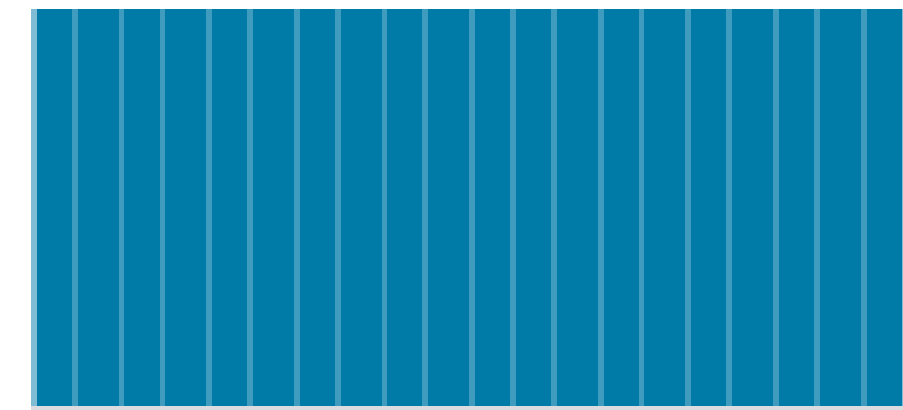
ศึกษาข้อมูลโอทอยและ Dataset

Train Data จะเป็นไฟล์ CSV ซึ่งมีทั้งหมด
ประมาณ 74,000 แถว และมีจำนวน 10 คอลัมน์
ที่ประกอบไปด้วย

- | | |
|-------------|-------------------|
| 1. ID | 6. Weight |
| 2. Sex | 7. Shucked Weight |
| 3. Length | 8. Viscera Weight |
| 4. Diameter | 9. Shell Weight |
| 5. Height | 10. Age |

train.csv (5.21 MB)

id



0

74.0k

ศึกษาข้อมูลโจทย์และ Dataset



การ Submission ต้องการผลลัพธ์เป็นไฟล์ CSV
ที่ประกอบไปด้วยคอลัมน์ ID และ Age
ซึ่งมีจำนวนทั้งหมดประมาณ 49,000 แถว

sample_submission.csv (467.74 kB)

id	Age
74.1k	10
123k	10

เตรียมและคัดกรองข้อมูล



```
missingtrain = dftrain.isnull().sum()
```

เช็คค่าว่างในแต่ละคอลัมน์

id	0
Sex	0
Length	0
Diameter	0
Height	0
Weight	0
Shucked Weight	0
Viscera Weight	0
Shell Weight	0
Age	0
dtype:	int64

เตรียมและคัดกรองข้อมูล

...

ดูผลสรุปของข้อมูล

```
dftrain.describe()
```

	id	Length	Diameter	Height	Weight	Shucked Weight	Viscera Weight	Shell Weight	Age
count	74051.000000	74051.000000	74051.000000	74051.000000	74051.000000	74051.000000	74051.000000	74051.000000	74051.000000
mean	37025.000000	1.317460	1.024496	0.348089	23.385217	10.104270	5.058386	6.723870	9.967806
std	21376.826729	0.287757	0.237396	0.092034	12.648153	5.618025	2.792729	3.584372	3.175189
min	0.000000	0.187500	0.137500	0.000000	0.056699	0.028349	0.042524	0.042524	1.000000
25%	18512.500000	1.150000	0.887500	0.300000	13.437663	5.712424	2.863300	3.968930	8.000000
50%	37025.000000	1.375000	1.075000	0.362500	23.799405	9.908150	4.989512	6.931453	10.000000
75%	55537.500000	1.537500	1.200000	0.412500	32.162508	14.033003	6.988152	9.071840	11.000000
max	74050.000000	2.012815	1.612500	2.825000	80.101512	42.184056	21.545620	28.491248	29.000000

```
dftrain['Height'] = dftrain['Height'].replace({0:0.348089})
```


เตรียมและคัดกรองข้อมูล



แปลงเพศเป็นตัวเลข

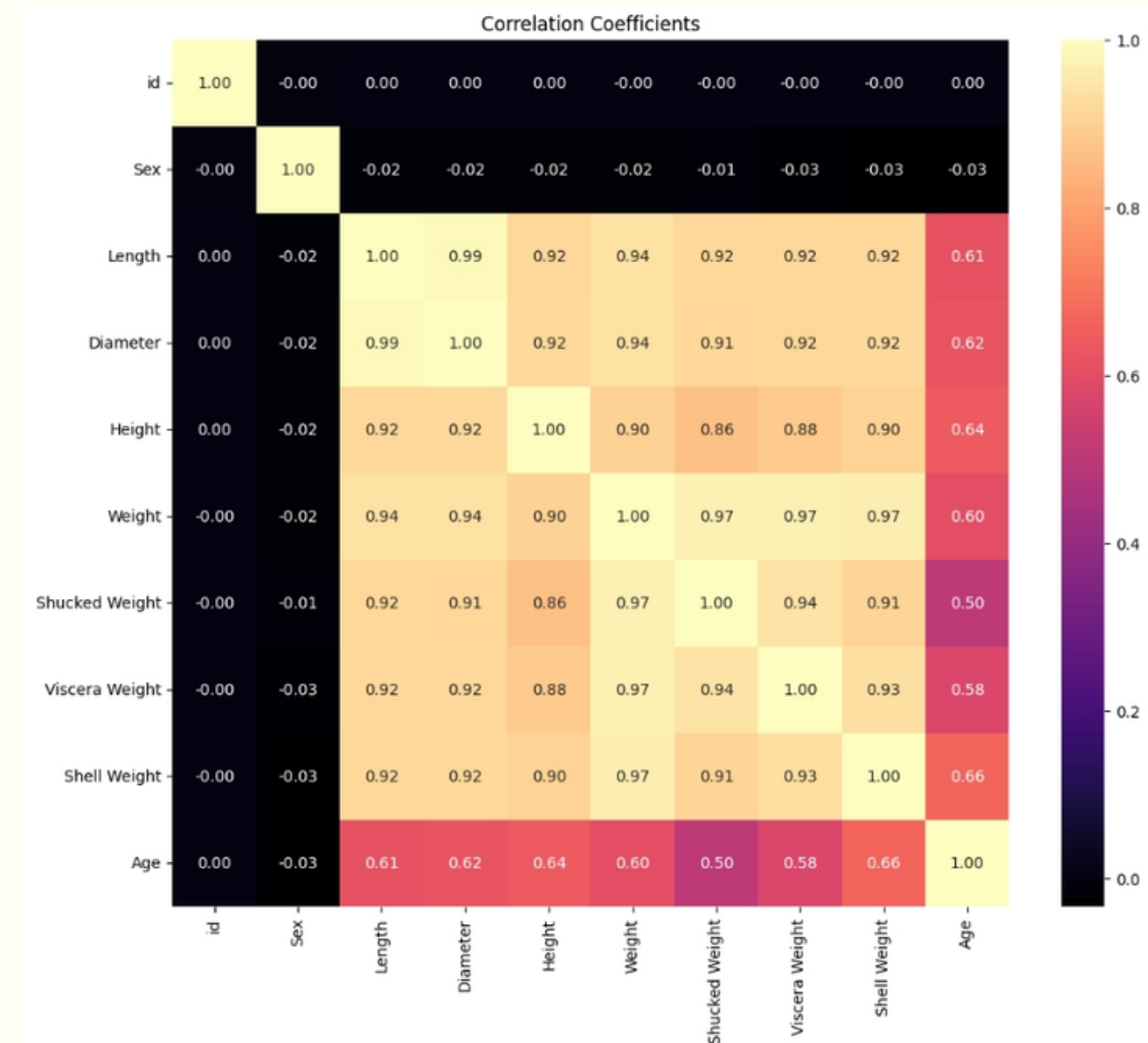
	id	Sex	Length	Diameter	Height	Weight	Shucked Weight	Viscera Weight	Shell Weight	Age
0	0	I	1.5250	1.1750	0.3750	28.973189	12.728926	6.647958	8.348928	9
1	1	I	1.1000	0.8250	0.2750	10.418441	4.521745	2.324659	3.401940	8
2	2	M	1.3875	1.1125	0.3750	24.777463	11.339800	5.556502	6.662133	9
3	3	F	1.7000	1.4125	0.5000	50.660556	20.354941	10.991839	14.996885	11
4	4	I	1.2500	1.0125	0.3375	23.289114	11.977664	4.507570	5.953395	8
...
74046	74046	F	1.6625	1.2625	0.4375	50.660556	20.680960	10.361742	12.332033	10
74047	74047	I	1.0750	0.8625	0.2750	10.446791	4.323299	2.296310	3.543687	6
74048	74048	F	1.4875	1.2000	0.4125	29.483480	12.303683	7.540967	8.079607	10
74049	74049	I	1.2125	0.9625	0.3125	16.768729	8.972617	2.919999	4.280774	8
74050	74050	I	0.9125	0.6750	0.2000	5.386405	2.055339	1.034757	1.700970	6

```
le = LabelEncoder()  
dftrain['Sex'] = le.fit_transform(dftrain['Sex'])
```

เตรียมและคัดกรองข้อมูล

Heat Map

```
X_Train = dftrain[['Shell Weight', 'Height', 'Diameter', 'Length']]
Y_Train = dftrain[['Age']]
```



เตรียมและคัดกรองข้อมูล



Scale ค่าให้อยู่ระหว่าง 0-1

```
scaler = MinMaxScaler()  
X_Train=pd.DataFrame(scaler.fit_transform(X_Train), index=X_Train.index, columns=X_Train.columns)  
Test=pd.DataFrame(scaler.fit_transform(Test), index=Test.index, columns=Test.columns)
```

	Shell Weight	Height	Diameter	Length
0	0.291978	0.128889	0.703390	0.732750
1	0.118087	0.093333	0.466102	0.499914
2	0.232686	0.128889	0.661017	0.657421
3	0.525660	0.173333	0.864407	0.828624
4	0.207773	0.115556	0.593220	0.582091
...
74046	0.431988	0.151111	0.762712	0.808080
74047	0.123069	0.093333	0.491525	0.486218
74048	0.282511	0.142222	0.720339	0.712206
74049	0.148979	0.106667	0.559322	0.561547
74050	0.058296	0.066667	0.364407	0.397192

Train Model & ปรับพารามิเตอร์



ใช้ KFold ในการแบ่งข้อมูล

```
cv = KFold(n_splits = 10)
i=0
for train_idx , test_idx in cv.split(X_Train1,Y_Train1):
    xxTrain , xxTest = X_Train.iloc[train_idx] , X_Train.iloc[test_idx]
    yyTrain , yyTest = Y_Train.iloc[train_idx] , Y_Train.iloc[test_idx]
```

Train Model & ปรับพารามิเตอร์



การปรับพารามิเตอร์ด้วย GridSearchCV

```
XGB_parameters = {  
    'n_estimators': [100, 500, 1000],  
    'learning_rate': [0.01, 0.001]  
}
```

```
xgb_grid_search = GridSearchCV(XGBRegressor(objective="reg:pseudohubererror"), XGB_parameters, scoring='neg_mean_absolute_error')  
xgb_grid_search.fit(xxTrain1, yyTrain1)  
  
xgb_best_params = xgb_grid_search.best_params_
```

Train Model & ปรับพารามิเตอร์



การ Train Model

```
lr = LinearRegression(**best_params)
lgb = LGBMRegressor(**lgb_best_params, early_stopping_rounds=500)
xgb = XGBRegressor(**xgb_best_params, early_stopping_rounds=500)
rf = RandomForestRegressor(**rf_best_params)
mlp.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')
```


ประเมิน Model



สร้าง Array มาเก็บผล MAE แต่ละรอบ

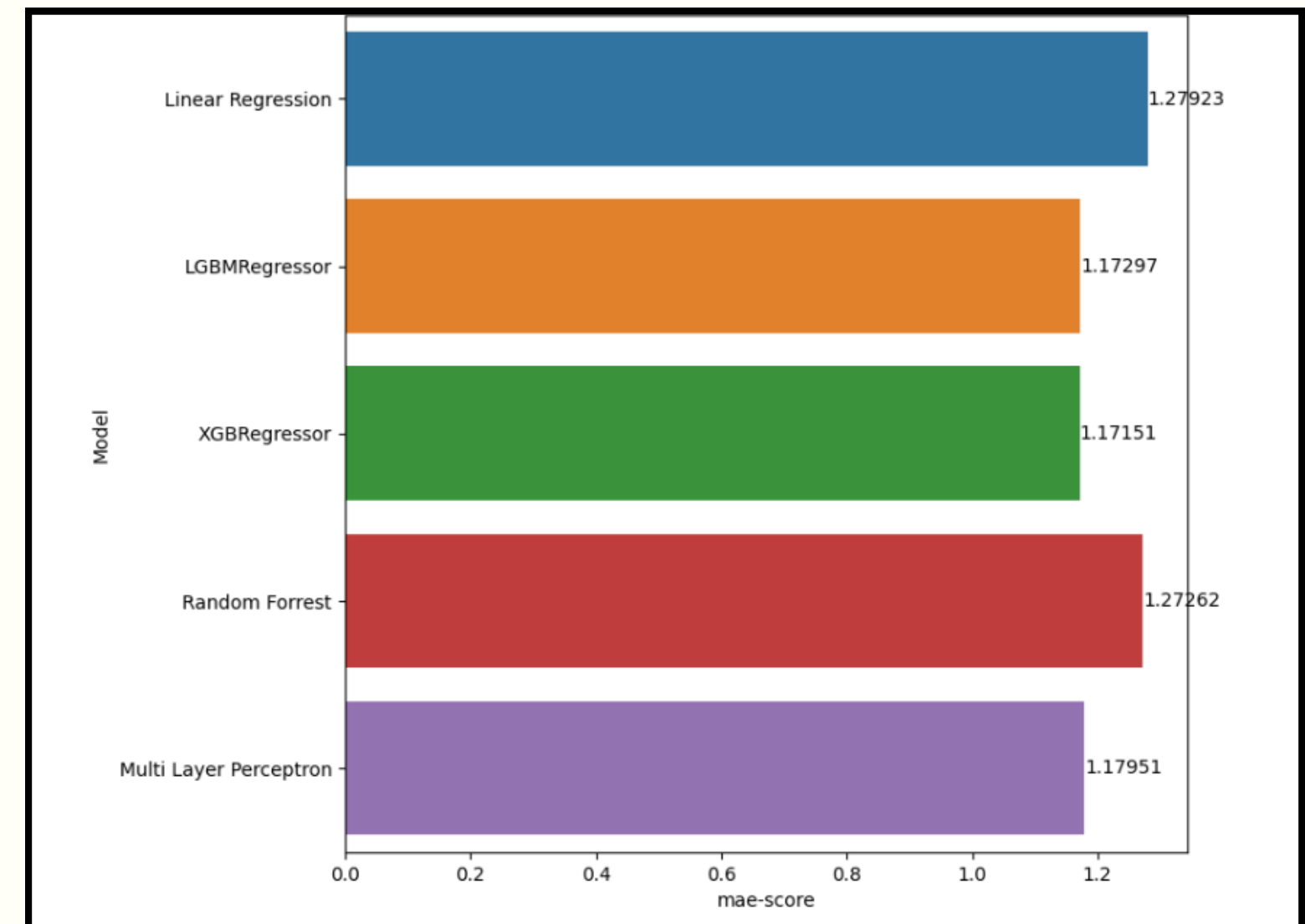
```
maeLRarr = []  
maeLGBarr = []  
maeXGBarr = []  
maeRFarr = []  
maeMLParr = []
```

```
predXGB = xgb.predict(xxTest1)  
maeXGB = np.sqrt(mean_absolute_error(yyTest1, np.round(predXGB)))  
maeXGBarr.append(maeXGB)
```

ประเมิน Model



นำค่าที่เก็บไว้ในแต่ละรอบมาเฉลี่ย
แล้วนำมาทำเป็นกราฟ



Submission

Playground Prediction Competition

Regression with a Crab Age Dataset

Playground Series - Season 3, Episode 16

k

Kaggle · 1,429 teams · 5 months ago

Overview

Data

Code

Models

Discussion

Leaderboard

Rules

Team

Submissions

Late Submission

...

Submissions




You selected 0 of 2 submissions to be evaluated for your final leaderboard score. Since you selected less than 2 submission, Kaggle auto-selected up to 2 submissions from among your public best-scoring unselected submissions for evaluation. The evaluated submission with the best Private Score is used for your final score.

Submissions evaluated for final score

AllSuccessfulSelectedErrors

Recent

Submission and Description	Private Score	Public Score	Selected
<div><div><div></div></div><div>XGB_submission (9).csv</div><div>Complete (after deadline) · now</div></div>	1.34584	1.35095	<input type="checkbox"/>

#	△	Team	Members	Score
1	▲ 2	Epikt		1.33429
2	▲ 12	Emincan Yilmaz		1.33510
3	▲ 3	Ravi Ramakrishnan		1.33513

นำไปประยุกต์ใช้



Streamlit

Streamlit เป็นเฟรมเวิร์ก App Open Source
สำหรับทีม Machine Learning และ Data Science
สร้างเว็บแอปที่สวยงามในไม่กี่นาที

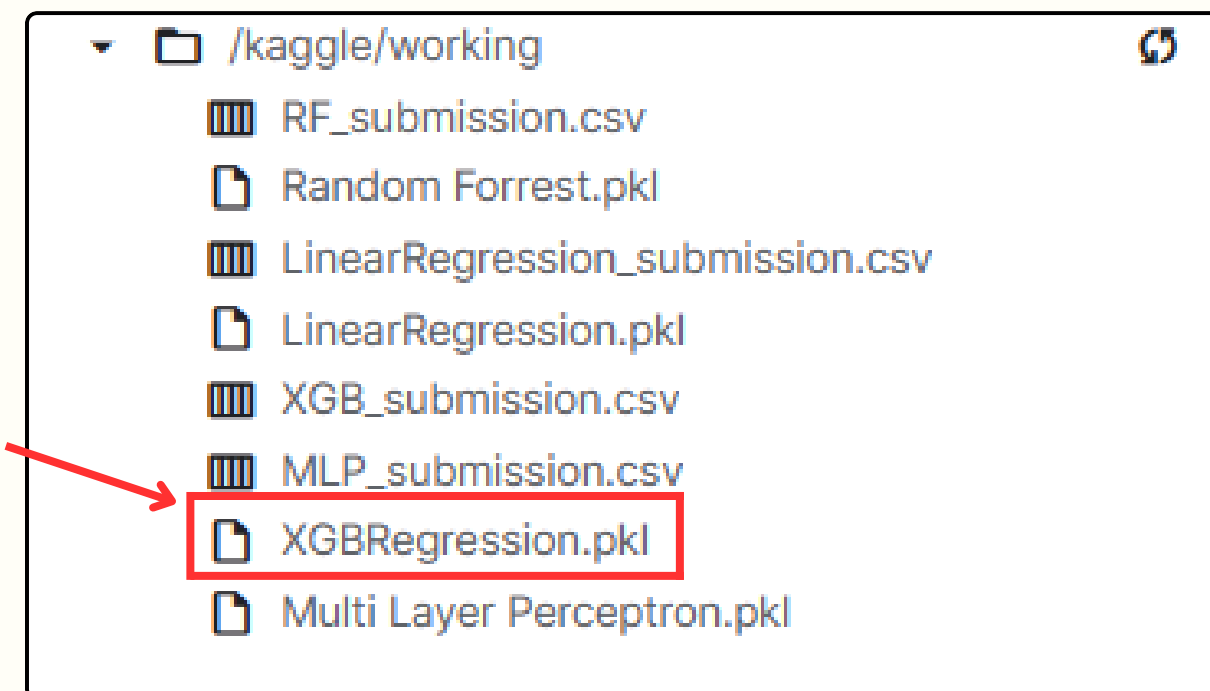
```
import streamlit as st
```

นำไปประยุกต์ใช้

ใช้ Library Pickle เพื่อนำออก Model ที่ต้องการ

```
with open("XGBRegression.pkl", 'wb') as file:  
    pickle.dump(xgb, file)
```

โหลด Model ที่ต้องการ



นำไปประยุกต์ใช้

ใช้ Library Pickle เพื่อนำเข้า Model ที่โหลดมา

```
with open('XGB_submission.pkl', 'rb') as file:  
    model = pickle.load(file)
```

ทำนาย input ที่รับมา และแสดงผลบนหน้าเว็บ

```
prediction = model.predict(input_data)  
show_predict = int(round(prediction[0],0))  
st.write(f'Predicted Crab Age: {show_predict}')
```


นำไปประยุกต์ใช้

Crab Age Prediction

Crab Sex

- ☒ Male
☐ Female
☐ Unknown

Length

1.66250

- +

Diameter

1.36250

- +

Height

0.47500

- +

Weight

46.05376

- +

Shucked Weight

16.72621

- +

Viscera Weight

7.38504

- +

Shell Weight

15.02524

- +

Predicted Crab Age

Predicted Crab Age: 17

สรุปผล

ในการทำ Model สำหรับการทำนายค่านั้น ซึ่งควรคำนึงถึง Input และ Output ของข้อมูล เพื่อเลือกใช้ Model ที่เหมาะสมกับประเภทข้อมูลนั้นๆแล้วนำมาผ่านการปรับ Parameter ต่างๆด้วยค่าที่เหมาะสม แล้วจึงมาเทียบประสิทธิภาพว่าโมเดลไหนให้ผลลัพธ์ดีกว่า

ซึ่งในโครงงานของเรานั้น พบว่า XGBoost ให้ผลลัพธ์ที่ดีที่สุด
เนื่องจากให้ค่า MAE ที่น้อยกว่า Model อื่นๆ
และยังส่ง Submission บน Kaggle ได้คะแนนดีที่สุดด้วย
เราจึงนำ Model มาปรับใช้บน Web Application



Thank you