Using BigQuery with Notebooks on AI Cloud Instance and Google Colab

BigQuery integrates well with Notebooks on Google Notebook AI Instance and Google Colab. In this section, we'll go through executing on BigQuery datasets and tables from Notebooks. There are a couple of ways to interact with BigQuery from Notebooks, but one quick and easy method is the use of the **'%bigquery'** magic command from the BigQuery client library, **'google-cloud-bigquery'**, to run queries with minimal syntax.

The **%%bigquery** magic runs a SQL query and returns the results as a pandas DataFrame. Here, we use the '%%bigquery' magic command to interact with BigQuery. To begin, open a Notebook on GCP AI Notebook Instance or from Colab:

1. If running on Google Colab, authenticate the notebook by running the code

   ```
   from google.colab import auth
   auth.authenticate_user()
   print('Authenticated')
   ```

2. Import Pandas and Matplotlib.

   ```
   import pandas as pd
   import matplotlib.pyplot as plt
   ```

3. Store the following query output as a Pandas DataFrame named **'litcoin_crypto'**. Place your project id after the **'--project'** attribute. Be sure to update the FROM field with your dataset and table IDs.

   ```
   %%bigquery --project ekabasandbox litcoin_crypto
   SELECT
     symbol,
     date,
     close,
     open,
     high,
     low,
     spread
   ```

```
FROM
  `crypto_data.markets`
WHERE
  symbol = 'LTC'
LIMIT 10
```

```
    symbol  date        close   open    high    low    spread
0   LTC 2013-04-28  4.35    4.3     4.4     4.18    0.22
1   LTC 2013-05-07  3.33    3.37    3.41    2.94    0.47
2   LTC 2013-05-03  3.04    3.39    3.45    2.4     1.05
3   LTC 2013-05-04  3.48    3.03    3.64    2.9     0.74
4   LTC 2013-05-05  3.59    3.49    3.69    3.35    0.34
5   LTC 2013-05-06  3.37    3.59    3.78    3.12    0.66
6   LTC 2013-05-02  3.37    3.78    4.04    3.01    1.03
7   LTC 2013-05-01  3.8     4.29    4.36    3.52    0.84
8   LTC 2013-04-29  4.38    4.37    4.57    4.23    0.34
9   LTC 2013-04-30  4.3     4.4     4.57    4.17    0.4
```

4. The variable ‘litcoin_crypto’ is a Pandas DataFrame. Now, let’s modify the data attributes and plot a bar chart.

```
# convert columns to numeric
litcoin_crypto = litcoin_crypto.apply(pd.to_numeric,
errors='ignore')

# check the datatypes
litcoin_crypto.dtypes
```

```
symbol      object
date        object
close      float64
open       float64
high       float64
low        float64
spread     float64
dtype: object
```

5. Plot the bar chart with the variable 'date' on the x axis and closing price on the y axis (see Figure 38-12).

```
# plot the bar chart
litcoin_crypto.plot(kind='bar', x='date', y='close')
plt.show()
```

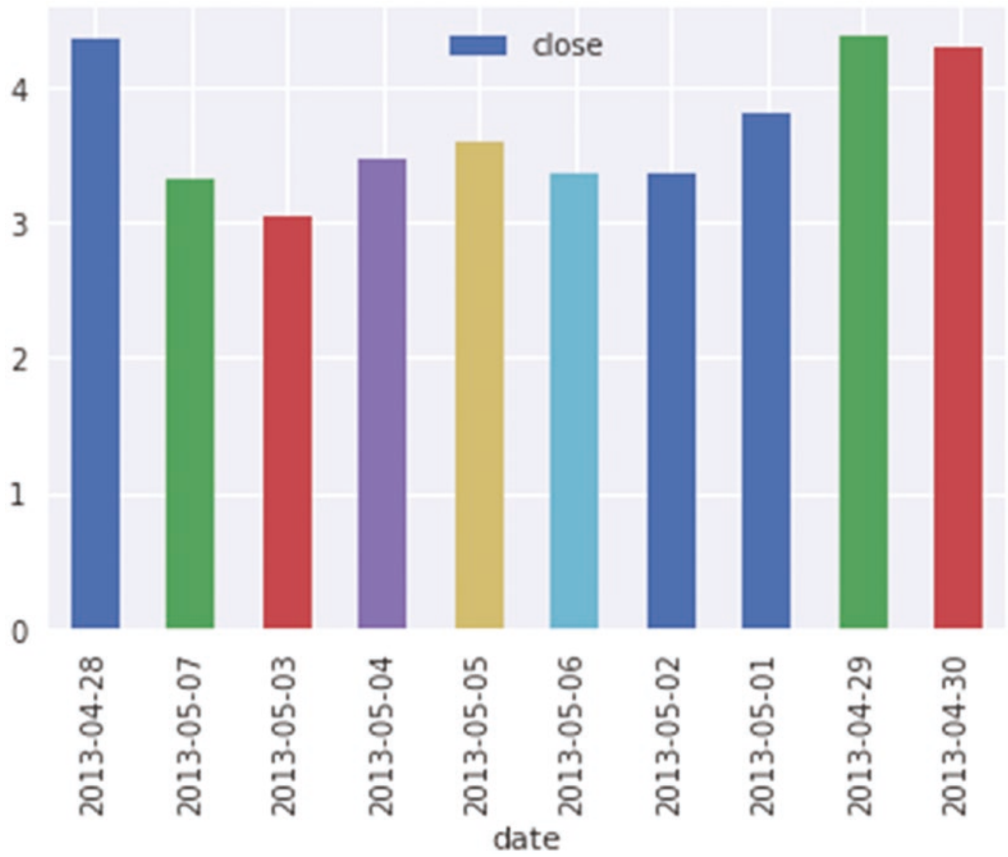

***Figure 38-12.** Litcoin crypto-currency bar chart plot*

BigQueryML

BigQuery machine learning makes it quick and easy to harness the power of machine learning on your datasets in BigQuery by using simple standard SQL commands. This functionality includes the capability to train and test models on the datasets by using subsets of the data, as well as the capability for automatic hyper-parameter tuning of the learning models.