```
# imported required packages
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

from google.colab import files
```

```
#dataset file .csv format uploaded
uploaded = files.upload()
```

Choose Files AMZN_1Y_...a_latest.csv

- **AMZN_1Y_Historical_Data_latest.csv**(text/csv) - 13123 bytes, last modified: 2023-10-25 - 100% done

```
Saving AMZN_1Y_Historical_Data_latest.csv to AMZN_1Y_Historical_Data_latest.csv
```

```
#Organized and cleaned data as required for this small project
from os import rename
amzn_df = pd.read_csv('AMZN_1Y_Historical_Data_latest.csv')

amzn_df = amzn_df.rename(columns={'Close/Last' : 'Close_price'})

columns_to_process = ['Close_price','Open','High','Low']

for col in columns_to_process:
  amzn_df[col] = amzn_df[col].str.replace('$','').astype(float)

amzn_df.head(5)
```

```
<ipython-input-3-a02bcc20d377>:10: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will
  amzn_df[col] = amzn_df[col].str.replace('$','').astype(float)
```

	Date	Close_price	Volume	Open	High	Low
0	10/24/2023	128.56	46477360	127.740	128.8000	126.34
1	10/23/2023	126.56	48259950	124.630	127.8800	123.98
2	10/20/2023	125.17	56406410	128.050	128.1700	124.97
3	10/19/2023	128.40	60961360	130.565	132.2400	127.47
4	10/18/2023	128.13	42699480	129.900	130.6699	127.51

```
#split data size for model and used Linear Regression model
from scipy.sparse import random
df_close_price = amzn_df[['Close_price']]

future_days = 15
df_close_price['Prediction'] = df_close_price[['Close_price']].shift(-future_days)
X = np.array(df_close_price.drop(['Prediction'], 1))[:-future_days]
y = np.array(df_close_price['Prediction'])[:-future_days]
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

regression_model = LinearRegression()
regression_model.fit(X_train, y_train)
```

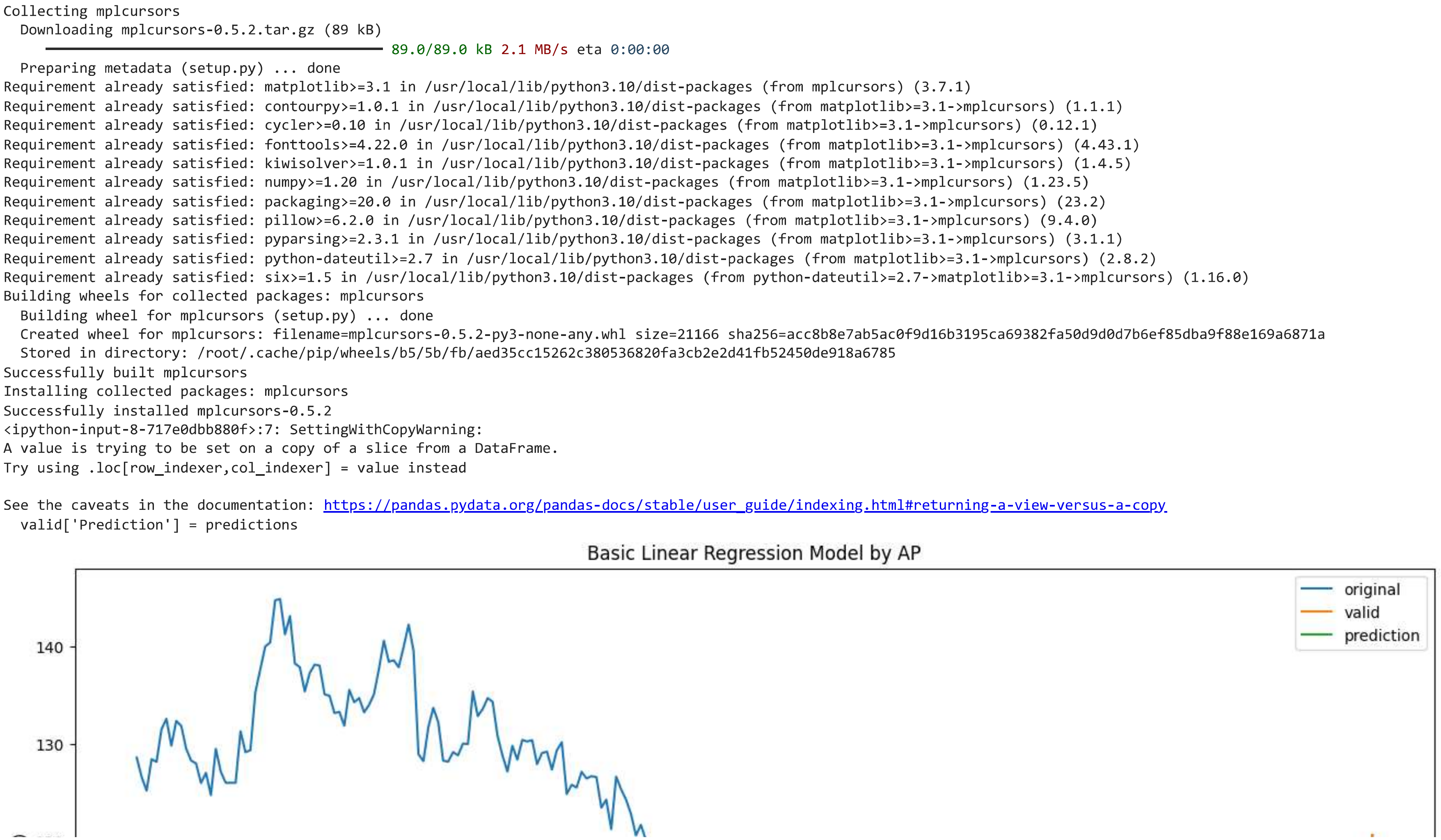
```
<ipython-input-4-45d636988e7d>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_close_price['Prediction'] = df_close_price[['Close_price']].shift(-future_days)
<ipython-input-4-45d636988e7d>:7: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only.
  X = np.array(df_close_price.drop(['Prediction'], 1))[:-future_days]
```

```
▾ LinearRegression
LinearRegression()
```

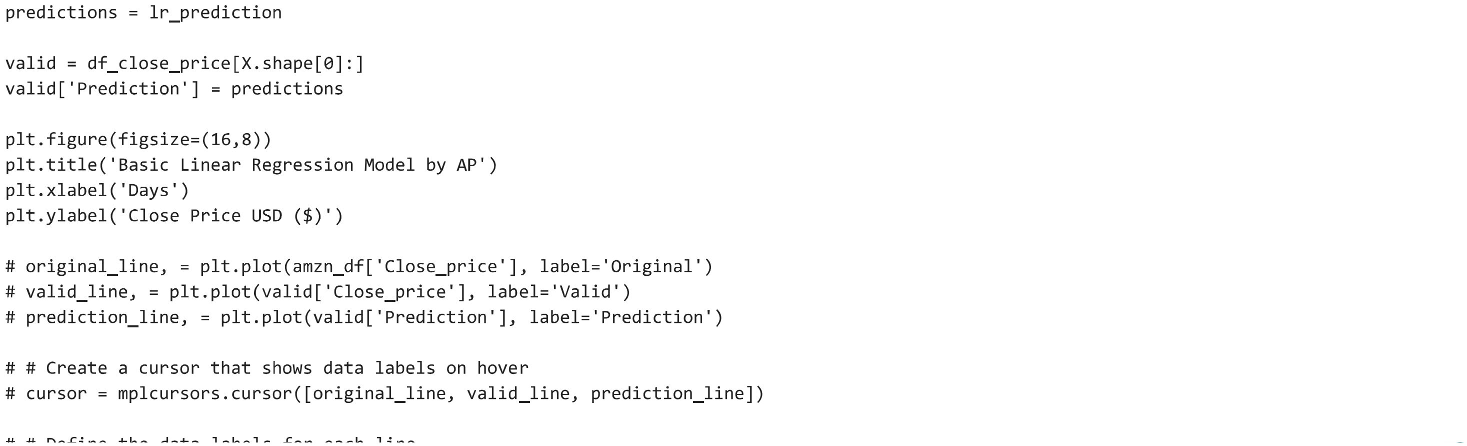
```
x_future = df_close_price.drop(['Prediction'], 1)[:-future_days]
x_future = x_future.tail(future_days)
x_future = np.array(x_future)
x_future
lr_prediction = regression_model.predict(x_future)
```

```
<ipython-input-5-4a6bdf325d14>:1: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only.
  x_future = df_close_price.drop(['Prediction'], 1)[:-future_days]
```

```
#Predictions and visualizing output
!pip install mplcursors
import mplcursors
```

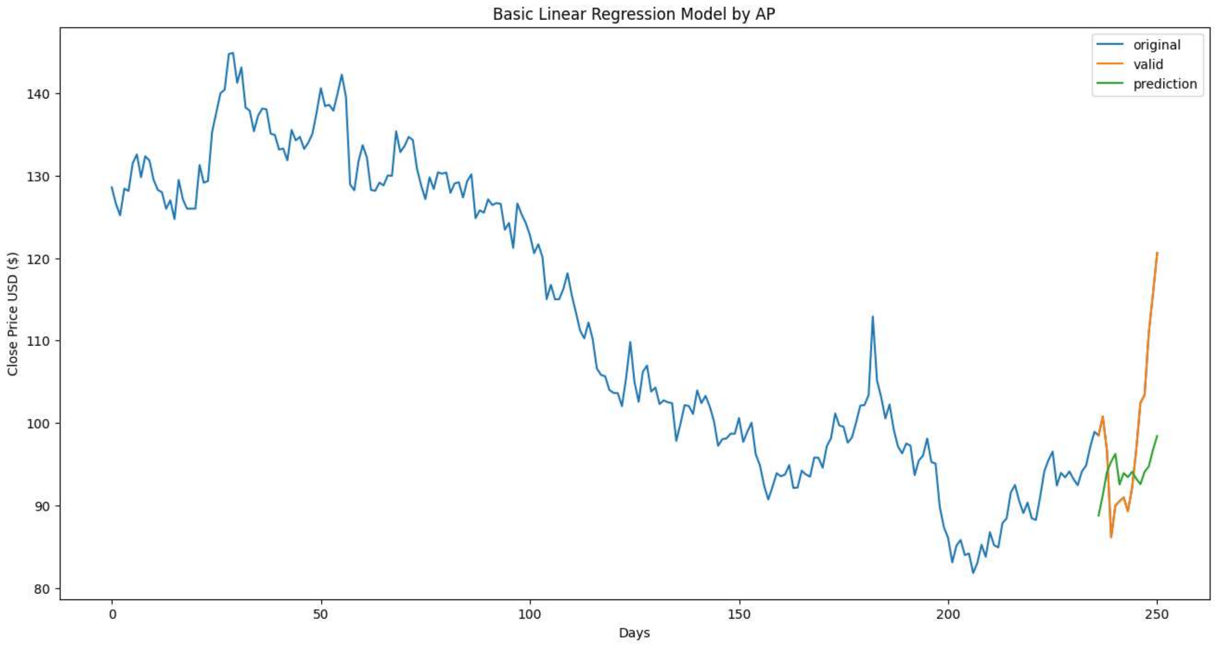


```
Collecting mplcursors
  Downloading mplcursors-0.5.2.tar.gz (89 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 89.0/89.0 kB 2.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: matplotlib>=3.1 in /usr/local/lib/python3.10/dist-packages (from mplcursors) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->mplcursors) (1.1.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->mplcursors) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->mplcursors) (4.43.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->mplcursors) (1.4.5)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->mplcursors) (1.23.5)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->mplcursors) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->mplcursors) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->mplcursors) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1->mplcursors) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib>=3.1->mplcursors) (1.16.0)
Building wheels for collected packages: mplcursors
  Building wheel for mplcursors (setup.py) ... done
  Created wheel for mplcursors: filename=mplcursors-0.5.2-py3-none-any.whl size=21166 sha256=acc8b8e7ab5ac0f9d16b3195ca69382fa50d9d0d7b6ef85dba9f88e169a6871a
  Stored in directory: /root/.cache/pip/wheels/b5/5b/fb/aed35cc15262c380536820fa3cb2e2d41fb52450de918a6785
Successfully built mplcursors
Installing collected packages: mplcursors
Successfully installed mplcursors-0.5.2
<ipython-input-8-717e0dbb880f>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  valid['Prediction'] = predictions
```



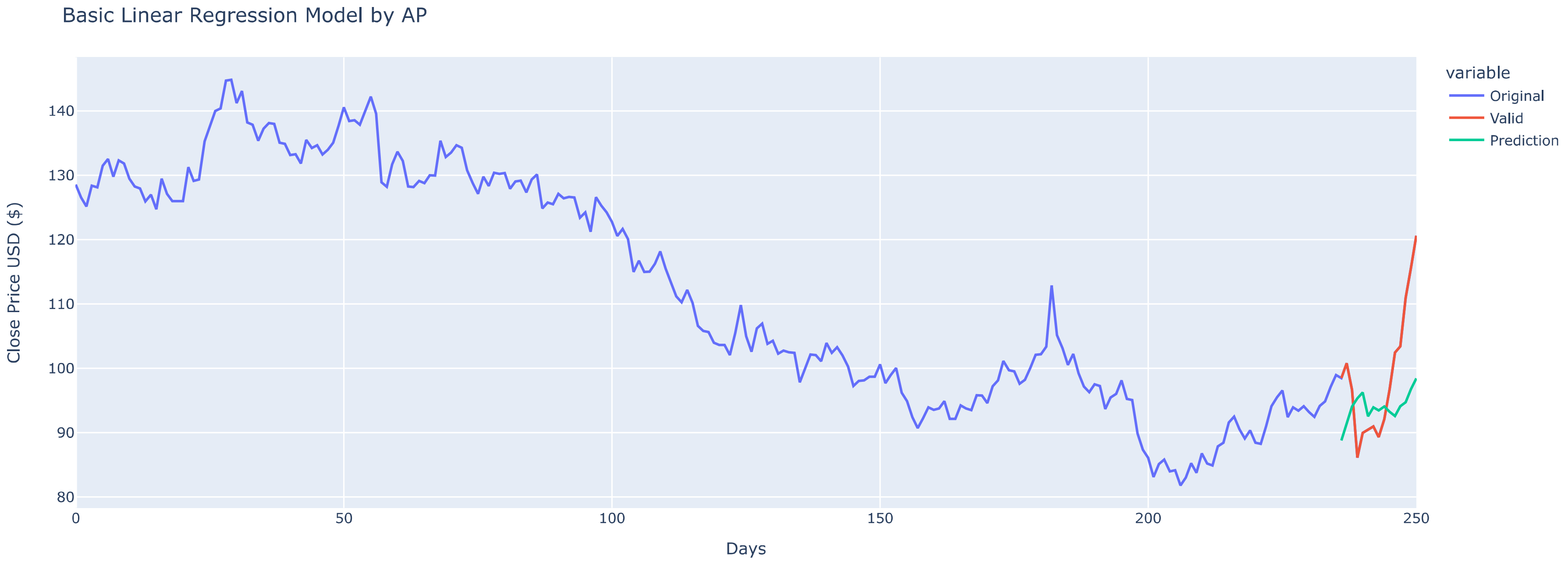
```
predictions = lr_prediction

valid = df_close_price[X.shape[0]:]
valid['Prediction'] = predictions

plt.figure(figsize=(16,8))
plt.title('Basic Linear Regression Model by AP')
plt.xlabel('Days')
plt.ylabel('Close Price USD ($)')

# original_line, = plt.plot(amzn_df['Close_price'], label='Original')
# valid_line, = plt.plot(valid['Close_price'], label='Valid')
# prediction_line, = plt.plot(valid['Prediction'], label='Prediction')

# # Create a cursor that shows data labels on hover
# cursor = mplcursors.cursor([original_line, valid_line, prediction_line])

# # Define the data labels for each line
# # Define the data labels for each line
# cursor.connect("add", lambda sel: sel.annotation.set_text(f"Value: {sel.target[1]:.2f}"))


plt.plot(amzn_df['Close_price'])
plt.plot(valid[['Close_price', 'Prediction']])
plt.legend(['original','valid','prediction'])
plt.show()
```

```
    <ipython-input-11-6b1d0bac6e2d>:4: SettingWithCopyWarning:


    A value is trying to be set on a copy of a slice from a DataFrame.
    Try using .loc[row_indexer,col_indexer] = value instead

    See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a
```



```
import plotly.express as px
data = pd.concat([amzn_df['Close_price'], valid[['Close_price', 'Prediction']]], axis=1)
data.columns = ['Original', 'Valid', 'Prediction']
```

```
fig = px.line(data, x=data.index, y=data.columns, labels={'value': 'Close Price USD ($)'})

fig.update_layout(
    title='Basic Linear Regression Model by AP',
    xaxis_title='Days',
    yaxis_title='Close Price USD ($)',
)

fig.show()
```



```
# Calculate the RMSE for accuracy of model
rmse = np.sqrt(mean_squared_error(valid['Close_price'], valid['Prediction']))
rmse
```

```
     11.516041039068071
```