1. **Price behaviors**
   The data here is the daily price information of bitcoin from 2017-07-01 to 2022-03-30, downloaded from Yahoo Finance. The price information consists of open, close, high and low price and the volume of bitcoin each day.
   The close price for bitcoin was relatively stationary from 2017-07-01 to 2020-11-01. After a steady growth of two months, the close price becomes relatively non-stationary from January 2021. overall the price has an increasing trend.

2. **Models**
   We use two models: one linear regression model to predict daily close price, the other is logistic regression model to classify daily return into three groups – positive, negative and close-to-zero. Both models are paying attention to the different price behaviors in the older period and the more recent period.

3. **Feature engineering**
   We extract several features for modeling. We use P_t to denote close price at day t.

   - logarithm of volume in previous day, denoted as $LogVol_{t-1}$
   - previous day's close price, denoted as P_{t-1}
   - previous k- day momentum of Close price, i.e $P_{t-1} - P_{t-k}$, denoted by r_k.
   - average of previous k-day return i.e $\frac{1}{k-1}(P_{t-1} - P_{t-k})/P_{t-k}$, denoted by $DReturn_k$.
   - average of previous k-day difference between high and low prices, denoted by d_k.
     $$d_k = \frac{1}{k}\sum_{i=1}^{k} \text{High}_{t-i} - \text{Low}_{t-i}$$
   - standard derivation of previous k - day close prices, denoted as s_k.
   - the day difference between day t and 2017-7-1, denoted as age.

     For the features involving k, the k's can be $2, 3, 7, 30, 70$.

4. **cross validation**
   We set apart the data after 2021-7-1 as cross validation set. The train/validation ratio is around 4.6.

5. **model details**

   - **Linear Regression**
     We use linear regression to predict close price P_t.

     a) feature selection using Lasso
     We group the train data into "stationary" and "non-stationary" using $s30$. If $s30 > 1800$, the close price is non-stationary. This division is consistent with the division before/after the date 2022-11-1.
     We use Lasso to conduct feature selections on each group.

b) model details
We divide above selected features into three parts.
X_c are the common informative features selected for both stationary and non-stationary data.
X_s (resp. X_{ns}) are the informative features selected only for stationary (resp. non-stationary) data.
P_s (resp. P_{ns}) is the close price for stationary (resp. non-stationary) data.

$$P_s = (X_c, 1) * \beta + X_s * \gamma_s + e_s \text{ for stationary data}$$

$$P_{ns} = (X_c, 1) * \beta + X_{ns} * \gamma_{ns} + e_{ns} \text{ for non-stationary data}$$

Here, β are fixed unknown parameters for both stationary and non-stationary data, γ_s (resp. γ_{ns}) are unknown parameters only for stationary (resp. non-stationary) data, the error terms are in different distribution for stationary and non-stationary data.

$$e_s \sim N(0, \sqrt{ws}\text{Id})$$

$$e_{ns} \sim N(0, \sqrt{wns}\text{Id})$$

here $\frac{1}{ws}$ (resp. $\frac{1}{wns}$) is the weight for stationary data (resp. non-stationary) data. The weights are linear proportional to the inverse of data size. Here non-stationary data has higher weights.

Written in matrix form:

$$P = \begin{pmatrix} P_s \\ P_{ns} \end{pmatrix} = \begin{pmatrix} (X_c & 1) & X_s & 0 \\ (X_c & 1) & 0 & X_{ns} \end{pmatrix} \begin{pmatrix} \beta \\ \gamma_s \\ \gamma_{ns} \end{pmatrix} + \begin{pmatrix} e_s \\ e_{ns} \end{pmatrix}$$

- **Logistic Regression Classification**
We directly work with daily return (DReturn defined as above). Daily return looks stationary overall, but ARIMA model seems not a good option here. Because the partial auto correlation and auto correlation are not strong.
we can divide daily return (DReturn) into three classes by setting two parameters $t1$ and $t2$. If $DReturn > t1$, it is class 2, meaning the return is relatively large positive; if $DReturn < t2$, it is class -2, meaning the return is relatively negative; if $DReturn$ is between $t1$ and $t2$, it is class 0, meaning the change is small close to 0.

a) **Choice of $t1$ and $t2$**
Again because of multicollinearity in all features, we conduct feature selection first via L1 regularized logistic regression.
For a fixed $t1$ and $t2$, use one-to-rest logistic regression to select important features for each class. Then use the union of these features in a logistic regression model.

From a range of $(t1, t2)$, select the one with best logistic regression model performance. For the details of measuring the performance, check the notebook.
The $(t1, t2)$ we choose here is $(0.015, -0.015)$.

6. **trading strategy**

   For regression models:

   - **long only**

     At the end of last day, we forecast next day's close price with the fitted regression models. If the forecast price is greater than previous close, we buy 1 unit; otherwise, nothing will be done.

   - **both short and long**

     Instead of doing nothing when predicted price is smaller than previous close, we go short 1 unit.

     In details:
     Long: if next day's predicted price > last day's close price:

     $$\text{daily return} = \frac{\text{next day's real price} - \text{last day's price}}{\text{last day's price}}$$

     Short: if next day's predicted price < last day's close price:

     $$\text{daily return} = \frac{\text{last day's price} - \text{next day's real price}}{\text{last day's price}}$$

   For classification model, there are also long only and short long version.
   For long version, if the predicted class is not -2 (here we choose predicted class not being -2 instead of predicted class being 2 is because the average return of class 0 is positive, we can still earn money if we buy on class 0 day), we buy 1 unit; For short, if the predicted class is -2, we go short 1 unit.

7. **model performances**
   We compute the daily return for different models on train and validation data using different trading strategies above.
   For the train data, we check the model performances on data before and after 2020-11-1 separately.

   Then we use the two measurements below

   - the average daily return
   - The Sharpe ratio
     Assume the risk free rate is 0:

     $$SR = \frac{E[R_p]}{\sigma(R_p)}$$

Here R_p is the daily return rate computed as above. The larger Sharpe ratio is the better the model performance is.

The baseline model to compare with is the daily return if we buy one unit everyday without any prediction.

Summary of model performances:
All the models perform better on train data than the baseline model. Especially on the more recent data (after 2020-11-1) using both short and long trading strategy. The regression models work better.
On more recent training data, using short long strategy the average daily returns for Baseline, weighted linear regression and logistic models are 0.049, 0.0078 and 0.0075.

But on the validation data set, the classification model is earning money but performs worse than the baseline model, while the regression model is losing money. The average daily return is 0.0018, -0.0031, 0.0003.

8. **future works**

   a) We have seen the models performs much better in train data than cross validation data. So we may update the model daily (or every k day) using all previous data before day t to predict day t's information. But this has a larger computational cost.

   b) We may combine the prediction results of different model together.