* **Introduction**
  + Bitcoin is a digital currency, also known as cryptocurrency, that can be exchanged for goods or services with vendors who accept Bitcoin as payment.
  + It was introduced in 2009 by an anonymous creator known as Satoshi Nakamoto and is the first successful cryptocurrency and uses blockchain technology to secure and verify transactions.
  + The Bitcoin network is public and open-source, which means that anyone can participate.
  + In fact, unlike traditional currencies issued by central banks or governments, Bitcoin is decentralized and transactions take place directly between users through a peer-to-peer network, without the intervention of central authorities.
  + Transactions are authenticated through Bitcoin's proof-of-work consensus mechanism, which rewards cryptocurrency miners for validating transactions.
  + The total number of Bitcoins in circulation is limited to 21 million, which means it cannot be inflated or manipulated in any way and its price is determined by the exchange of supply and demand.
* **Goal**
  + Is it possible to do price forecasting of Bitcoin using machine learning methods in combination with the technical features of its blockchain?
* **Dataset**
  + The dataset containing the Bitcoin information is freely available on the Blockchain.com website
  + The dataset contains ~100k records ❓ describing from 2016 to the present ❓(with half-hour intervals) the most important features of Bitcoin
  + The features have been divided into 4 main categories:
    - Currency statistics: describe its price trend (e.g. market price, number of bitcoins in circulation...)
    - Block details: describe the technical characteristics of its blockchain (e.g. block size, number of transactions...)
    - Mining information: describe the characteristics of the consensus mode “Pow” (e.g. miners revenue, difficulty...)
    - Network activity: describe the actual use of Bitcoin as a method of exchange of value (e.g. number of transactions made, cost per transaction...)
* **Project pipeline**
  + The project is structured like this:
    - Data crawling: Bitcoin data retrieval via API call to Blockchain.com
    - Feature engineering: manipulation, visualization and extraction of features
    - Models train / validation: to train the models and evaluate them by performing hyperparameter tuning
    - Models testing: Test the models and compare the results to answer the initial question
  + Project carried out with Apache Spark (but during feature engineering I converted the Spark dataframe to a Pandas one to make some plots).
* **Data crawling / Feature engineering**
  + I simply make a call to the api to retrieve the data, check for null values, and save the dataset to disk
    - Since there are no missing values there was no need to process the data
    - Since there are no categorical features there was no need to encode the data
  + **Adding new features**
    - I decided to add some features that could help us predict the Bitcoin price:
      * Next market price: represents the price of Bitcoin for the next day (this will be the target variable on which to make predictions)
      * Rate of change: indicator that measures the percentage of price changes over a period of time, allows investors to spot security momentum and other trends
      * Simple Moving Averages: indicators that calculate the average price over a specified number of days. They are commonly used by traders to identify trends and potential buy or sell signals
  + **Dataset spit**
    - I decided to split the dataset into two sets:
    - Train / Validation set: will be used to train the models and validate the performances
    - Test set: will be used to perform price prediction on never-before-seen data (the last 3 months of the original dataset will be used).
  + **Feature selection**
    - Finally I decided to select features based on their correlation with the market price using the Pearson method.
    - They were divided into 3 groups to see the differences according to their use:
      * All: contains all features
      * Most correlated: contains features that have a correlation value > 0.7
      * Least correlated: contains the features that have a correlation value <= 0.7
* **Models train / validation**
  + Algorithms used
    - Linear regression
    - Generalized linear regression
    - Random forest regressor
    - Gradient boosting tree
  + Metrics used
    - RMSE, MSE, MAE, MAPE, R2, R2adj
  + **Pipeline**
    - In order to train and validate the model, I'll try several approaches:
      * Simple: Make predictions using the chosen base model
      * Simple with normalization: Like the previous one but features are normalized
    - At this point, the features that gave on average the most satisfactory results (for each model) are chosen and proceeded with:
      * Hyperparameter tuning: model validation to find the best parameters to use
      * Cross Validation: validate the performance of the model with the chosen parameters
        + **Note: explain how Walking Forward Cross Validation works and the voting system ❗**
      * Validate final model: validate the model with the chosen parameters
      * Train final model: train the final model on the whole train / validation set to be ready to make predictions on market price
  + **Results**
    - **❓**
* **Models testing** 
  + **Pipeline**
    - After loading the final previously trained models, the test set is divided into further mini-sets of 1 week, 15 days, 1 month and 3 months to see how the models' performance degrades as time increases.
  + **Results**
    - ❓
* **Conclusions**
  + Vantaggi del dataset time series
  + Considerazione finale sulle features utilizzate
  + Considerazioni finali sui risultati dei modelli
  + Risposta alla domanda iniziale
  + Lavori futuri
    - Utilizzare neural networks e compararli con algoritmi di ml normali