

Dilan's Travel Guide - Forecasting

How many first readers and purchases will the business achieve next month (on the 120th day)?

```
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

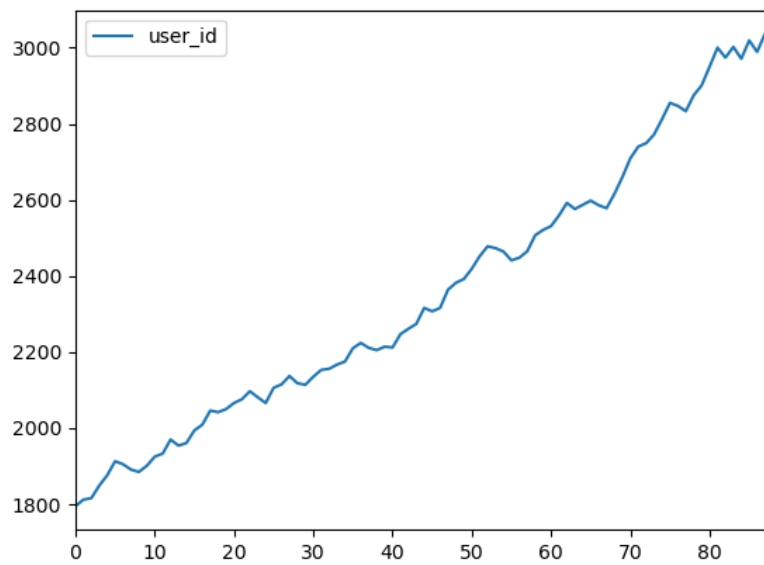
I predict the number of first readers for the 120th day.

```
#I import the dataset of first readers.
read = pd.read_csv('/home/...../dilan/dilans_first_read', delimiter = ';', names =
['my_time', 'event', 'country', 'user_id', 'source', 'topic'])
```

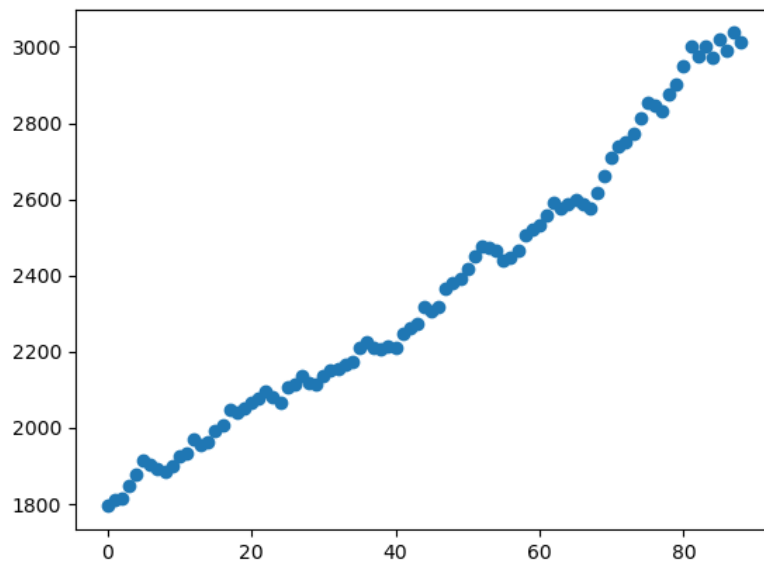
```
#I format the data.
read['my_date'] = pd.to_datetime(read['my_time']).dt.date
```

```
#I prepare the data for plotting the trend.
daily_fread = read.groupby('my_date').count()
daily_fread = daily_fread.reset_index(drop=True)
daily_fread = daily_fread[['user_id']]
```

```
#I plot the trend of the number of first readers.
daily_fread.plot()
plt.savefig('daily_fread_line.png')
```



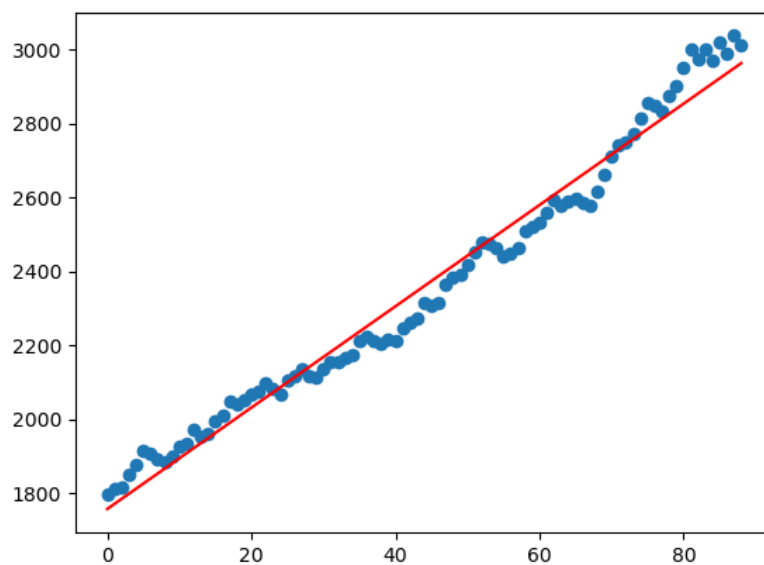
```
#I create the scatter plot of daily first readers.  
x = daily_fread.index  
y = daily_fread.user_id  
plt.scatter(x,y)  
plt.savefig('daily_fread_scatter.png')
```



```
#I fit a linear regression model.  
model = np.polyfit(x,y,1)  
predict = np.poly1d(model)  
model
```

```
array([ 13.70, 1756.67 ])
```

```
#I fit the linear regression line.  
x_lin_reg = range(0,89)  
y_lin_reg = predict(x_lin_reg)  
plt.scatter(x,y)  
plt.plot(x_lin_reg, y_lin_reg, c = 'r')  
plt.savefig('regression_fread.png')  
plt.show()
```



```
#I obtain the r2 score for the linear regression model.  
from sklearn.metrics import r2_score  
r2_score(y,predict(x))
```

0.97

```
predict(120)
```

3401.60

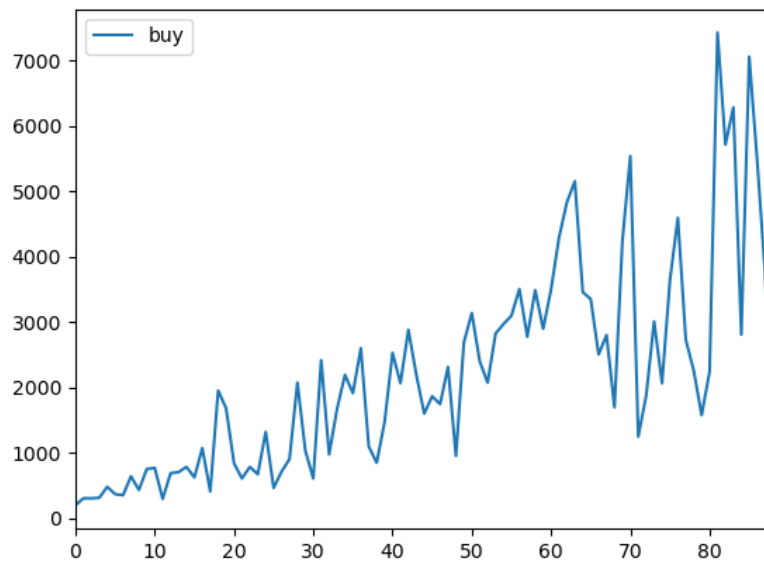
The prediction indicates that Dilan's business will have 3402 first readers on the 120th day. Based on the R^2 score, this appears to be a reliable prediction.

I predict the revenue for the 120th day.

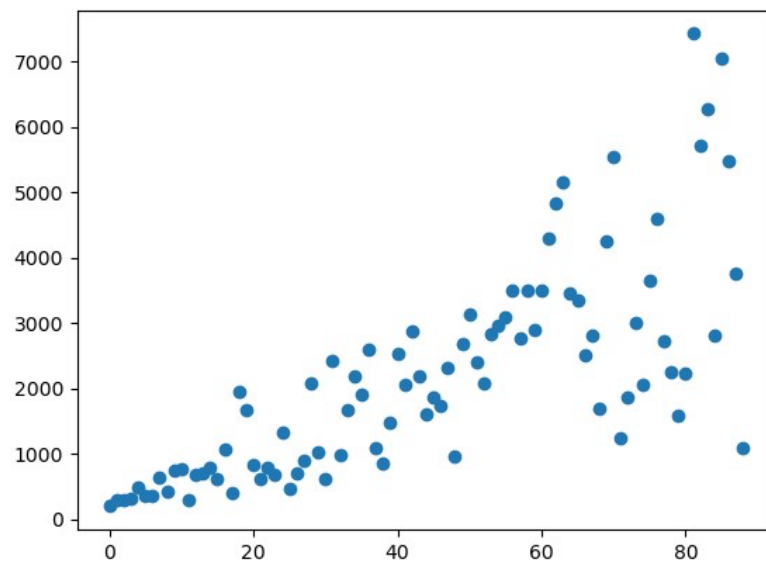
```
#I read the dataset for purchases.
buy = pd.read_csv('/home/...../dilan/dilans_buy', delimiter = ';', names =
['my_time', 'event', 'user_id', 'buy'])
```

```
#I calculate the sum of daily purchases (daily revenue).
daily_sum = buy.groupby('my_date')[['buy']].sum()
```

```
#I draw a line chart of daily revenues.
daily_sum = daily_sum.reset_index(drop=True)
daily_sum.plot()
plt.savefig('daily_revenue_line.png')
```



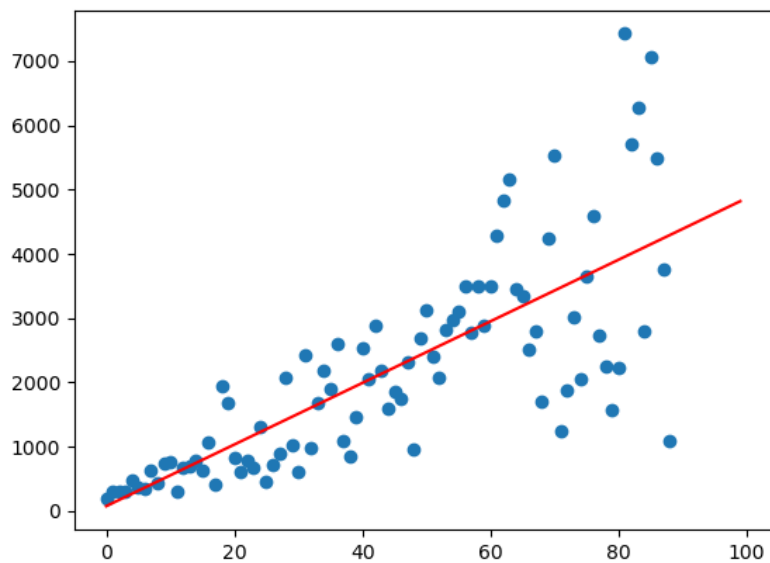
```
#I draw a scatter plot of daily revenues.
x = daily_sum.index
y = daily_sum.buy
plt.scatter(x,y)
plt.savefig('daily_revenue_scatter.png')
```



```
#I fit the linear regression model.  
x = daily_sum.index  
y = daily_sum.buy  
model = np.polyfit(x,y,1)  
predict = np.poly1d(model)  
model
```

```
#I draw the regression line.  
x_lin_reg = range(0,89)  
y_lin_reg = predict(x_lin_reg)  
plt.scatter(x,y)  
plt.plot(x_lin_reg, y_lin_reg, c = 'r')  
plt.savefig('regression1.png')  
plt.show()
```

```
array([47.89, 78.09])
```



```
#I get the r2 score for the linear regression model.
from sklearn.metrics import r2_score
r2_score(y,predict(x))
```

0.58

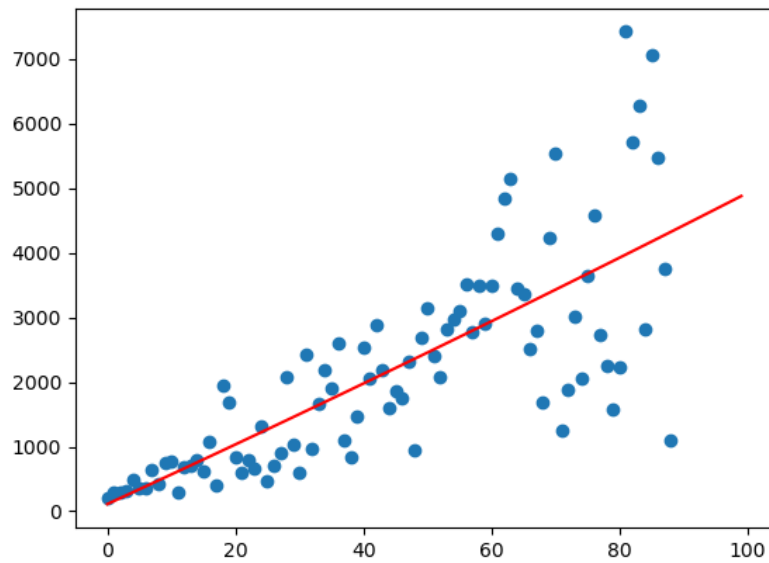
```
#I predict the revenue for the 120th day.
predict(120)
```

5824.7

```
#I fit the polinom regression model.
model = np.polyfit(x,y,2)
predict = np.poly1d(model)
model
```

array([2.53e-02, 4.57e+01, 1.10e+02])

```
#I fit the polinomial regression line.
x_lin_reg = range(0,89)
y_lin_reg = predict(x_lin_reg)
plt.scatter(x,y)
plt.plot(x_lin_reg, y_lin_reg, c = 'r')
plt.savefig('regression2.png')
plt.show()
```



```
#I predict the revenue for the 120th day.  
predict(120)
```

5954.0

According to the linear and polynomial models, Dilan's business revenue is projected to fall within the range of \$5,800 to \$6,000 on the 120th day. However, the R^2 score indicates that only half of the variation in the dependent variable is explained by the independent variables.