Financial Data Acquisition for Predictive Analysis

W205 Final Project

Ben Attix, Jeffrey Hsu, Shane Conner

Domain Background & Proposal

- Using freely available financial data to predict stock market performance
- Build historical dataset to explore and then fit machine learning algorithm on top
- Use machine learning model to apply findings to new data from current market
- Three V’s
    - Variety: Five different datasets
    - Volume: 8GB stored on HDFS
    - Velocity: One time loading of historical data. Working on real-time loading

Data Acquisition

- Original plan was to scrape data from multiple sources (Finviz, Yahoo! Finance, etc.)
  - Data quality issues

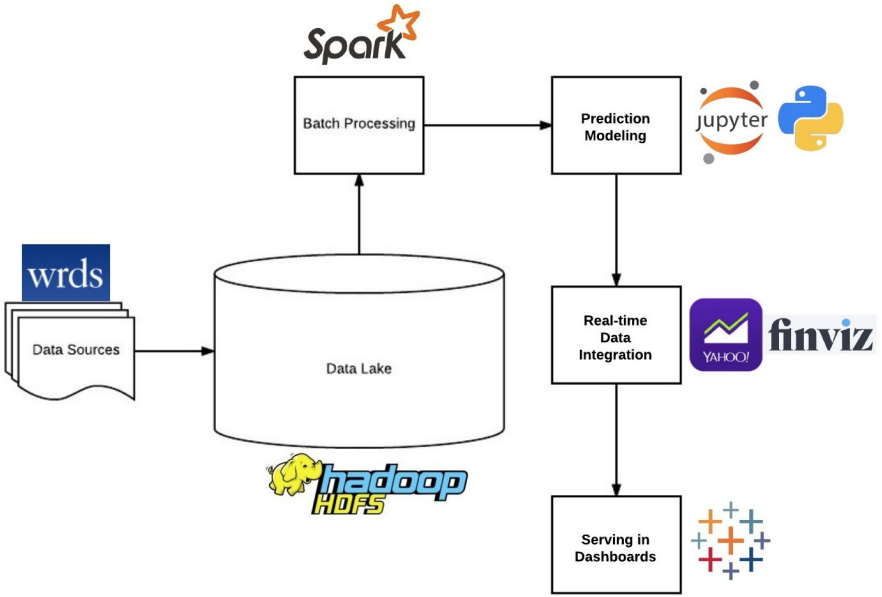
- Wharton Research Data Services (WRDS) makes vast amount of historical data available to students
  - Command line access via SSH
  - Browser-based access

- Datasets Used:
  1. CRSP Compustat
  2. Financial Ratios
  3. Beta Suite
  4. Recommendations
  5. Linking Table

Data Lake Architecture

Requirements

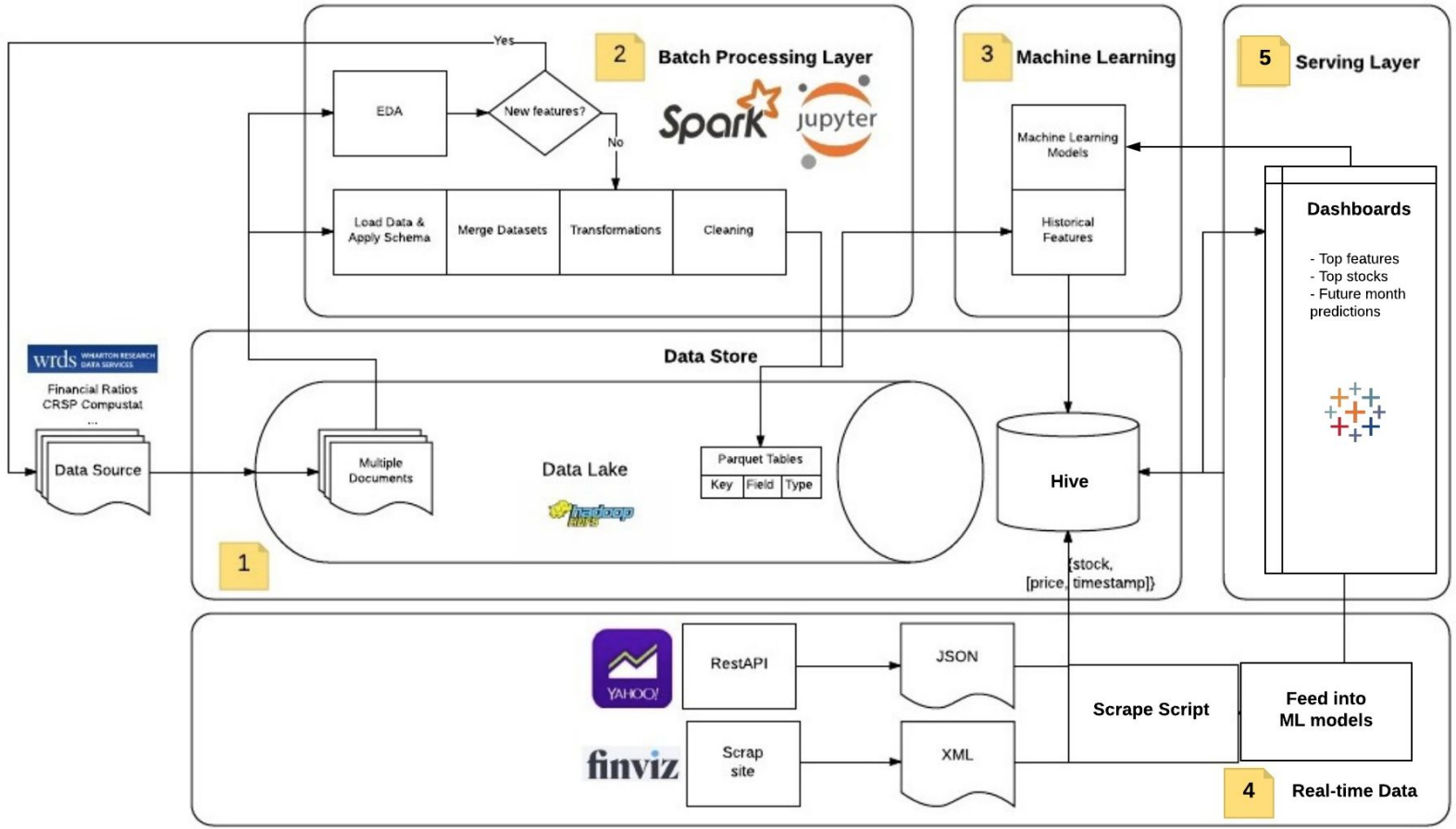
- Flexibility on handling
  - different data types
  - different data structures
  - new data sources
- Large amount of features
- Scalability



Data Pipelines

- 1. Ingest raw datasets into data lake
- 2. Apply schema and transformation in batches
- 3. Train ML via structured historical stock data
- 4. Serve result in dashboards and queries
- 5. Retrieve real-time stock performance

Architecture Detail



Batch Processing with Pyspark Dataframe

- Apply schema on read and create data frames

```
sparkcsv = "com.databricks.spark.csv"
financial_suite = sqlContext.read.format(sparkcsv)
        .options(header='true')
        .load(financial_suite_path, schema=schema_fin_suite)
```

- Merge datasets by stock ticker

```
df = fin_suite
        .join(link_table, fin_suite.gvkey == link_table.GVKEY, 'leftouter')
        .drop(link_table.GVKEY).dropDuplicates()
```

- Create unique key

```
df = df.withColumn('GVKEY_year_mth'
        , concat(
                col('gvkey'), lit('-'), getYear(col('public_date')),
                lit('-'), getMonth(col('public_date'))))
```

- Group by stock & order by date

```
w = Window().partitionBy(col("GVKEY"))
        .orderBy(col("GVKEY_year_mth"))
```

- Add historical stock price

```
df = df.withColumn(
        First_new_col,
        lag(col(variable),
        -month,None).over(w))
```

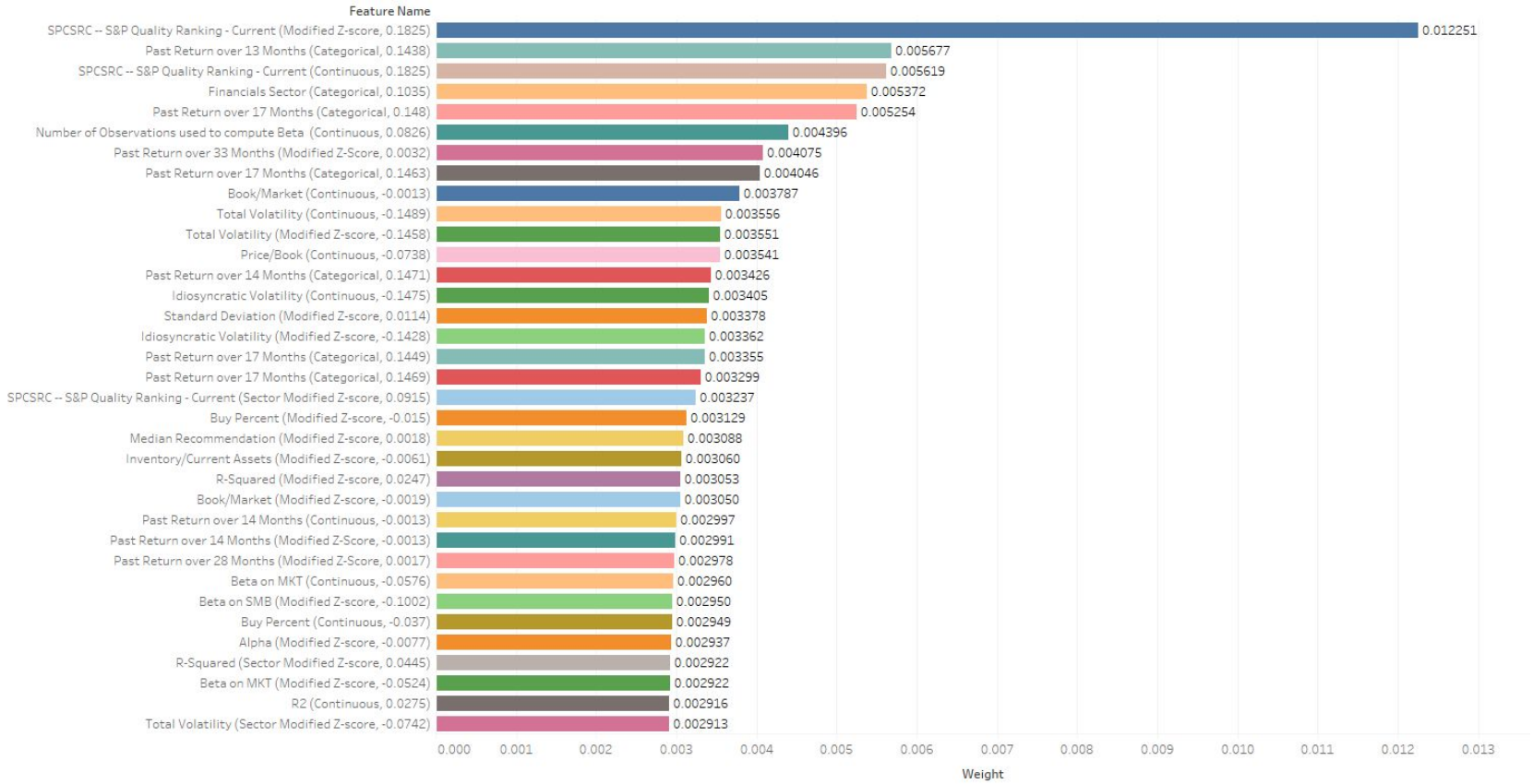
Technical challenges

- Find good data sources
- Sheer amount of data and trying to make sense of it
- Handling nulls
- Installing Spark 2.0 for Machine Learning
- Learning to use Spark dataframes
- Fitting an accurate model

Data Management and Techniques

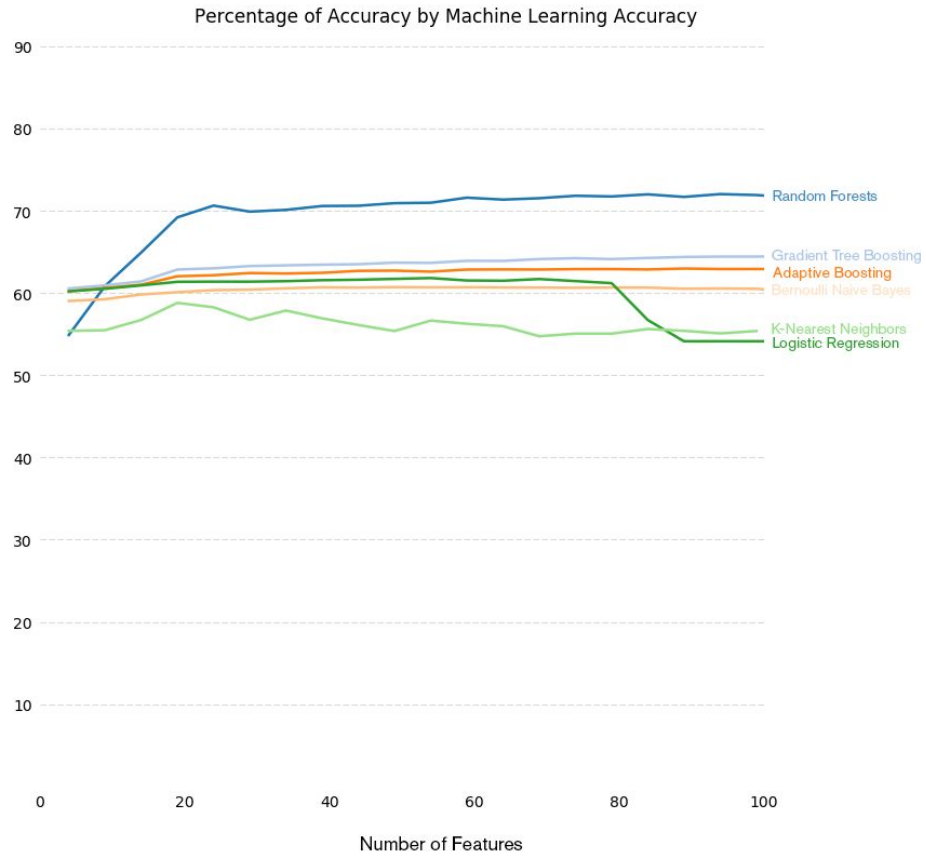
- **Relative Features**: implemented modified Z-score - a more robust measure of a security's variables relative to current day market and sector.
- **Mitigating Outliers**: reduced outliers by trimming back each variable's bottom and top 5% for current day market and sector.
- **Handling Nulls**: maximized observations by forward and backfilling nulls by security, sector, and market whole in that respective order.
- **Categorical:** added binary values (dummy variables) to quantify categorical data such as sector and month of year.

Feature Results



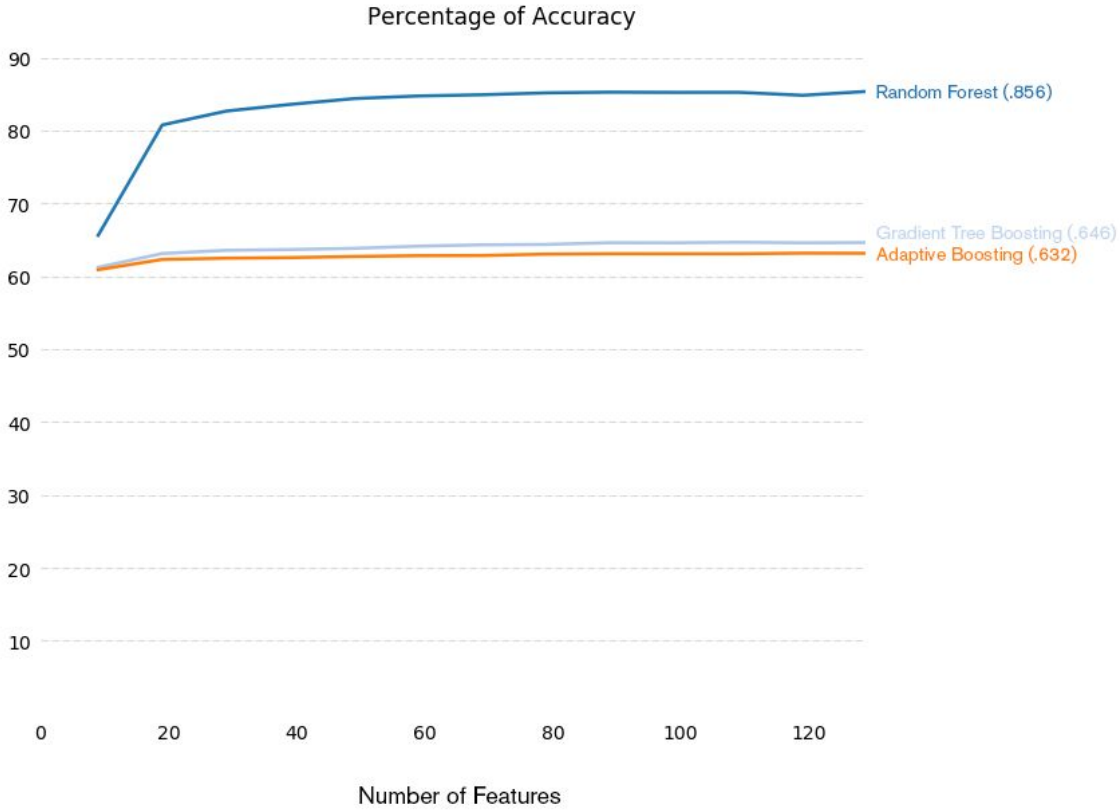
Machine Learning Algorithms

- Classification
  - Logistic Regression
  - Bernoulli Naive Bayes
  - Random Forests
  - Adaptive Boosting
  - Gradient Tree Boosting
- Magnitude
  - Linear Regression
  - Ridge Regression



Selected Model Results

- Random Forests
  - Accuracy: **0.856**
  - Log Loss: **0.522**
- Adaptive Boosting
  - Accuracy: 0.632
  - Log Loss: 0.691
- Gradient Tree Boosting
  - Accuracy: 0.646
  - Log Loss: 0.628



Limitations and Extensions

- Limitations
  - Reliable source of real-time data
  - Unknowns of accuracy
  - Historic lackluster results
  - Efficient Market Hypothesis
- Extensions
  - Additional features and observations
  - Streaming of predictions
  - Comparison of predicted outcomes across dependent variables

References

- Wharton Data Services: https://wrds-web.wharton.upenn.edu/wrds/index.cfm
- Project Github Repo: https://github.com/jeffrey-hsu/W205_Project
- Dataframe documentations with Databricks: https://docs.databricks.com/spark/latest/dataframes-datasets/index.html#
- Finviz: http://finviz.com/