ชุดข้อมูล: ข้อมูลทั่วไปของนักศึกษาคณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ ปีการศึกษา 2567 (3.1)

Link: https://docs.google.com/spreadsheets/d/1Ro8KTCX40HuK9s2vXS9qR9SjfitL8Y89/edit?usp=sharing&ouid=115842441160570853727&rtpof=true&sd=true

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mour
```

อ่านข้อมูลเข้ามาเก็บใน DataFrame

```
#เช็คข้อมูลเบื้องต้น
import pandas as pd
df = pd.read_excel('/content/drive/My Drive/แบบสอบถามนักศึกษา.xlsx')
df.head()
```

	เพศ	ชั้นปี	สาขา	เกรดเฉลี่ยสะสม	จังหวัดภูมิลำเนา	โรงเรียนที่จบการศึกษามัธยมศึกษาตอนปลาย	รายได้ต่อเดือนของนักศึกษา (บาท)	หอพัก	รายจ่ายต่อเดือนของนักศึกษา (บาท)	รายได้ต่อเดือนของครอบครัว (บาท)
0	หญิง	ชั้นปีที่ 2	วัสดุศาสตร์	2.09	สงขลา	โรงเรียนเทศบาล 1 (เอ็งเสียงสามัคคี)	10000	หอนอก	5000	1000000
1	หญิง	ชั้นปีที่ 4	วิทยาการคอมพิวเตอร์	3.25	นครศรีธรรมราช	โรงเรียนทุ่งใหญ่เฉลิมราชอนุสรณ์รัชมังคลากภิเษก	13000	หอนอก	10000	300000
2	ชาย	ชั้นปีที่ 3	คณิตศาสตร์	3.20	สงขลา	โรงเรียนหาดใหญ่วิทยาลัย	3000	หอนอก	3000	10000

```
#จำนวน record และ feature (3.2)
df.shape
```

```
(171, 18)
```

```
#เช็ค feature (3.3)
df.columns
```

```
Index(['เพศ', 'ชั้นปี', 'สาขา', 'เกรดเฉลี่ยสะสม', 'จังหวัดภูมิลำเนา',
       'โรงเรียนที่จบการศึกษามัธยมศึกษาตอนปลาย',
       'รายได้ต่อเดือนของนักศึกษา (บาท)', 'หอพัก',
       'รายจ่ายต่อเดือนของนักศึกษา (บาท)', 'รายได้ต่อเดือนของครอบครัว (บาท)',
       'การกู้ กยศ.', 'ความต้องการในการศึกษาต่อ',
       'มีความสนใจที่จะรับราชการหรือไม่',
       'ชั่วโมงในการเรียนในห้องเรียนเฉลี่ยต่อสัปดาห์',
       'ยานพาหนะที่ใช้มาเรียน ', 'ระบบปฏิบัติการโทรศัพท์',
       'ระยะทางจากที่พักมาห้องเรียน (กิโลเมตร)',
       'การใช้อินเตอร์อินเน็ตกี่ชั่วโมงใน 1 วัน '],
      dtype='object')
```

ทำความสะอาดข้อมูล (3.4)

```
#Data cleansing
df = df.drop(['จังหวัดภูมิลำเนา', 'โรงเรียนที่จบการศึกษามัธยมศึกษาตอนปลาย', 'การกู้ กยศ.', 'ความต้องการใ
df.head()
```

	เพศ	ชั้นปี	สาขา	เกรดเฉลี่ยสะสม	รายได้ต่อเดือนของนักศึกษา (บาท)	หอพัก	รายจ่ายต่อเดือนของนักศึกษา (บาท)	รายได้ต่อเดือนของครอบครัว (บาท)	ยานพาหนะที่ใช้มาเรียน	ระยะทางจากที่พักมาห้องเรียน (กิโลเมตร)	กา อินเ อิน ชั่ว ใ
0	หญิง	ชั้นปีที่ 2	วัสดุศาสตร์	2.09	10000	หอนอก	5000	1000000	รถส่วนตัว	7.6	
1	หญิง	ชั้นปีที่ 4	วิทยาการคอมพิวเตอร์	3.25	13000	หอนอก	10000	300000	รถส่วนตัว	2.0	

```
#เช็ค Null
df.isnull().sum()
```

	0
เพศ	0
ชั้นปี	0
สาขา	0
เกรดเฉลี่ยสะสม	0
รายได้ต่อเดือนของนักศึกษา (บาท)	0
หอพัก	0
รายจ่ายต่อเดือนของนักศึกษา (บาท)	0
รายได้ต่อเดือนของครอบครัว (บาท)	0
ยานพาหนะที่ใช้มาเรียน	0
ระยะทางจากที่พักมาห้องเรียน (กิโลเมตร)	0
การใช้อินเตอร์อินเน็ตกี่ชั่วโมงใน 1 วัน	0

dtype: int64

```
#ลบ Outlier
def remove_outliers(df, column):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    upper_bound = Q3 + 1.5 * IQR
    df.drop(df[df[column] > upper_bound].index, inplace=True)
```

```
columns_to_check = [
    'รายได้ต่อเดือนของนักศึกษา (บาท)',
    'รายจ่ายต่อเดือนของนักศึกษา (บาท)',
    'รายได้ต่อเดือนของครอบครัว (บาท)',
]

for column in columns_to_check:
    remove_outliers(df, column)


#ตรวจสอบจำนวน record และ feature ที่เหลือ
df.shape
```

```
(145, 11)
```

```
#Data transformation
df = pd.get_dummies(df, dtype = int)
df.head()
```

	เกรดเฉลี่ยสะสม	รายได้ต่อเดือนของนักศึกษา (บาท)	รายจ่ายต่อเดือนของนักศึกษา (บาท)	รายได้ต่อเดือนของครอบครัว (บาท)	ระยะทางจากที่พักมาห้องเรียน (กิโลเมตร)	การใช้อินเตอร์อินเน็ตกี่ชั่วโมงใน 1 วัน	เพศ_ชาย	เพศ_หญิง	ชั้นปี_ชั้นปีที่ 2	ช ปี_ช ปีที่
2	3.20	3000	3000	10000	8.0	14	1	0	0	
3	2.60	6000	5600	60000	3.0	6	1	0	0	
4	2.98	6000	5000	35000	2.0	9	1	0	1	
16	2.90	6000	4500	70000	0.5	12	0	1	1	
18	3.24	6000	5500	18000	0.5	10	1	0	0	

5 rows × 29 columns

```
#เช็ค feature
df.columns
```

```
Index(['เกรดเฉลี่ยสะสม', 'รายได้ต่อเดือนของนักศึกษา (บาท)',
       'รายจ่ายต่อเดือนของนักศึกษา (บาท)', 'รายได้ต่อเดือนของครอบครัว (บาท)',
       'ระยะทางจากที่พักมาห้องเรียน (กิโลเมตร)',
       'การใช้อินเตอร์อินเน็ตกี่ชั่วโมงใน 1 วัน ', 'เพศ_ชาย', 'เพศ_หญิง',
       'ชั้นปี_ชั้นปีที่ 2', 'ชั้นปี_ชั้นปีที่ 3', 'ชั้นปี_ชั้นปีที่ 4',
       'สาขา_คณิตศาสตร์', 'สาขา_จุลชีววิทยา', 'สาขา_ชีววิทยา', 'สาขา_ฟิสิกส์',
       'สาขา_วัสดุศาสตร์', 'สาขา_วิทยาการคอมพิวเตอร์',
       'สาขา_วิทยาศาสตร์พอลิเมอร์', 'สาขา_สถิติ', 'สาขา_เคมี',
       'สาขา_เคมี - ชีววิทยาประยุกต์', 'สาขา_เทคโนโลยีชีวภาพ',
       'สาขา_เทคโนโลยีสารสนเทศและการสื่อสาร', 'หอพัก_หอนอก', 'หอพัก_หอใน',
```

```
       'ยานพาหนะที่ใช้มาเรียน _รถรับจ้าง', 'ยานพาหนะที่ใช้มาเรียน _รถส่วนตัว',
       'ยานพาหนะที่ใช้มาเรียน _รถเมล์มอ', 'ยานพาหนะที่ใช้มาเรียน _เดิน'],
      dtype='object')
```

เตรียม Features (X) และ Target (y) (3.5)

```
X = df.drop('รายจ่ายต่อเดือนของนักศึกษา (บาท)', axis=1)
y = df['รายจ่ายต่อเดือนของนักศึกษา (บาท)']
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 49
```

```
#บันทึกเก็บลงไฟล์ csv (3.6)
df.to_csv("ข้อมูลทั่วไปนักศึกษา_Cleansed.csv", index = False)
```

วิธีการดำเนินการ (4)

เลือกอัลกอริทึม: Linear regression, Decision tree, kNN (4.1)

วิธีการประเมิน (Evaluation matrices) ที่เลือกใช้: MSE (Mean Squared Error), RMSE (Root Mean Squared Error), MAE (Mean Absolute Error) (4.3)

```
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.model_selection import GridSearchCV
import numpy as np

def calculate_evaluation_metrics(y_test, y_pred):
  mse = mean_squared_error(y_test, y_pred)
  rmse = np.sqrt(mse)
  mae = mean_absolute_error(y_test, y_pred)
  return round(mse, 2), round(rmse, 2), round(mae, 2)
```

Linear regression

```
from sklearn.linear_model import LinearRegression
model_LR = LinearRegression()
model_LR.fit(X_train, y_train)
```

```
LinearRegression()
```

```
y_pred1 = model_LR.predict(X_test)
calculate_evaluation_metrics(y_test, y_pred1)
```

```
(1049285.83, 1024.35, 808.15)
```

Decision tree regression

```
from sklearn.tree import DecisionTreeRegressor
dt_regressor = DecisionTreeRegressor(random_state = 49)

param_grid = {
    'max_depth': range(1, 4),
    'min_samples_split': range(1, 4),
    'min_samples_leaf': range(1, 4),
    'criterion': ['squared_error', 'friedman_mse', 'absolute_error', 'poisson']
}

# ใช้ GridSearchCV เพื่อค้นหาค่าที่ดีที่สุด
grid_search = GridSearchCV(dt_regressor, param_grid, cv = 5, scoring = 'neg_mean_squared_err
grid_search.fit(X_train, y_train)

# แสดงผลลัพธ์
print(f"Best Hyperparameters: {grid_search.best_params_}")
best_model_DTR = grid_search.best_estimator_
```

Show hidden output

Best Hyperparameters: {'criterion': 'absolute_error', 'max_depth': 2, 'min_samples_leaf': 1, 'min_samples_split': 2}

```
y_pred2 = best_model_DTR.predict(X_test)
calculate_evaluation_metrics(y_test, y_pred2)
```

```
(941810.34, 970.47, 763.79)
```

kNN regression

```
from sklearn.neighbors import KNeighborsRegressor
knn = KNeighborsRegressor()
```

```
param_grid = {
    'n_neighbors': range(1, 12),
    'weights': ['uniform', 'distance'],
    'metric': ['euclidean', 'manhattan']
}

# ใช้ GridSearchCV เพื่อค้นหาค่าที่ดีที่สุด
grid_search = GridSearchCV(knn, param_grid, cv = 5, scoring = 'neg_mean_squared_error')
grid_search.fit(X_train, y_train)

# แสดงผลลัพธ์
print(f"Best Hyperparameters: {grid_search.best_params_}")
best_model_kNNR = grid_search.best_estimator_
```

```
Best Hyperparameters: {'metric': 'euclidean', 'n_neighbors': 3, 'weights': 'uniform'}
```

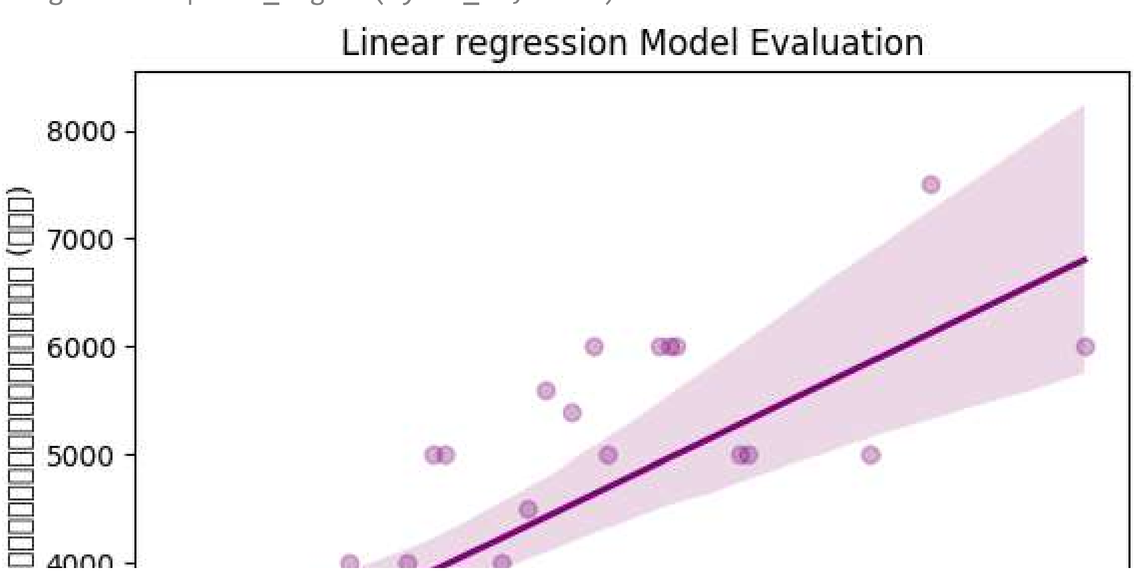
```
y_pred3 = best_model_kNNR.predict(X_test)
calculate_evaluation_metrics(y_test, y_pred3)
```

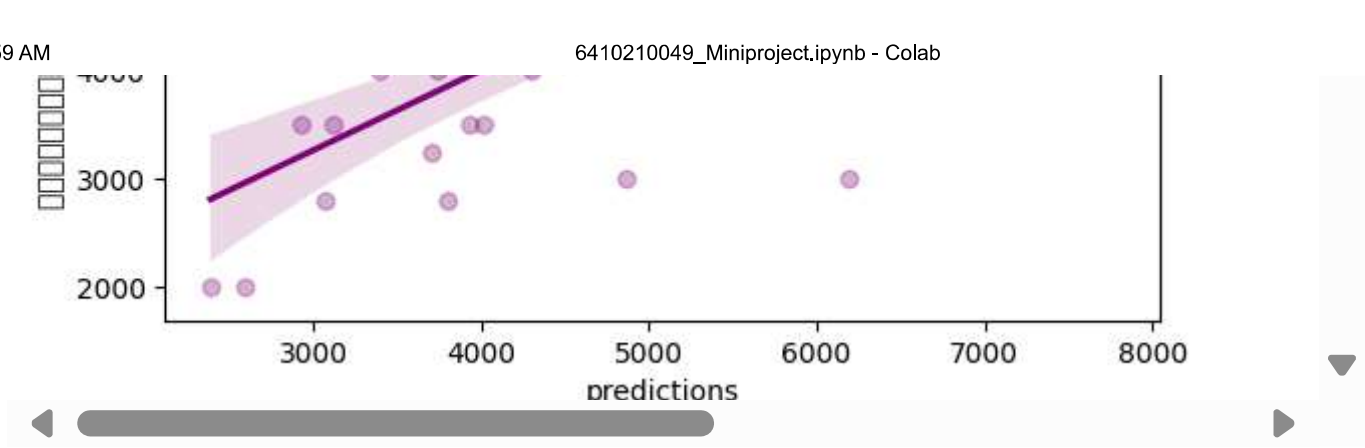
```
(769444.06, 877.18, 739.89)
```

Visualize the relationship between predicted and actual values (5.1)

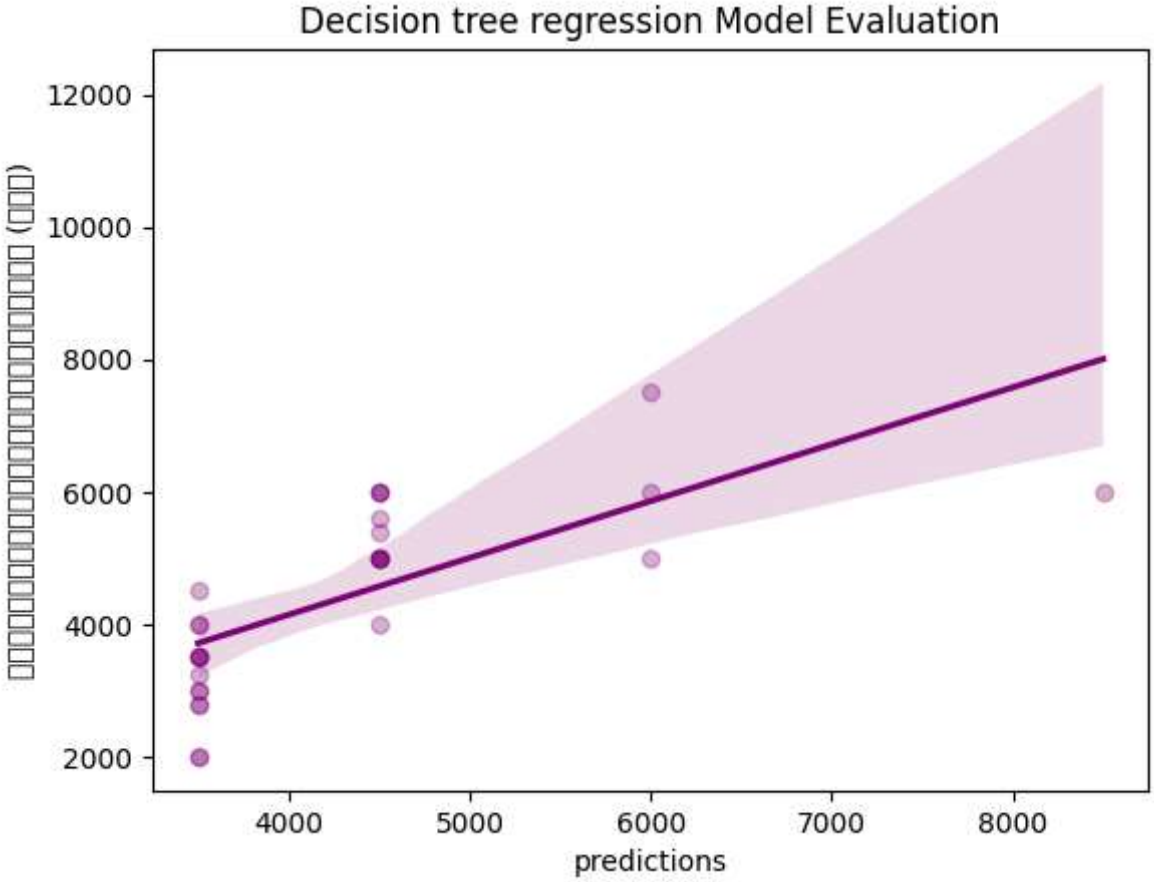
```
import seaborn as sns
import matplotlib.pyplot as plt


#Linear regression
sns.set_palette('RdPu_r')
sns.regplot(x = y_pred1, y = y_test, scatter_kws={'alpha': 0.3})
plt.xlabel('predictions')
plt.title('Linear regression Model Evaluation')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
  fig.canvas.print_figure(bytes_io, **kw)
```

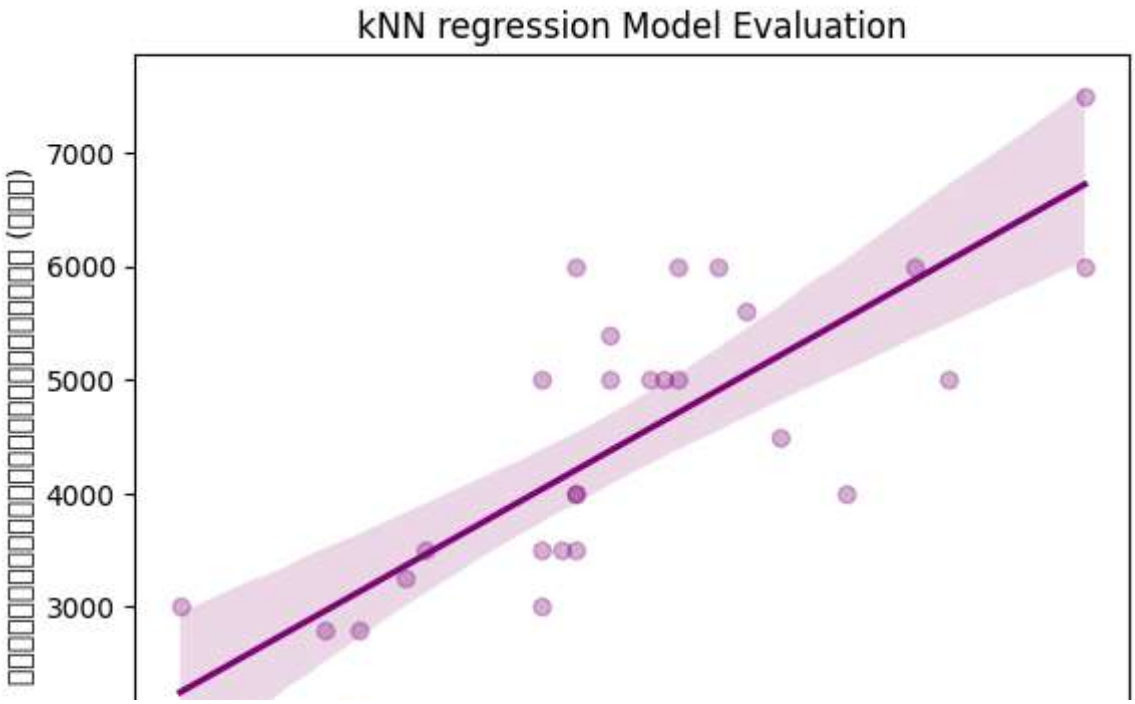





```
#Decision tree regression
sns.set_palette('RdPu_r')
sns.regplot(x = y_pred2, y = y_test, scatter_kws={'alpha': 0.3})
plt.xlabel('predictions')
plt.title('Decision tree regression Model Evaluation')
plt.show()
```




```
#kNN regression
sns.set_palette('RdPu_r')
sns.regplot(x = y_pred3, y = y_test, scatter_kws={'alpha': 0.3})
plt.xlabel('predictions')
plt.title('kNN regression Model Evaluation')
plt.show()
```


kNN regression Model Evaluation
7000
6000
5000
4000
3000