STOCK VALUE PREDICTION

• Siddhant Sanjeev (15BM6JP45)
• Loveleen Kaur (15BM6JP21)
• Ankit Sonthalia (15BM6JP06)
All characters and events depicted in this project are entirely modeled. Any similarity to actual events or stock price movements is purely awesome.
OUTLINE

• Implementing probabilistic and statistical models

• Making inferences from models

• Computing integration of models

• Creating database of predictions
STOCKS AND MARKET DATA

http://finance.yahoo.com/
q?s=DATA
Fits the $n^{th}$ degree polynomial on data points
Here $n$ is chosen to be 5

$\text{input1} = (1: \text{length})$;
$\text{input2} = \text{input1}^2$;
$\text{input3} = \text{input1}^3$;
$\text{input4} = \text{input1}^4$;
$\text{input5} = \text{input1}^5$;

$\text{tsstock} = \text{ts}(\text{st})$;
$\text{lmstock} = \text{tslm}(\text{tsstock} \sim \text{input1} + \text{input2} + \text{input3} + \text{input4} + \text{input5})$
SES MODEL

tsstock=ts(st);
sesstock=ses(tsstock)

sesstock$fitted
#Forecasts

sesstock$upper
#High Confidence Interval

sesstock$lower
#Low Confidence Interval

\[ \hat{y}_{T+1|T} = \alpha y_T + \alpha (1-\alpha) y_{T-1} + \alpha (1-\alpha)^2 y_{T-2} + \ldots \]
\[ \hat{y}_{T+1|T} = \alpha y_T + (1-\alpha) \hat{y}_{T|T-1} \]

\( \alpha = \) Smoothing Parameter
Weights decrease exponentially

**R Code Snippet**

**Forecast with 95% C.I.**
**HOLT-WINTERS MODEL**

```
tsstock=ts(st);

hwstock=HoltWinters(tsstock,
gamma=FALSE)

forecast = predict(hwstock,
n.ahead = 100,
prediction.interval=T,
level = 0.95)

Holt_Wintrs_Fit=hwstock$fitted
```

**R Code Snippet**

**Holt-Winters Model**

Breaks data into level(a), trend(b) and seasonality(s)

\[ Y[t+h] = a[t] + h \times b[t] + s[t - p + 1 + (h - 1) \text{ mod } p] \]

\[ a[t] = \alpha (Y[t] - s[t-p]) + (1-\alpha) (a[t-1] + b[t-1]) \]

\[ b[t] = \beta (a[t] - a[t-1]) + (1-\beta) b[t-1] \]

\[ s[t] = \gamma (Y[t] - a[t]) + (1-\gamma) s[t-p] \]
**FEED FORWARD MODEL**

Artificial Neural Network
No direction cycle
Flow of info in 1 direction: Forward

```r
  tsstock = ts(st);
  nnstock = nnetar(tsstock);
  predict = forecast(nnstock);
  predict$mean[1]  # Predicts the value
```

**R Code Snippet**
ARIMA MODEL

R Code Snippet

```r
tsstock = ts(st);
arstock = arima(tsstock, order = c(15, 3, 3), method = c("CSS"));
pred = forecast(arstock);
pred$mean[1]
```

**ARIMA Coefficients**

<table>
<thead>
<tr>
<th>ar1</th>
<th>-1.59241</th>
<th>ar10</th>
<th>-0.894651</th>
</tr>
</thead>
<tbody>
<tr>
<td>ar2</td>
<td>-2.392885</td>
<td>ar11</td>
<td>-0.611519</td>
</tr>
<tr>
<td>ar3</td>
<td>-2.312248</td>
<td>ar12</td>
<td>-0.393537</td>
</tr>
<tr>
<td>ar4</td>
<td>-2.180835</td>
<td>ar13</td>
<td>-0.213871</td>
</tr>
<tr>
<td>ar5</td>
<td>-2.162685</td>
<td>ar14</td>
<td>-0.086129</td>
</tr>
<tr>
<td>ar6</td>
<td>-1.891472</td>
<td>ar15</td>
<td>-0.00816</td>
</tr>
<tr>
<td>ar7</td>
<td>-1.684254</td>
<td>ma1</td>
<td>-0.336171</td>
</tr>
<tr>
<td>ar8</td>
<td>-1.394506</td>
<td>ma2</td>
<td>0.254239</td>
</tr>
<tr>
<td>ar9</td>
<td>-1.157487</td>
<td>ma3</td>
<td>-0.958154</td>
</tr>
</tbody>
</table>

**Autoregressive Integrated Moving Average**

\[ X_t - \phi_1 X_{t-1} - \ldots - \phi_p X_{t-p} = W_t + \theta_1 W_{t-1} + \ldots + \theta_q W_{t-q} \]

Difference between \( d \)th order ARMA is ARIMA
The scatter plot between market and stock value is plotted. Linear regression is calculated on it.

R Code Snippet

```r
mar = as.numeric(k + 1);
for (j in (i-k):i-1)
{
    mar[j-(i-k)+1] = market[j];
}

corstock = lm(stock ~ mar)
corstock = ts(corstock)
pred = forecast(corstock$fitted.values)
```
The scatter plot between stock value at time $t_n$ and $t_{n-1}$ is plotted. Linear regression is calculated on it.

\textbf{R Code Snippet}

```
for(j in 2:k)
    temp[i-1]=st[j];

value[p]=cor(temp,st)
p=p+1;

corstock=ts(lm(st~temp))
pred=forecast(corstock$fitted.values)
```

\textbf{Autocorrelation Values}

```
• **Correlation Table** provides a linear correlation between the predicted value and real stock values.
• Higher the Autocorrelation better the estimated values.

<table>
<thead>
<tr>
<th>Model No</th>
<th>Model Name</th>
<th>Correlation values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Polynomial Model</td>
<td>0.9462837</td>
</tr>
<tr>
<td>2</td>
<td>Holt-Winters Model</td>
<td>0.9956402</td>
</tr>
<tr>
<td>3</td>
<td>Feed Forward Model</td>
<td>0.9587529</td>
</tr>
<tr>
<td>4</td>
<td>ARIMA Model</td>
<td>0.9951617</td>
</tr>
<tr>
<td>5</td>
<td>SES Model</td>
<td>0.9957231</td>
</tr>
<tr>
<td>6</td>
<td>Market-Stock Correlation</td>
<td>0.9273835</td>
</tr>
<tr>
<td>7</td>
<td>Stock Autocorrelation</td>
<td>0.9957109</td>
</tr>
</tbody>
</table>
COMPUTING INTEGRATION OF MODELS

- Weighted average of the predicted models

Applying different set of weights to the selected models.
Calculation of Chi-square value and selecting appropriate weight for the prediction

C Code Snippet
The isotonic regression finds a non-decreasing approximation of a function while minimizing the mean squared error on the training data.

R Code Snippet

Minimize $\sum_i w_i(y_i - \hat{y}_i)^2$
Subject to $\hat{y}_{min} = \hat{y}_1 \leq \hat{y}_2 \ldots \leq \hat{y}_n = \hat{y}_{max}$

- `x1=isoreg(db[,1],db[,2])`
- `x2=isoreg(db[,3],db[,4])`
- `x3=isoreg(db[,5],db[,6])`
- `x4=db[,7]`
- `y1=isoreg(x1$y,x2$y)`
- `y2=isoreg(x3$y,x4)`
- `z=isoreg(y1$y,y2$y)`
### CHI-SQUARE VALUES

- Chi-squared = \( \frac{(\text{observed}-\text{expected})^2}{\text{expected}} \)
- Lower the value of Chi square higher the chances of success.

<table>
<thead>
<tr>
<th>Model No</th>
<th>Model Name</th>
<th>Chi-square Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Polynomial Model</td>
<td>117.732204</td>
</tr>
<tr>
<td>2</td>
<td>Holt-Winters Model</td>
<td>8.199687</td>
</tr>
<tr>
<td>3</td>
<td>Feed Forward Model</td>
<td>203.591952</td>
</tr>
<tr>
<td>4</td>
<td>ARIMA Model</td>
<td>8.941534</td>
</tr>
<tr>
<td>5</td>
<td>SES Model</td>
<td>8.132946</td>
</tr>
<tr>
<td>6</td>
<td>Market-Stock Correlation</td>
<td>153.113115</td>
</tr>
<tr>
<td>7</td>
<td>Stock Autocorrelation</td>
<td>8.396684</td>
</tr>
<tr>
<td>8</td>
<td>Weighted average of predicted Models</td>
<td>8.051282</td>
</tr>
<tr>
<td>9</td>
<td>Isotonic Regression</td>
<td>8.466684</td>
</tr>
</tbody>
</table>
Disadvantages of RF

• Random forest is observed to overfit for datasets with noisy classification/regression tasks.

• Random forest is more helpful for the classification of the data and not for the prediction of the continuous data.

• For data including categorical variables with different number of levels, random forests are biased in favor of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data.
NEWS SENTIMENT ANALYSIS TO PREDICT STOCK MARKET TRENDS
GET RICH!

Acquiring The Text Streams:

Few current streams of news items acquired from Google and Yahoo:

- GoogleFinanceSource ("NYSE:ACN")
- GoogleNewsSource ("ACN")
- YahooFinanceSource ("ACN")
- YahooInplaySource()
- YahooNewsSource ("ACN")

Usage of webCorpus function for turning them into corpus objects.

3769 News stories captured today by the same (documents).
Breaking documents down into Sentences

- Do the sentiment scoring at sentence level
- Extract Relevant sentences.

Focusing on relevant sentences instead of tagging the entire news story

Apple Inc. (AAPL) sank 2.8 percent, the most since October, to $605.23. After rising to a record on April 9, the most valuable technology company fell for a fourth day in the longest losing streak since December.

Coinstar Inc. (CSTR) surged 7.3 percent to $65.78. The owner of the Redbox movie-rental kiosks said first-quarter sales and profit exceeded its previous projection and lifted its earnings forecast for 2012 to at least $4.40 a share.
Forming Document Term Matrix out of Corpora

- Scrubbing
- Merging
- Deleting
- Converting

Document Term Matrix
Identifying Polarity of Words

Use Harvard General Inquirer to create list of sentiment words

Scoring Text Corpus

For Example, the sentence:

- “ACN continues its phenomenal run” is a positive sentence as count(positive) = 2 and count(negative) = 0
- “Cracks develop in PCLN” is negative heading as count(positive) = 0 and count(negative) = 1
Visualization

POSITIVE WORD CLOUD

NEGATIVE WORD CLOUD
Interpreting the Sentiment Score

Sentiment score should fall in the range 0-100%

- 23 negative terms occurring in the sentences
- 107 positive terms occurring in the sentences
- 89 sentences with sign +1
- 10 sentences with sign -1
- A naive sentiment score of $\frac{89}{89 + 10} = \frac{89}{99} \approx 89.89\%$

Sentiment Score = Positive instances / Total instances
Thank You