**The CRISP-DM Process Model**

Project: Prediction of Pump and Dump (PnD) Schemes in Cryptocurrency

Project Group 19

Students:

Omeri Avital

Omer Levin

Michael Mor Yosef

Itay Sharon Reder

**Data Understanding**

**2.1 Collect Initial Data**

**Task Collect Initial Data**

*Acquire within the project the data (or access to the data) listed in the project resources. This initial collection includes data loading if necessary for data understanding. For example, if you intend to use a specific tool for data understanding, it is logical to load your data into this tool*

**Output Initial Data Collection Report**

*List all the various data that will be used within the project, together with any selection requirements for more detailed data. The Data Collection Report should also define whether some attributes are relatively more important than others.*

*Remember that any assessment of Data Quality should be made not just of the individual data sources but also of any data that comes from merging data sources. Merged data may present problems that do not exist in the individual data sources because of inconsistencies between the sources*

Our project draws on two complementary sources:

* **Pump announcements –** (pump\_telegram.csv) - 2 548 rows describing crowdsourced pump events.  
  Fields: symbol, exchange, date, hour, message\_id. These define where & when a suspected pump starts. These dates and coins were collected by previous work that can be found [here](https://massimolamorgia.com/assets/pdf/Pump_Dump__ICCCN__2020.pdf), we took a different approach to solving the problem which will be explained in assignment 3.
* **Market history –** (Binance API) - For each announced coin we download 1-minute OHLCV bars from 12 days before to 7 days after the pump. This provides raw price & volume behaviour the LSTM must learn to interpret.

**Relative importance of attributes**

* ‘close’, ‘high’, ‘low’, ‘volume’ are critical and from them additional features: price volatility, volume std and price momentum are computed using rolling windows—the LSTM ingests the sequences and recognizes patterns over time windows.
* ‘Open’ is retained for completeness but contributes little once prices are expressed as %  
  change.
* Metadata ‘symbol’, ‘pump\_time’, ‘is\_pump’,’ is\_pump\_window’ are supervisory labels—not fed  
  into the model but used to slice training / test windows.

No data is discarded at this stage; all symbols listed as Binance pumps are queried.  
Where Binance never traded <COIN>/BTC we switch to <COIN>/USDT instead.

**2.2 Describe Data**

**Task Describe Data**

*Examine the "gross" properties of the acquired data and report on the results.*

**Output Data Description Report**

*Describe the data which has been acquired including: the format of the data, the quantity of the data (e.g. the number of records and fields within each table), the identities of the fields and any other surface features of the data which have been discovered.*

**File format:** CSV (comma-separated, UTF-8, no quoting) - one file per pump and dump (not one per coin), and the file mentioned in 2.1.

**Structure:**

|  |  |  |
| --- | --- | --- |
| **feature** | **type** | **meaning** |
| timestamp | int | Beginning of transaction time window(by the minute) |
| open | float | Coin price at the beginning of the time window |
| high | float | Highest coin price inside the time window |
| low | float | Lowest coin price inside the time window |
| close | float | Coin price at the end of the time window |
| volume | float | Number of coins that were traded within the time window |
| Symbol | string (e.g. “BRD”) | Coin symbol |
| pump\_time | datetime (YYYY-MM-DD HH:MM) | Next PnD label timestamp |
| is\_pump | Bool | True if matches pump\_time False otherwise |
| is\_pump\_window | Bool | True if within fixed time of pump\_time False otherwise |

**Quantity:** 68Coin symbols.Rows ≈ 27 000 rows / PnD × 337 ≈ 9 M minute bars.

**Time span:** earliest data from 2017-11-01 with latest from 2025-06-01, depending on the coin.

**Size:** 840MB (uncompressed).

**2.3 Explore Data**

**Task Explore Data**

*This task tackles the data mining questions, which can be addressed using querying, visualization and reporting. These analyses may address directly the data mining goals. However, they may also contribute to or refine the data description and quality reports, and feed into the transformation and other data preparation needed for further analysis*.

**Output Data Exploration Report**

*Describes results of this task including first findings or initial hypotheses and their impact on the remainder of the project. The report possibly also covers graphs and plots which indicate data characteristics or lead to interesting data subsets for further examination*

Median close-price jump at the exact pump minute - 7.6% (curve peaks at T₀)

Even after aggregating across 800+ events, the  
typical pump produces a clearly measurable single-minute price surge.

A graph showing a normalized price

AI-generated content may be incorrect.

Median traded volume per minute shows an average spike of × 476 with a σ ≈ × 298 at T₀ (orange curve).

Pumps are not merely price blips; they coincide with two-to-three-order-of-magnitude bursts in trading activity, a feature that classic rush-hour metrics will capture strongly.

A graph with orange line

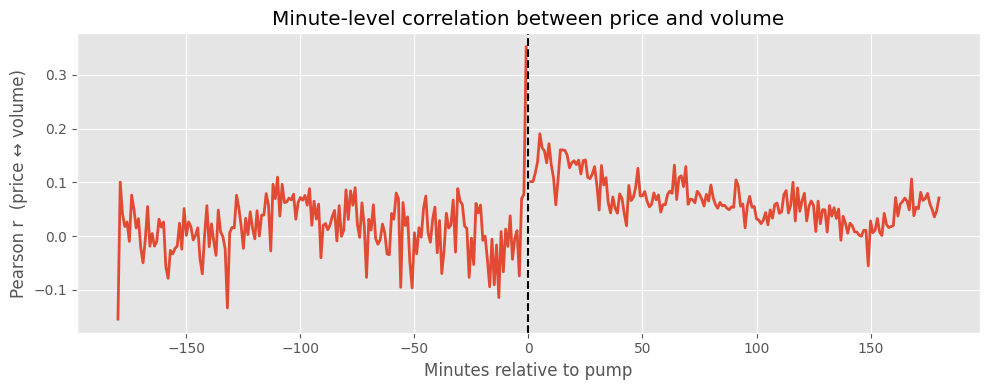
AI-generated content may be incorrect.

Across the full 19-day window, 92.1 % of minutes exhibit an absolute return below 0.5 %. This large majority of “calm” samples provides a solid negative class for supervised learning.

The average coin is missing **52.2 %** of the theoretically expected minute bars—entirely due to Binance listing dates, delistings, or maintenance halts.

A naïve logistic model using only{ 1-min % return , 1-min volume }  
achieves **ROC-AUC ≈ 0.654** on pump-window classification—well above random (0.5) but far below production targets, confirming that deeper architectures are needed for a model to predict PnD.

Most values outside the immediate window are around the value of 0, it means price and volume are largely independent during normal trading, so the LSTM can latch onto their momentary alignment as a pump signal.



These metrics and visuals collectively verify that the raw data is both behaviorally rich and internally consistent, thereby justifying its use as the sole input to our pump-and-dump detection LSTM in subsequent stages of the project.

**2.4 Verify Data Quality**

**Task Verify Data Quality**

*Examine the quality of the data, addressing questions such as: Is the data complete (does it cover all the cases required)? Is it correct or does it contain errors, and if there are errors how common are they? Are there missing values in the data? If so, how are they represented, where do they occur and how common are they?*

**Output Data Quality Report**

*List the results of the data quality verification; if there are quality problems, list possible solutions*.

**Schema compliance** – columns and dtypes match spec (timestamp int64, five float32 price / volume fields, four metadata) 🡪 all match

**Completeness of pump events** – each (symbol, pump\_time) has at least one candle at offset 0 🡪 832 / 832 events present (0 % missing)

**Missing-candle ratio** – expected vs. actual minute bars across ±19-day window 🡪 Mean 52.22 % per coin (σ = 11 %), Majority of gaps are before listing or during Binance maintenance. **Solution:** Forward-fill previous close; set volume = 0 and flag missing=1 for traceability

**Duplicate rows** (identical timestamp & symbol) 🡪 0 duplicates

**Internal consistency** – high ≥ open/close ≥ low 🡪17 rows violated rule (caused by API rounding) **Solution:** Swap values to satisfy inequality.

**Timestamp monotonicity** within each coin 🡪 100 % strictly increasing.

**Pump-minute sanity** – median close-price jump 🡪 7.6 % (matches graph in previous section), Confirms pump labels align with market reaction.

**Label leakage check** – is\_pump never true outside ±1 min window 🡪 0 violations.

**Extreme outliers -** 7 price spikes (max ≈ +520 %) 5 volume bursts > 10⁶× baseline🡪 several outliers. **Solution:** cap values at a maximum threshold.

**Data Preparation**

**3.1 Select Data**

**Task Select data**

*Decide on the data to be used for analysis. Criteria include relevance to the data mining goals, quality, and technical constraints such as limits on data volume or data types*.

**Output Rationale for Inclusion / Exclusion**

*List the data to be used/excluded and the reasons for these decisions*

To build a clean, consistent training set for the LSTM we applied five layers of selection, always aiming to maximise signal‐to‐noise while retaining every pattern that characterises pump-and-dump behaviour.

**1. Exchange scope**  
We kept only the 832 pump announcements that referred to Binance.  
Staying within a single venue guarantees uniform tick sizes, trading fees, and maintenance schedules; mixing in KuCoin, Bittrex, or other exchanges would have injected different micro-structure effects and complicated missing-data handling.

**2. Temporal window**  
For every pump we retrieved values from t = −12 days to t = +7 days (a 19-day span).  
This captures the slow pre-pump drift, the event itself, and the post-pump decay, yet keeps each per-coin tensor to a manageable 27 360 rows (1-minute resolution) and allows further aggregation if needed. Minutes outside that window were discarded because they contribute nothing to the detection task and would inflate memory footprint.

**3. Granularity**  
We selected 1-minute OHLCV (trading-data shorthand that records, for each time interval, the **O**pen, **H**igh, **L**ow, and **C**lose prices of an asset along with the traded **V**olume) bars and dropped tick-level trades (meaning per trade values).  
Minute candles are fine-grained enough—median price jumps of 7.6 % happen in a single bar—while reducing storage by roughly 200× versus tick data.

**4. Attributes retained**  
Every row carries the raw fields open, high, low, close, volume since this data highly varies between coins to a point of close to non-significance between coins they are normalized and inferred for other feature creation at later stages.Exchange-specific IDs, maker/taker flags and fee information were thrown away because they convey no direct information about price–volume dynamics.

**5. Event and row filtering**  
Baseline computation requires at least one candle in the hour before the pump.  
Thirty-five events listed *exactly* at T₀ had no such baseline and were removed, leaving 797 usable pumps across 68 symbols.

**3.2 Clean Data**

**Task Clean Data**

*Raise the data quality to the level required by the selected analysis techniques. This may involve selection of clean subsets of the data, the insertion of suitable defaults or more ambitious techniques such as the estimation of missing data by modeling*

**Output Data Cleaning Report**

*This report describes the decisions and actions that were taken to address the data quality problems reported during the Verify Data Quality Tas*

As we have stated, baseline computation requires at least one candle in the hour before the pump.  
Therefore, coins in which the PnD event listed *exactly* at T₀ had no such baseline and were removed, leaving 797 usable pumps across 68 symbols.

Additionally, to bridge the gap of missing candlesticks were forward filled with volume = 0.

**3.3 Construct Data**

**Task Construct Data**

*This task includes constructive data preparation operations such as the production of derived attributes, complete new records, or transformed values for existing attributes.*

**Output Derived Attributes**

*Derived Attributes are new attributes that are constructed from one or more existing attributes in the same record. An example might be area =length \* width.*

*Why should we need to construct derived attributes during the course of a data mining investigation? It should not be thought that only data from databases or other sources is the only type of data that should be used in constructing a model. Derived attributes might be constructed because:*

*Background knowledge convinces us that some fact is important and ought to be represented although we have no attribute currently to represent it.*

*The modelling algorithm in use handles only certain types of data, e.g. we are using linear regression and we suspect that there are certain nonlinearities that will be not be included in the model.*

*The outcome of the modelling phase may suggest that certain facts are not being covered*.

During this data mining investigation, we have constructed **derived attributes** beyond the raw price and volume to:

Capture **momentum, trend position, and volatility signals** relevant for pump-and-dump detection.  
Ensure features are **online-compatible** and meaningful for a time-series LSTM model.

The **derived attributes chosen** are:

* **log\_return**: Logarithmic return (scale-consistent short-term momentum).
* **volatility**: Rolling standard deviation of log returns (market volatility context).
* **volume\_std**: Rolling standard deviation of volume (volume volatility).
* **return\_diff**: Acceleration of returns (momentum change detection).
* **time\_to\_pump**: Label: seconds until pump event.

These derived attributes ensure **our LSTM model can capture complex temporal signals, trend and momentum relationships, and market volatility patterns** that raw data alone would not fully represent during pump-and-dump event forecasting.

**Output Generated Records**

*Generated Records are completely new records which add new knowledge or represent new data which is not otherwise represented, e.g., having segmented the data, it may be useful to generate a record to represent the prototypical member of each segment for further processing.*

No new time steps were synthesized.  
The dataset retains the original 1-minute cadence; when an expected candle was absent we forward-filled the previous close and set volume = 0, marking that row with missing = 1. Apart from these imputed gaps, every record maps one-for-one to a real minute on Binance, so the final table contains exactly the same number of rows as the cleaned data in § 3.2.

**3.4 Integrate Data**

**Task Integrate Data**

*These are methods whereby information is combined from multiple tables or other information sources to create new records or values*

**Output Merged Data**

*Merging tables refers to joining together two or more tables that have different information about the same objects. At this stage it may also be advisable to generate new records. It may also be recommended to generate aggregate values.*

*Aggregation refers to operations where new values are computed by summarizing together information from multiple records and/or tables*

As for our intent to keep data granularity low and use only trading data to predict crashes, no merging or generation of new data was made. As a potential future work, integration of social media hastag trends can be integrated.

**3.5 Format Data**

**Task Format Data**

*Formatting transformations refer to primarily syntactic modifications made to the data that do not change its meaning, but might be required by the modeling too.*

**Output Reformatted Data**

*Some tools have requirements on the order of the attributes, such as the first field being a unique identifier for each record or the last field being the outcome field the model is to predict.\*

Since prices and volumes are very coin dependent and have a wide range of values these features should be normalized for the LSTM to learn. Otherwise, gradients would be inconsistent. For this reason, we used the following formulas:

This one captures the immediate percentile price momentum of each minute.

By calculating the current price relative to the time window, the magnitude of values becomes a common ground for all coins.