Financial Prediction Application

Christopher Ochs
Matthew Krauskopf
Philip Tallo
Kyle O’connor

Advisor: Professor Yizong Cheng

Abstract

To allow users to have better control over their finances, we will provide a web-based multifaceted application. The application will allow for end users to examine specific publicly traded stocks and get a prediction, with a specific degree of certainty, on whether to buy or sell a particular stock. This functionality will allow for users to manage their finances comfortably.

Goals

- Forecast stock prices based on prior data with a machine learning model

- Create an interface for the user to interact with financial data and machine learning models

- Create a testing framework so that any future development will meet any existing requirements

Intellectual Merits

- Implemented multiple APIs together to create a streamlined data acquisition pipeline that is constantly pulling the best data for the model.

- We are building on top of the SoA financial forecasting models and attempting to optimize the parameters of the model in order to break current performance thresholds.

- Design of a web application that allows users to quickly understand the predicted trends of different financial forecasting models.

Broader Impacts

- Reduces the difficulty threshold for the novice investor to enter the market with confidence and security.

- Allows small investment firms access to the latest financial forecasting technology to better level the playing field with firms that have a massive capitol advantage.

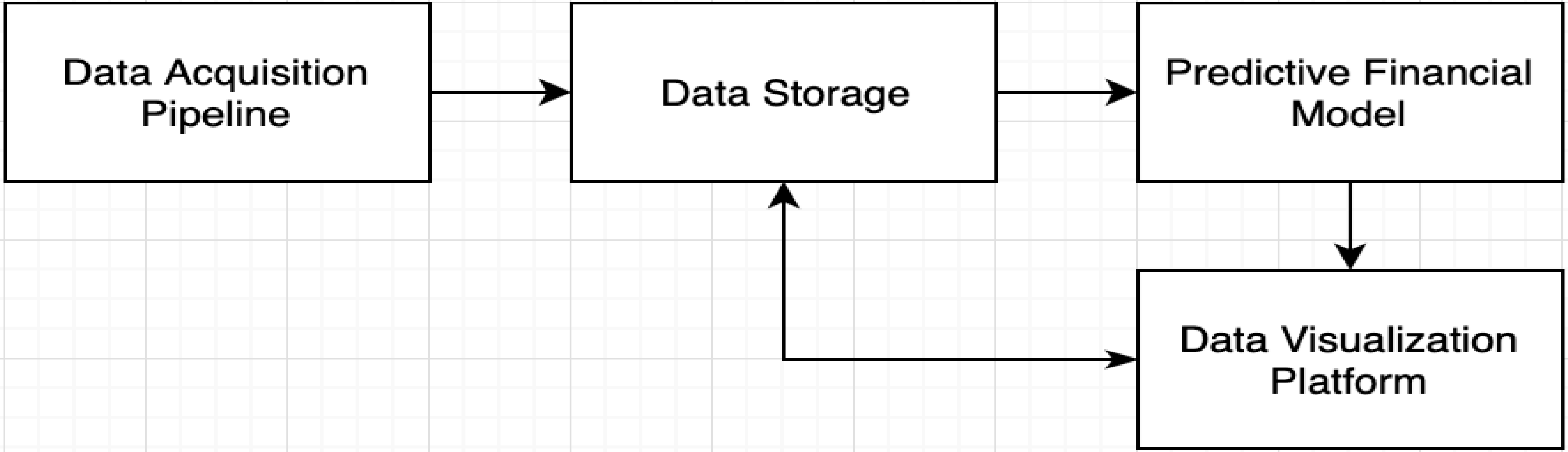
- Bringing machine learning models into the public domain instead of proprietary models owned by private firms

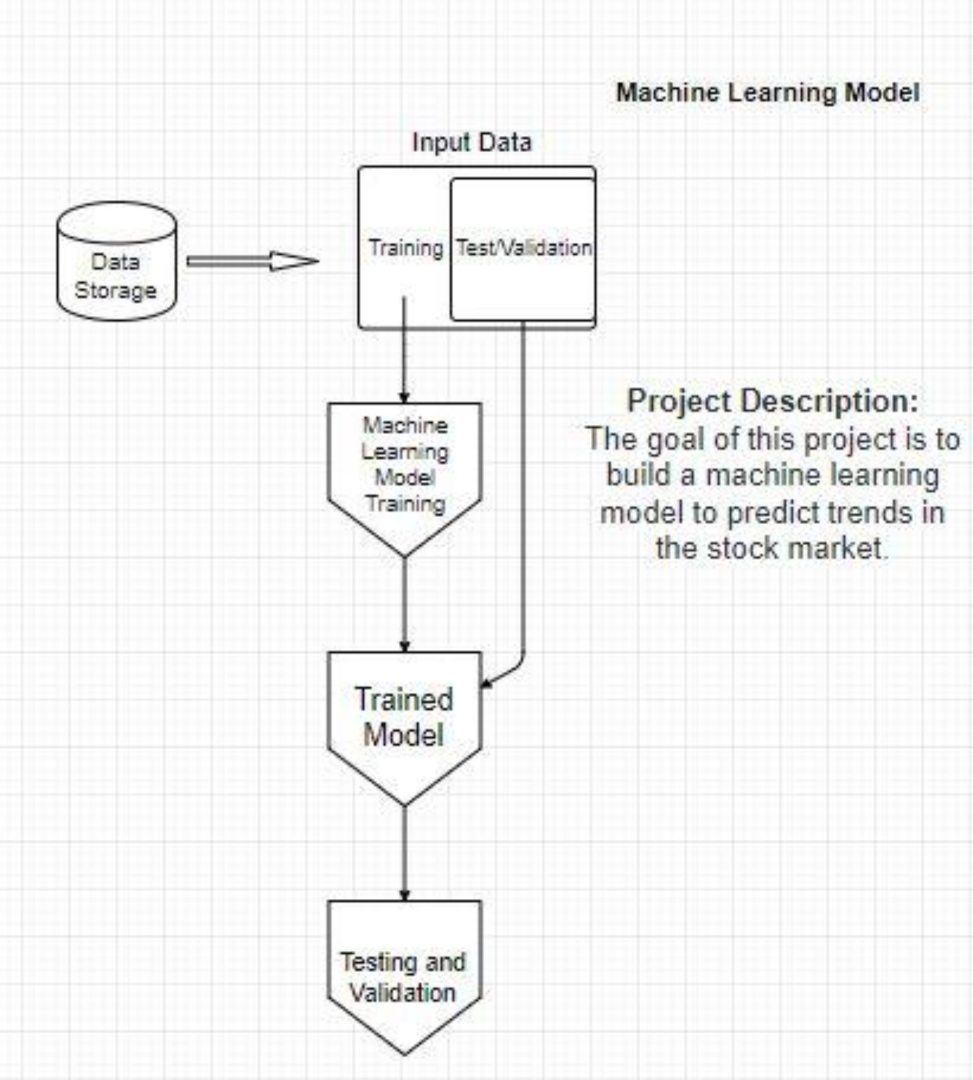
Design Specifications

Financial Predictive Modeling

The goal of this project is to build a machine learning model to predict trends in the stock market.

Design Level 0 : High Level Overview, arrows in this diagram represent the way data flows through our system


Data Acquisition Pipeline
Data Storage
Predictive Financial Model
Data Visualization Platform



Machine Learning Model

Project Description:

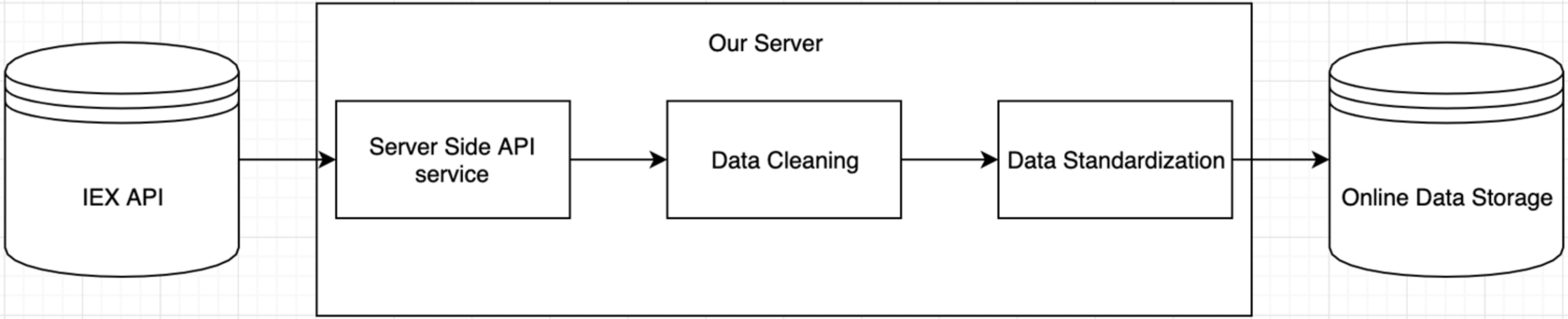
The goal of this project is to build a machine learning model to predict trends in the stock market.

Data Acquisition Pipeline

The goal of this project is to build a machine learning model to predict trends in the stock market.

Design Level 1 : High Level Overview, arrows in this diagram represent the way data flows through our system

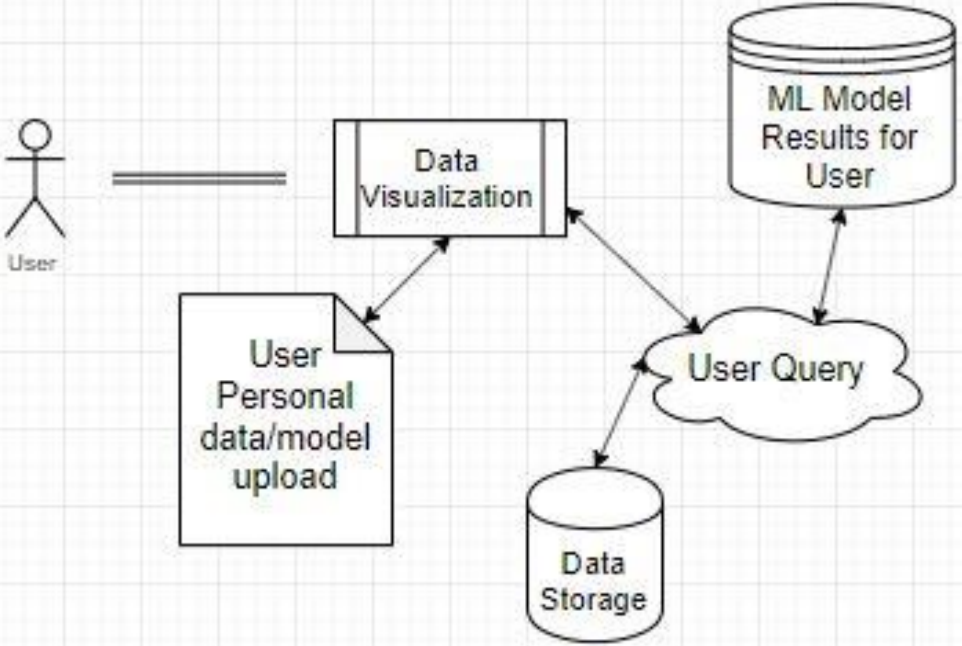
Note: This data acquisition pipeline will be running continuously for two purposes. To train our model to be more accurate, as well as to supply our data visualization platform with all the data it needs.

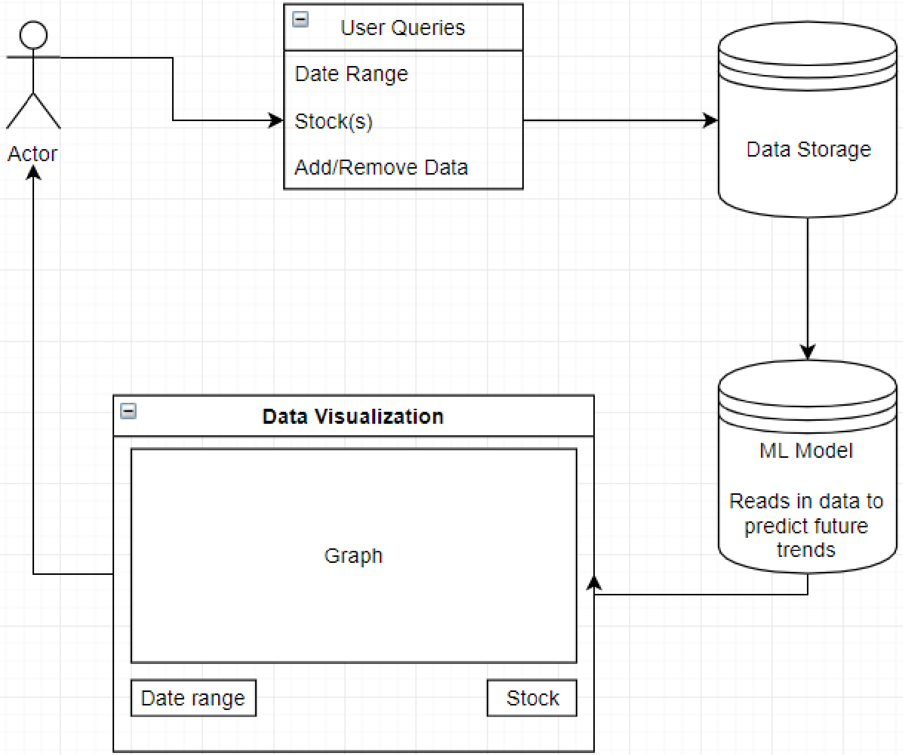


Data Visualization

Project Description:

The goal of this project is to build a machine learning model to predict trends in the stock market.





User Queries: User will be able to query for certain stocks (including general DOW), what date range they want, and be able to exclude date ranges from model consideration.

Data Visualization: Display graphs with respective dates/value, and show where stock is expected to trend as predicted by our model.

Technologies Used

Python: Flask - for the

Heroku - Platform as a Service

Keras/Tensorflow - ML model development

React/Redux - Front-end web application

Project Milestones (1)

- ~~(12/14 - Kyle) Setup data storage in the Cloud~~
- ~~(12/14 - Phil) Develop the Server Side API Service~~
- ~~(1/13 - Chris) Develop the data cleaning and standardization methods~~
- ~~(1/21 - Matt) Acquire the data to train the model~~
- ~~(1/21 - Matt) Set up a continuous flow of data into the model~~
- ~~(2/21 - Kyle & Phil) Refine ANN architecture~~
- ~~(2/21 - Kyle) Train the model~~
- ~~(2/28 - Kyle) Refine the hyper parameters of the model~~
- ~~(3/21 - Chris) Develop a continuous integration pipeline for deployment~~

Project Milestones (2)

- ~~(3/21 - Phil) Design the UI for the website~~
- ~~(3/21 - Phil) Develop the web app~~
- ~~(3/28 - Chris) Integrate the model into the web app~~
- ~~(3/28 - Chris) Develop a query system for the User~~

From 3/28 -> 4/14, refine everything we have.

Results

- Fully functioning web server
- Pipeline for data collection
- Models to perform transfer learning on
- Light visualization of webapp (needs polishing)

Challenges

- There are numerous charges that APIs and Databases require. We have found work arounds these issues, but when product is released we can simply pay for these services.

- Long training times for starting from nothing to create model. We worked around this by using transfer learning on a pretrained model.

Thank You

Christopher Ochs - ochscs@mail.uc.edu

Matthew Krauskopf - krauskmr@mail.uc.edu

Philip Tallo - tallopt@mail.uc.edu

Kyle O’Connor - oconnokb@mail.uc.edu