Predictive Analytics Models, Generative A.I., and Blockchain

Dr. Reza Barkhi
Reza@vt.edu

Analytics Mindset
&
Decision Context
Analyze - Four Types of Analytics

- Descriptive – What happened?
- Diagnostic – Why did it happen?
- Predictive – What will happen?
- Prescriptive – How can we make it happen?
Business Understanding

**DEFINE CRITICAL SUCCESS FACTOR (CSF)?**

**WHAT ARE SEVERAL CSF FOR MCDONALDS?**

**KNOW YOUR INDUSTRY RATIOS AND TRENDS**

**MACRO ECONOMIC IMPACTS**
Modeling & Evaluation
Relationships &
Simple Linear Regression

Simple Linear Regression Model
### Excerpt From Cereals Data Set: Eight Fields, First 16 Cereals

<table>
<thead>
<tr>
<th>Cereal Name</th>
<th>Manufacture</th>
<th>Sugars</th>
<th>Calories</th>
<th>Protein</th>
<th>Fat</th>
<th>Sodium</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>100% Bran</td>
<td>N</td>
<td>6</td>
<td>70</td>
<td>4</td>
<td>1</td>
<td>130</td>
<td>68.4030</td>
</tr>
<tr>
<td>100% Natural Bran</td>
<td>Q</td>
<td>8</td>
<td>120</td>
<td>3</td>
<td>5</td>
<td>15</td>
<td>33.9837</td>
</tr>
<tr>
<td>All-Bran</td>
<td>K</td>
<td>5</td>
<td>70</td>
<td>4</td>
<td>1</td>
<td>260</td>
<td>59.4255</td>
</tr>
<tr>
<td>All-Bran Extra Fiber</td>
<td>K</td>
<td>0</td>
<td>50</td>
<td>4</td>
<td>0</td>
<td>140</td>
<td>93.7049</td>
</tr>
<tr>
<td>Almond Delight</td>
<td>R</td>
<td>8</td>
<td>110</td>
<td>2</td>
<td>2</td>
<td>200</td>
<td>34.3848</td>
</tr>
<tr>
<td>Apple Cinnamon Cheerios</td>
<td>G</td>
<td>10</td>
<td>110</td>
<td>2</td>
<td>2</td>
<td>180</td>
<td>29.5095</td>
</tr>
<tr>
<td>Apple Jacks</td>
<td>K</td>
<td>14</td>
<td>110</td>
<td>2</td>
<td>0</td>
<td>125</td>
<td>33.1741</td>
</tr>
</tbody>
</table>

**We are interested in estimating the nutritional rating of a cereal, given its sugar content.**

The prediction was too high by 57.3916 \( - 50.765 = 6.6266 \) rating points!

This vertical distance of 6.6266 rating points, in general, is known variously as the **prediction error**, **estimation error**, or **residual**.

The regression line is written in the form: \( \hat{y} = b_0 + b_1x \), called the regression equation, where:
- \( \hat{y} \) is the estimated value of the response variable;
- \( b_0 \) is the y-intercept of the regression line;
- \( b_1 \) is the slope of the regression line;
- \( b_0 \) and \( b_1 \), together, are called the **regression coefficients**.

The estimated cereal rating equals 59.953 minus 2.4614 times the sugar content in grams.

\[ \hat{y} = 59.853 - 2.4614x \]

We of course seek to minimize the overall size of our prediction errors. Least squares regression works by choosing the unique regression line that minimizes the sum of squared residuals over all the data points.
Another Example

Relationship between:

Travel Time
and
Distance
The ANOVA Table for Simple Linear Regression

<table>
<thead>
<tr>
<th>Source</th>
<th>Sum of Squares</th>
<th>Degrees of Freedom</th>
<th>Mean Square</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>SSR</td>
<td>m</td>
<td>MSR = (\frac{SSR}{m})</td>
<td>F = (\frac{MSR}{MSE})</td>
</tr>
<tr>
<td>Error (or residual)</td>
<td>SSE</td>
<td>n - m - 1</td>
<td>MSE = (\frac{SSE}{n - m - 1})</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>SST = SSR + SSE</td>
<td>n - 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The regression equation is
Distance = 6.00 + 2.00 Time

**Analysis of Variance**

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>SS</th>
<th>MS</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>1</td>
<td>216.00</td>
<td>216.00</td>
<td>144.00</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>8</td>
<td>12.00</td>
<td>1.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>9</td>
<td>228.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Predictor Coef SE Coef T P**

| Constant        | 6.0000 | 0.9189 | 6.53 | 0.000 |
| Time            | 2.0000 | 0.1667 | 12.00 | 0.000 |

\[ S = 1.22474 \quad R-Sq = 94.7\% \quad R-Sq(adj) = 94.1\% \]

A high leverage point is an observation that is extreme in the predictor space.

High Leverage Point

A high leverage point is an observation that is extreme in the predictor space.

Predicted Values for New Observations

<table>
<thead>
<tr>
<th>New Obs</th>
<th>Fit SE Fit</th>
<th>95% CI</th>
<th>95% PI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18.091</td>
<td>0.352 (17.294, 18.888)</td>
<td>(15.329, 20.853)</td>
</tr>
</tbody>
</table>
An observation is influential if the regression parameters alter significantly based on the presence or absence of the observation in the data set.
R Statistical Package

• A statistical computer program made available through the Internet under the General Public License (GPL).

• R works fundamentally by the question-and-answer model:
  
  You enter a command and press Enter,
  and
  R gives you the Output

• Need to learn how to interpret Output from R.

Regression in R

```r
> attach(thuesen)
> summary(lm(short.velocity~blood.glucose))

Call:
  lm(formula = short.velocity ~ blood.glucose)

Residuals:
     Min       1Q   Median       3Q      Max
-0.40141 -0.14760 -0.02202  0.03001  0.43490

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.09781    0.11748   9.345 6.26e-09 ***
blood.glucose 0.02196    0.01045   2.101 0.0479    

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2167 on 21 degrees of freedom
(1 observation deleted due to missingness)

Multiple R-squared: 0.1737,    Adjusted R-squared: 0.1343
F-statistic: 4.414 on 1 and 21 DF,    p-value: 0.0479
```

```r
> plot(blood.glucose,short.velocity)
> abline(lm(short.velocity~blood.glucose))
```
Correlation in R

```r
> cor(thuesen,use="complete.obs")

          Blood.glucose short.velocity
blood.glucose 1.0000000    0.4167546
short.velocity 0.4167546    1.0000000
```

```r
> cor.test(blood.glucose,short.velocity)

Pearson's product-moment correlation

Data: blood.glucose and short.velocity

```

```
t = 2.101, df = 21, p-value = 0.0479

Alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.005496682 0.707429479

Sample estimates:

```
cor
0.4167546
```

Multiple Regression Model
Multiple Regression & Model Building

Excerpt From Cereals Data Set: Eight Fields, First 16 Cereals

<table>
<thead>
<tr>
<th>Cereal Name</th>
<th>Manufacture</th>
<th>Sugars</th>
<th>Calories</th>
<th>Protein</th>
<th>Fat</th>
<th>Sodium</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>100% Bran</td>
<td>N</td>
<td>6</td>
<td>70</td>
<td>4</td>
<td>1</td>
<td>130</td>
<td>68.4030</td>
</tr>
<tr>
<td>100% Natural Bran</td>
<td>Q</td>
<td>8</td>
<td>120</td>
<td>3</td>
<td>5</td>
<td>15</td>
<td>33.9837</td>
</tr>
<tr>
<td>All-Bran</td>
<td>K</td>
<td>5</td>
<td>70</td>
<td>4</td>
<td>1</td>
<td>260</td>
<td>59.4255</td>
</tr>
<tr>
<td>All-Bran Extra Fiber</td>
<td>K</td>
<td>0</td>
<td>50</td>
<td>4</td>
<td>0</td>
<td>140</td>
<td>93.7049</td>
</tr>
<tr>
<td>Almond Delight</td>
<td>R</td>
<td>8</td>
<td>110</td>
<td>2</td>
<td>2</td>
<td>200</td>
<td>34.3848</td>
</tr>
<tr>
<td>Apple Cinnamon Cheerios</td>
<td>G</td>
<td>10</td>
<td>110</td>
<td>2</td>
<td>2</td>
<td>180</td>
<td>29.5095</td>
</tr>
<tr>
<td>Apple Jacks</td>
<td>K</td>
<td>14</td>
<td>110</td>
<td>2</td>
<td>0</td>
<td>125</td>
<td>33.1741</td>
</tr>
</tbody>
</table>
In simple linear regression, we used a straight line (of dimension 1) to approximate the relationship between the response and one predictor.

A multiple regression model uses a linear surface, such as a plane or hyperplane, to approximate the relationship between a continuous response (target) variable, and a set of predictor variables. Categorical predictor variables may be included through the use of indicator (dummy) variables.

The estimated nutritional rating equals 52.174 minus 2.2436 times the grams of sugar plus 2.8665 times the grams of fiber.

To interpret $b_1 = -2.2436$, we say that “the estimated decrease in nutritional rating for a unit increase in sugar content is 2.2436 points, when fiber content is held constant.”

To interpret $b_2 = 2.8665$ as follows: “the estimated increase in nutritional rating for a unit increase in fiber content is 2.8408 points, when sugar content is held constant.”
Recall that errors in prediction are measured by the residual.

\[ SSE = \sum (y - \hat{y})^2 \]

\[ y - \hat{y} = 72.8018 - 60.7735 = 12.0283 \]

Sum of Squares Error = an overall measure of the error in prediction resulting from the use of the estimated regression equation.

\[ SSR = \sum (\hat{y} - \bar{y})^2 \]

Total Sum of Squares = a measure of the total variability in the values of the response variable alone, without reference to the predictor.

\[ SST = \sum (y - \bar{y})^2 \]

Sum of Squares Regression = a measure of the overall improvement in prediction accuracy when using the regression as opposed to ignoring the predictor information.

\[ R^2 = \frac{SSR}{SST} \]

For multiple regression, \( R^2 \) is interpreted as the proportion of the variability in the target variable that is accounted for by its linear relationship with the set of predictor variables.

**The regression equation is**

\[ Rating = 52.2 - 2.24 \text{ Sugars} + 2.87 \text{ Fiber} \]

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Coef</th>
<th>SE Coef</th>
<th>T</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>52.174</td>
<td>1.556</td>
<td>33.54</td>
<td>0.000</td>
</tr>
<tr>
<td>Sugars</td>
<td>-2.2436</td>
<td>0.1632</td>
<td>-13.75</td>
<td>0.000</td>
</tr>
<tr>
<td>Fiber</td>
<td>2.8665</td>
<td>0.2979</td>
<td>9.62</td>
<td>0.000</td>
</tr>
</tbody>
</table>

\[ S = 6.12733 \quad R^2 = 81.6\% \quad R^2(\text{adj}) = 81.1\% \]

**Analysis of Variance**

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Sum of Squares</th>
<th>Degrees of Freedom</th>
<th>Mean Square</th>
<th>F</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>12188.6</td>
<td>2</td>
<td>6094.3</td>
<td>162.32</td>
<td>0.000</td>
</tr>
<tr>
<td>Residual Error</td>
<td>2740.7</td>
<td>73</td>
<td>37.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>14929.3</td>
<td>75</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Source**

- Regression
- Error (or residual)
- Total

**Mean**

- \( MSR = \frac{SSR}{m} \)
- \( MSE = \frac{SSE}{n-m-1} \)

Whenever a new predictor variable is added to the model, the value of \( R^2 \) always goes up. If the new variable is useful, the value of \( R^2 \) will increase significantly; if the new variable is not useful, the value of \( R^2 \) may barely increase at all.

If the new variable is useful, then \( s \) will decrease, but if the new variable is not useful for predicting the target variable, then \( s \) may in fact increase. This type of behavior makes \( s \), the standard error of the estimate, a more attractive indicator than \( R^2 \) of whether a new variable should be added to the model, because \( R^2 \) always increases when a new variable is added, regardless of its usefulness.
F-Test for the Significance of the Overall Regression Model

One may apply a separate t-test for each predictor \(x_1, x_2, \text{ or } x_3\), examining whether a linear relationship exists between the target variable \(y\) and that particular predictor. However, the F-test considers the linear relationship between the target variable \(y\) and the set of predictors (e.g., \(\{x_1, x_2, x_3\}\)), taken as a whole.

Multiple Regression in R

The data are in the cystfibr data frame in the ISwR package.

```r
> attach(cystfibr)
> summary(lm(pemax~age+sex+height+weight+bmp+fev1+rv+frc+tlc))
```

Call:
```
lm(formula = pemax ~ age + sex + height + weight + bmp + fev1 + rv + frc + tlc)
```

Residuals:
```
Min      1Q  Median      3Q     Max
-37.338  -11.532   1.081   13.386  33.405
```

Coefficients:
```
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) 176.0582    225.8912  0.779 0.448
```

27

28
Multiple Regression in R

|        | Estimate | Std. Error | t value | Pr(>|t|) |
|--------|----------|------------|---------|----------|
| age    | -2.5420  | 4.8017     | -0.529  | 0.604    |
| sex    | -3.7368  | 15.4598    | -0.242  | 0.812    |
| height | -0.4463  | 0.9034     | -0.494  | 0.628    |
| weight | 2.9928   | 2.0080     | 1.490   | 0.157    |
| bmp    | -1.7449  | 1.1552     | -1.510  | 0.152    |
| fev1   | 1.0807   | 1.0809     | 1.000   | 0.333    |
| rv     | 0.1970   | 0.1962     | 1.004   | 0.331    |
| frc    | -0.3084  | 0.4924     | -0.626  | 0.540    |
| tlc    | 0.1886   | 0.4997     | 0.377   | 0.711    |

Residual standard error: 25.47 on 15 degrees of freedom
Multiple R-squared: 0.6373, Adjusted R-squared: 0.4197
F-statistic: 2.929 on 9 and 15 DF, p-value: 0.03195

Multiple Regression in R – Model Reduction

```r
summary(lm(pemax~age+sex+height+weight+bmp+fev1+rv+frc+tlc))
```

|        | Estimate | Std. Error | t value | Pr(>|t|) |
|--------|----------|------------|---------|----------|
| (Intercept) | 176.0582 | 225.8912 | 0.779   | 0.448    |
| age    | -2.5420  | 4.8017     | -0.529  | 0.604    |
| sex    | -3.7368  | 15.4598    | -0.242  | 0.812    |
| height | -0.4463  | 0.9034     | -0.494  | 0.628    |
| weight | 2.9928   | 2.0080     | 1.490   | 0.157    |
| bmp    | -1.7449  | 1.1552     | -1.510  | 0.152    |
| fev1   | 1.0807   | 1.0809     | 1.000   | 0.333    |
| rv     | 0.1970   | 0.1962     | 1.004   | 0.331    |
| frc    | -0.3084  | 0.4924     | -0.626  | 0.540    |
| tlc    | 0.1886   | 0.4997     | 0.377   | 0.711    |
Multiple Regression in R – Model Reduction

> summary(lm(pemax~age+sex+height+weight+bmp))

...  

|          | Estimate | Std. Error | t value | Pr(>|t|) |
|----------|----------|------------|---------|----------|
| (Intercept) | 280.4482 | 124.9556   | 2.244   | 0.0369 * |
| age       | -3.0750  | 3.6352     | -0.846  | 0.4081   |
| sex       | -11.5281 | 10.3720    | -1.111  | 0.2802   |
| height    | -0.6853  | 0.7962     | -0.861  | 0.4001   |
| weight    | 3.5546   | 1.5281     | 2.326   | 0.0312 * |
| bmp       | -1.9613  | 0.9263     | -2.117  | 0.0476 * |

...  

> summary(lm(pemax~weight+bmp))

...  

|          | Estimate | Std. Error | t value | Pr(>|t|) |
|----------|----------|------------|---------|----------|
| (Intercept) | 124.8297 | 37.4786    | 3.331   | 0.003033 ** |
| weight    | 1.6403   | 0.3900     | 4.206   | 0.000365 *** |
| bmp       | -1.0054  | 0.5814     | -1.729  | 0.097797 . |

...  

> summary(lm(pemax~weight))

...  

|          | Estimate | Std. Error | t value | Pr(>|t|) |
|----------|----------|------------|---------|----------|
| (Intercept) | 63.5456  | 12.7016    | 5.003   | 4.63e-05 *** |
| weight    | 1.1867   | 0.3009     | 3.944   | 0.000646 *** |

...
Multiple Regression in R – Model Reduction-Correlation: age, weight, and height

```r
> summary(lm(pemax~age+weight+height))
... Estimate Std. Error t value Pr(>|t|)
(Intercept) 64.65555  82.40935 0.785 0.441
age 1.56755    3.14636  0.499 0.623
weight 0.86949   0.85922  1.012 0.323
height -0.07608   0.80278 -0.095 0.925
... 
> summary(lm(pemax~age+height))
... Estimate Std. Error t value Pr(>|t|)
(Intercept) 17.8600  68.2493 0.262 0.796
age 2.7178    2.93250  0.927 0.364
height 0.3397    0.69000 0.492 0.627

> summary(lm(pemax~age))
... Estimate Std. Error t value Pr(>|t|)
(Intercept) 50.408  16.657 3.026 0.00601
age 4.055   1.088  3.726 0.00111*
... 
> summary(lm(pemax~height))
... Estimate Std. Error t value Pr(>|t|)
(Intercept) -33.2757 40.0445  -0.831  0.4145
height 0.9319   0.2596  3.590 0.0015
... 
```

Multiple Regression in Excel

Please Watch Video at Home:

HTTPS://WWW.YOUTUBE.COM/WATCH?V=HGFHEFWK7VQ
Logistics Regression

Logistic regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome.

The outcome is measured with a dichotomous variable (in which there are only two possible outcomes).

Often taken to apply to a binary dependent variable.
Logistics Regression

The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features).

It allows one to say that the presence of a risk factor increases the odds of a given outcome by a specific factor.

The model itself simply models probability of output in terms of input.

Logistic Regression

Logistic regression is used in various fields:

- Trauma and Injury Severity Score, which is widely used to predict mortality in injured patients. Many other medical scales used to assess severity of a patient.

- Predicting the risk of developing a given disease based on observed characteristics of the patient (age, sex, body mass index, results of various blood tests, etc.).

- Predicting the probability of failure of a given process, system or product, or repairing a ship on-time.

- Predicting a customer's propensity to purchase a product or halt a subscription.

- Predicting the likelihood of a person's choosing to be in the labor force, and a business application would be to predict the likelihood of a homeowner defaulting on a mortgage.
Logistics Regression

- Logistic regression can be binomial, ordinal or multinomial.

- Binomial or binary logistic regression deals with situations in which the observed outcome for a dependent variable can have only two possible types, "0" and "1" (which may represent, for example, "dead" vs. "alive" or "win" vs. "loss").

- Multinomial logistic regression deals with situations where the outcome can have three or more possible types (e.g., "disease A" vs. "disease B" vs. "disease C") that are not ordered.

- Ordinal logistic regression deals with dependent variables that are ordered.

Logistics Regression in R

```r
> data.frame(smoking,obesity,snoring,n.tot,n.hyp)

<table>
<thead>
<tr>
<th>smoking</th>
<th>obesity</th>
<th>snoring</th>
<th>n.tot</th>
<th>n.hyp</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>No</td>
<td>No</td>
<td>60</td>
<td>5</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>17</td>
<td>2</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>187</td>
<td>35</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>85</td>
<td>13</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>51</td>
<td>15</td>
</tr>
<tr>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>23</td>
<td>8</td>
</tr>
</tbody>
</table>

> hyp.tbl <- cbind(n.hyp,n.tot-n.hyp)
> hyp.tbl

<table>
<thead>
<tr>
<th>n.hyp</th>
<th>n.tot-n.hyp</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>55</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>35</td>
<td>152</td>
</tr>
<tr>
<td>13</td>
<td>72</td>
</tr>
<tr>
<td>15</td>
<td>36</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
</tr>
</tbody>
</table>

> glm(hyp.tbl~smoking+obesity+snoring,family=binomial("logit"))

or simply:

> glm(hyp.tbl~smoking+obesity+snoring,binomial)
Logistic Regression in R

The other way to specify a logistic regression model is to give the proportion of diseased in each cell:

```r
> prop.hyp <- n.hyp/n.tot
> glm.hyp <- glm(prop.hyp~smoking+obesity+snoring, +
  binomial,weights=n.tot)

Call: glm(formula = hyp.tbl ~ smoking + obesity + snoring, ...
Coefficients:

(Intercept) smokingYes obesityYes snoringYes
-2.37766 -0.06777  0.69531  0.87194
```

```r
> glm.hyp <- glm(hyp.tbl~smoking+obesity+snoring,binomial)

> summary(glm.hyp)

Coefficients:

| Estimate  | Std. Error | z value | P    | r(>|z|)   |
|-----------|------------|---------|------|-----------|
| (Intercept) | -2.37766   | 0.38018 | -6.254 | 4e-10 *** |
| smokingYes | -0.06777   | 0.27812 | -0.244 | 0.8075    |
| obesityYes |  0.69531   | 0.28509 |  2.439 | 0.0147 *  |
| snoringYes |  0.87194   | 0.39757 |  2.193 | 0.0283 *  |

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 1
This is the table of primary interest.
Here, we get estimates of the regression coefficients, standard errors of same, and tests for whether each regression coefficient can be assumed to be zero.

Note: The layout is nearly identical to the corresponding part of the lm output.

Logistics Regression in Excel
Please Watch Video at Home:

HTTPS://WWW.YOUTUBE.COM/WATCH?V=EKRJDURXAU0
Clustering & Classification

Clustering

Dividing the population into similar groups called clusters based on common attributes.

- **Connectivity models**: Data points closer in data space exhibit more similarity to each other. Two approaches:
  1. Start with classifying all data points into separate clusters & then aggregating them as the distance decreases.
  2. All data points are classified as a single cluster and then partitioned as the distance increases.

- **Centroid models**: Iterative clustering algorithms where similarity is derived by the closeness of a data point to the centroid of the clusters (e.g., *K-Means* clustering algorithm).

- **Distribution models**: Probability that all data points in the cluster belong to the same distribution (For example: Normal, Gaussian).

- **Density Models**: Varied density of data points in the data space isolates various different density regions and assign the data points within these regions in the same cluster.
Clustering

Cluster analysis or clustering is the task of grouping similar objects into a cluster.

Grouping unlabeled examples is called clustering.

Clustering is used for exploratory data analysis, and statistical data analysis.

Applications:
- Anomaly detection
- Pattern recognition
- Image analysis
- Information retrieval
- Bioinformatics
- Data compression
- Computer graphics
- Machine learning

Clustering can be formulated as a multi-objective optimization problem.

Supervised & Unsupervised Learning: Classification Vs Clustering

How many different classes of things do we have here? What are they? What characteristics do they have that identify them as belonging to one class or another?
Supervised and Unsupervised Learning

Let’s think about characteristics that define them... Can you think of others?

Supervised Learning (Classification)

Supervised Learning: I know the classes and I know the labels for at least some of the data.

<table>
<thead>
<tr>
<th>ID</th>
<th>Number of Eyes</th>
<th>Number of Legs</th>
<th>Color</th>
<th>Is Fuzzy?</th>
<th>Has Stripes?</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Blue</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>Gray</td>
<td>Y</td>
<td>N</td>
<td>Kitten</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>White</td>
<td>Y</td>
<td>N</td>
<td>Kitten</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>Orange</td>
<td>Y</td>
<td>Y</td>
<td>Kitten</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>4</td>
<td>Green</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>Green</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>Orange</td>
<td>N</td>
<td>Y</td>
<td>Kitten</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2</td>
<td>Blue</td>
<td>Y</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>Purple</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>4</td>
<td>Brown</td>
<td>Y</td>
<td>Y</td>
<td>Kitten</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>4</td>
<td>White</td>
<td>N</td>
<td>Y</td>
<td>Kitten</td>
</tr>
</tbody>
</table>

How do I get the labels?

Training Data!
Supervised Learning (Classification)

Supervised Learning: I know the classes and I know the labels for at least some of the data.

<table>
<thead>
<tr>
<th>ID</th>
<th>Number of Eyes</th>
<th>Number of Legs</th>
<th>Color</th>
<th>Is Fuzzy?</th>
<th>Has Stripes?</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Blue</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>Gray</td>
<td>Y</td>
<td>N</td>
<td>Kitten</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>White</td>
<td>Y</td>
<td>N</td>
<td>Kitten</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>Orange</td>
<td>Y</td>
<td>Y</td>
<td>Kitten</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>4</td>
<td>Green</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>Green</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>Orange</td>
<td>N</td>
<td>Y</td>
<td>Monster</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2</td>
<td>Blue</td>
<td>Y</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>Purple</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>4</td>
<td>Brown</td>
<td>Y</td>
<td>Y</td>
<td>Kitten</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>4</td>
<td>White</td>
<td>Y</td>
<td>N</td>
<td>Kitten</td>
</tr>
</tbody>
</table>

Classification Algorithms

Classifier (Model)

If (Is Fuzzy = "Y") THEN Kitten = "Y"

Examples?

Test Data!

Let’s test it on part of the data we held out of the training set. We will call this part of the data - test data. ID 8? Fuzzy = Y... Oh no! ID 9? Fuzzy = N... Ok. Maybe this isn’t our best model?
Supervised Learning (Classification)

Supervised Learning: I know the classes and I know the labels for at least some of the data.

<table>
<thead>
<tr>
<th>ID</th>
<th>Number of Eyes</th>
<th>Number of Legs</th>
<th>Color</th>
<th>Is Fuzzy?</th>
<th>Has Stripes?</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Blue</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>Gray</td>
<td>Y</td>
<td>N</td>
<td>Kitten</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>White</td>
<td>Y</td>
<td>N</td>
<td>Kitten</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>Orange</td>
<td>Y</td>
<td>Y</td>
<td>Kitten</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>4</td>
<td>Green</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>Green</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>Orange</td>
<td>N</td>
<td>Y</td>
<td>Monster</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2</td>
<td>Blue</td>
<td>Y</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>Purple</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>4</td>
<td>Brown</td>
<td>Y</td>
<td>Y</td>
<td>Kitten</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>4</td>
<td>White</td>
<td>Y</td>
<td>N</td>
<td>Kitten</td>
</tr>
</tbody>
</table>

Classification Algorithms

Classifier (Model)

If (Is Fuzzy = “Y”) AND (Number of Eyes = 2) THEN Kitten = “Y”

Test Data!

Let’s try a new model...
ID 8? Fuzzy = Y, Number of Eyes = 1... OK
ID 9? Fuzzy = N, Number of Eyes = 1... OK
ID 10? Fuzzy = Y, Number of eyes = 2... OK
ID 11? Fuzzy = Y, Number of eye = 2... OK

Supervised Learning (Classification)

If we have trained our model and tested our model. At this point, we can use our trained and tested model to classify unlabeled data (if we are happy with the performance of our model in testing).

Unlabeled Data to be Classified

Kitten?

If it will always work perfectly?

Are we comfortable with the model we have? Why or Why Not? Could we do better? How?
Please watch the Following short video at Home
Unsupervised Learning (Clustering)

Unsupervised Learning: I do not have labels. In other words, I don’t know what classes exist or which records belong to which classes.

<table>
<thead>
<tr>
<th>ID</th>
<th>Number of Eyes</th>
<th>Number of Legs</th>
<th>Color</th>
<th>Is Fuzzy?</th>
<th>Has Stripes?</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>Blue</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>Gray</td>
<td>Y</td>
<td>N</td>
<td>Kitten</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>White</td>
<td>Y</td>
<td>N</td>
<td>Kitten</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>Orange</td>
<td>Y</td>
<td>Y</td>
<td>Monster</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>4</td>
<td>Green</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>Green</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2</td>
<td>Orange</td>
<td>Y</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2</td>
<td>Blue</td>
<td>Y</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2</td>
<td>Purple</td>
<td>N</td>
<td>N</td>
<td>Monster</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>4</td>
<td>Brown</td>
<td>Y</td>
<td>Y</td>
<td>Kitten</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>4</td>
<td>White</td>
<td>Y</td>
<td>N</td>
<td>Kitten</td>
</tr>
</tbody>
</table>

I have no idea what these records in the table correspond to (what their class or label is)... so now what do I do?

I can cluster based on the similarities of the records to see if I can find “clusters” in the data.

Please watch the Following short video at Home
Unsupervised Learning (Clustering)

What happens if I cluster the data based on number of eyes and legs (to stay in 2-dimensional space for now)? The monsters always seem to cause us problems, why? What is the problem with our other two monsters? What choices do we have to handle this?
What happens if I cluster the data based on number of eyes and fuzziness?

What changed? What is the difference between this and our classification method from earlier? What happens if I add Cookie Monster to the data I’m clustering?

Cluster: A collection of data objects
  ◦ similar (or related) to one another within the same group
  ◦ dissimilar (or unrelated) to the objects in other groups

Cluster analysis (or clustering, data segmentation, ...)
  ◦ Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters

Unsupervised learning: no predefined classes (i.e., learning by observations vs. learning by examples: supervised)

In our previous examples, we were using pictures. These help illustrate the concept, but give a false sense of class identification that we do not have when we cluster (we do not know that there are monsters and kittens). We just know the data points and often those data points do not separate as cleanly as we would like.
Clustering (Unsupervised Learning)

A good clustering method will produce high quality clusters

- high intra-class similarity: cohesive within clusters
- low inter-class similarity: distinctive between clusters

The quality of a clustering method depends on

- the similarity measure used by the method
- its implementation, and
- ability to discover some or all hidden patterns

For more details on Algorithms, please watch the video below:

https://www.youtube.com/watch?v=esmzYhuFnds
Machine Learning

Steps in Learning
Abstraction - training

Observations → Data → Model

<table>
<thead>
<tr>
<th>Distance</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.9m</td>
<td>1s</td>
</tr>
<tr>
<td>19.6m</td>
<td>2s</td>
</tr>
<tr>
<td>44.1m</td>
<td>3s</td>
</tr>
<tr>
<td>78.5m</td>
<td>4s</td>
</tr>
</tbody>
</table>

\[ g = 9.8 \text{m/s}^2 \]

Heuristics – Training Machines
Given a data set, a supervised learning algorithm optimizes a model that finds the combination of feature values that result in the target output.

**Examples:**
- An email is spam
- News is fake
- An application will default on loan
- A company will go bankrupt

---

**Machine Learning - Clustering**

![Clustering Diagram](chart)

- How crunchy the food is
- How sweet the food tastes

- Lettuce, celery, carrot, cucumber, green bean, apple, pear
- Nuts, grape, orange, banana, cheese, fish, shrimp, bacon

---
Classification Using Naïve Bayes

Probability of an event changes based on knowledge of related event.

\[ P(\text{Junk Email}/\text{Email is about Viagra}) > P(\text{Junk Email}) \]

Examples:
- Junk email (spam) filtering
- Intrusion detection in computer networks
- Medical diagnosis given a set of observed symptoms
- Auditing and Fraud detection

Clustering: Nearest-K Algorithm
Tomato a fruit? – Similarity by distance

Nearest K is Lazy
Example: Spam Emails

Spam - Joint Probability
Combination of Events

Step 3 — Unlike k-NN algorithm, Naive Bayes learner has separate stages.

**Naive Bayes classification syntax**

Using the `naiveBayes()` function in the `e1071` package

**Building the classifier:**

```r
m <- naiveBayes(train, class, laplace = 0)
```

- `train` is a data frame or matrix containing training data
- `class` is a factor vector with the class for each row in the training data
- `laplace` is a number to control the Laplace estimator (by default, 0)

The function will return a naive Bayes model object that can be used to make predictions.

**Making predictions:**

```r
p <- predict(m, test, type = "class")
```

- `m` is a model trained by the `naiveBayes()` function
- `test` is a data frame or matrix containing test data with the same features as the training data used to build the classifier
- `type` is either "class" or "raw" and specifies whether the predictions should be the most likely class value or the raw predicted probabilities

The function will return a vector of predicted class values or raw predicted probabilities depending upon the value of the `type` parameter.

**Example:**

```r
sms_classifier <- naiveBayes(sms_train, sms_type)
sms_predictions <- predict(sms_classifier, sms_test)
```
Generative A.I.  
(Artificial Intelligence)

Chatgpt A.I.

Definition
- Generative AI refers to deep-learning models that can generate high-quality text, images, and other content based on the data used for training.
  - Open A.I – Chatgpt: Tables, Worksheets, Flowcharts
  - Bard – Google
  - Microsoft – Bing Chat, 365 Copilot

Sample Applications
- OpenAI’s chatbot, powered by its latest large language model, can write poems, tell jokes, and churn out essays that look like a human created them.
- Generative models can also learn the grammar of software code, molecules, natural images, and a variety of other data types.

Video:
https://research.ibm.com/blog/what-is-generative-AI?utm_content=SRCWW&p1=Search&p3=4370007631416319555&5=e&gclid=EAIaIQobChMfZx46fNgAMIVuSFBSR1hLAz1EAAYASABBYo8FaXvID_BwE&gclsrc=aw.ds
Generative AI’s evolution

- 1952: Alan Turing writes a paper on a computer that could play chess, laying the groundwork for modern AI.
- 1962: Marvin Minsky publishes “Steps Toward Artificial Intelligence,” discussing the potential of AI.
- 2002: The first commercially available AI-powered systems are introduced.
- 2006: IBM’s Watson wins Jeopardy!
- 2013: Google launches its first AI system, Google Brain.
- 2016: Amazon launches Alexa, its first AI-powered virtual assistant.
- 2021: OpenAI introduces DALL-E, a system that can generate images from text descriptions.
- 2022: Generative AI is used to create music and art.

Generative AI use cases

- Visual Content
  - Image Enhancement
  - Video Prediction
  - 3D shape Generation
- Audio Generation
  - Music Composing
  - TTS Generator
  - STS Conversion
- Text Generation
  - Creative Writing
  - Chatbots
  - Translation
- Code Generation
  - Code Generation
  - Code compilation
  - Bug Fixing
Generative A.I. Tools

- Visual
  - Image Generators
  - Video Generators
  - Design Generators
- Audio
  - Voice Generators
  - Music Generators
- Text Generators
- Code Generators

Generative Adversarial Network Architecture

- Random Input Vector (Random noise)
- Generator Model
  - Generated fake example
- Real examples
- Discriminator Model
  - Binary classification
  - Update Model
  - Real
  - Fake

Update Model
Generative A.I. Mechanism

Large language models (LLMs) are models with billions or trillions of parameters that have enabled generative AI models to write engaging text, generate photorealistic images.

Future:

- Write code
- Design new drugs
- Develop products
- Chat assistants
- Work paper files
- Email replies - helpful tone
- Redesign business processes
- Transform supply chains.

- Text Generation
- Text to Image
- Speech to text
- Image to text
Oversight risks of Generative A.I.

- **Lack of transparency** -
  - models are unpredictable, and little understanding of how they work.
- **Accuracy** -
  - inaccurate and fabricated answers.
- **Bias** -
  - policies or controls required to detect biased outputs.
- **Intellectual property (IP) and copyright** -
  - no verifiable data governance and protection assurances regarding confidential enterprise information.
- **Cybersecurity and fraud** -
  - malicious actors’ use of generative AI systems for cyber and fraud attacks (e.g., deep fakes).
- **Sustainability** -
  - significant use of electricity.

---

Generative A.I. Risks

- **Bias**
- **Spread of Misinformation**
- **Job Displacement / Loss**
- **Privacy**
- **Who controls Gen AI?**
- **Overreliance on Gen AI Apps**
- **Unintended Consequences**
- **Environmental Impact**
Oversight risks of Generative A.I.

Example LLM Capabilities and Near-term Use Cases in CSG

<table>
<thead>
<tr>
<th>Example LLM Capabilities</th>
<th>Near-term Use Cases in CSG (In Order of Complexity)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content Creation &amp; Augmentation</td>
<td>Automated ACW</td>
</tr>
<tr>
<td>Q &amp; A and Discovery</td>
<td>Knowledge Asset Creation</td>
</tr>
<tr>
<td>Tone of Content</td>
<td>Virtual Assistants</td>
</tr>
<tr>
<td>Summarization</td>
<td></td>
</tr>
<tr>
<td>Simplification</td>
<td></td>
</tr>
<tr>
<td>Classification of Content</td>
<td></td>
</tr>
</tbody>
</table>

LLM - Large Language Model
CSG – Customer Service and Support
ACW – After-Call Work

89

Generative A.I. Accounting/Business Apps

plooto

Improved insights and decision-making
Increased efficiency and cost savings
Targeted communications
Automation of repetitive tasks
Fraud detection
Improved compliance
Improved customer experience

Document analysis
Financial report generation
Financial analysis and forecasting

10 WAYS
Generative AI is remaking finance & accounting processes

90
Generative A.I. Design Examples

Generative A.I. and Analytics

Data Analytics can leverage generative AI by:

• Allowing machines to create and test hypotheses based on all available data sources

• Generating business insights and updating them dynamically over time and as data changes

• Bridging the gaps in the data itself and through the use of clustering and other methods

• Filling missing data
Generative A.I. & Data Analytics Value Chain

Getting the data
- Classification, tagging, data cleaning, etc.

Analyzing the data
- Broader business context, automation, create synthetic data to build supervised learning data sets for training machine learning models

Generating insights
- Mimic human inferencing processes to contextualize analytics results

Delivering Insights and Driving Decisions
- Near-real-time insights without the need for human intervention to add context

Generative A.I. & Data Analytics Value Chain

Data Augmentation:
- Enhance training of predictive models by creating synthetic data that mimics the distribution of real-world data
- Improve model generalization and performance, especially when real data is scarce.

Anomaly Detection:
- Model normal data distribution
- Identify anomalies by detecting data points that deviate from learned pattern.

Imputation:
- Deal with missing data, filling gaps by generating plausible values based on existing patterns, improving the accuracy of analysis.

Visualization and Exploration:
- Generate visual content
- Facilitate exploration of complex datasets
- Aid analysts in identifying trends and patterns.
Generative A.I. and Deep Learning Videos


Introduction to Blockchain

A Consensus protocol assures every new block added to the Blockchain is the only version of truth agreed by the majority of the nodes (51%) on the blockchain network.

Distributed Trust

Fundamental notion of Blockchain

Centralized Trust
A Central node assures transactions between parties.

Distributed Trust
A Consensus protocol assures every new block added to the Blockchain is the only version of truth agreed by the majority of the nodes (51%) on the blockchain network.
Key Technology of Blockchain

- Distributed Accounting
- Consensus Mechanisms
- Encryption Algorithms
- Smart Contracts

Blockchain & Distributed Trust

Definition
- Distributed database, shared among nodes of a blockchain network, and new data added after validated and approved by network participants and cryptographically linked.
- Distributed trust model of computing removes the need for a centralized trust mechanism.
- Consensus protocol: 51% of nodes in network agree on same version of truth before a new block is added to the chain of existing blocks.

Potential effects of blockchain:
- Reduce agency costs and information asymmetry
- Increase transparency and audibility
- Increase trust and reliability
- Reduce human error, cost, and fraud
- Improve data quality
- Solve privacy paradox
Decentralized Autonomous Organization (DAO)

- A software-based organization with decentralized governance.
- Contracts are building blocks of the firms in the economy.
- Blockchain and Smart contracts will help the proliferation of DAOs.
- Human intervention deals with unexpected situations in DAO.

Blockchain considerations

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transparency</td>
<td>Limited understanding</td>
</tr>
<tr>
<td>Integrity</td>
<td>Lack of technology maturity</td>
</tr>
<tr>
<td>Immediacy</td>
<td>Lack of regulation</td>
</tr>
<tr>
<td>Tractability</td>
<td>Lack of data quality</td>
</tr>
<tr>
<td>Innovation</td>
<td>Lack of Central Control</td>
</tr>
<tr>
<td>Scalability</td>
<td>Lack of Digital and Physical Mapping</td>
</tr>
<tr>
<td>Trust</td>
<td>Implementation</td>
</tr>
<tr>
<td>Remove intermediaries- Cost</td>
<td>Complexity</td>
</tr>
</tbody>
</table>
Blockchain Distributed Ledger

**Key characteristics of the distributed ledger:**

- Transactions recorded among network participants.
- Network participants control and develop consensus.
- Eliminate the need for central authority.
- Record timestamped tamper-proof transactions.
- Single, verifiable, & recognized source of truth.
- Smart contract code is stored on a blockchain that is triggered by transactions, potentially reading or writing from the blockchain or other systems.

Smart Contracts Benefits

**Smart Contracts** are computer code that are self-executing based on event or time triggers such that transactions happen automatically once the terms of the agreement are satisfied. The benefits of Smart contracts are:

- **Auditability:**
  Transparency and visibility of transactions data.
- **Autonomy:**
  Remove bias and contradictory interests of parties.
- **Precision:**
  Assure any diversion from specific code they execute.
- **Redundancy:**
  Distributed ledger duplicates documents over blockchain.
- **Speed:**
  Smart contracts save time and effort
Blockchain will not eliminate accountants & CPAs but will impact Profession

- Less effort on sampling, testing, and validating transactions
- Focus on internal controls, complex transactions, and investigating anomalies & irregularities
- New opportunities for assurance services in cybersecurity and sustainability
- New opportunities for third-party assurance services

Implications of Blockchain on 5 COSO Components

**Control Environment**
Blockchain records transactions with minimal human intervention and manages integrity and ethics. Intertwining of an entity with other entities or persons participating in a blockchain and how to manage the control environment as a result is a challenge.

**Risk Assessment**
Blockchain creates new risks and mitigate extant risks, promotes accountability, maintains record integrity, and provides an irrefutable record where authorizing/sending a message or record is traceable and cannot be denies.

**Control Activities**
Blockchain facilitates control activities. Blockchain and smart contracts effectively and efficiently support global business, minimizing human error and fraud. The collaborative aspects of blockchain introduce additional complexity due to decentralization and lack of single party accountability.

**Information & Communication**
Enhanced visibility of transactions and availability of data facilitate management communication of financial information to key stakeholders, supporting better auditability of information transacted on a blockchain.

**Monitoring Activities**
Blockchain facilitates monitoring timeliness and detail, impacting auditing practice. Smart contracts and Internet of Things (IoT) alter future of monitoring.
COBIT 2019 Framework

Standards provide guidance about how to conduct audits of technology and may help with blockchain.

Segregation of Duties, Fraud Triangle, and Distributed Trust

Segregation of incompatible duties:

- **Authorizing**
  - Transactions

- **Recording**
  - Transactions

- **Maintaining**
  - Custody of Assets
Blockchain and Accounting

A paradigm shift in the accounting profession?

Help accountants get clarity over transaction history of their organizations without investing time and resources on maintaining and reconciling ledgers

Blockchain value to Accounting Functions:

Immutability
- no one can modify/alter records, once verified and approved

Security
- Personal data and information

Trust
- Consensus-based Distributed Trust

Speedy transactions
- Digital content distribution- information available to stakeholders without replication

Mitigation of risk and frauds
- Smart contracts, digital rights management controlling access

New Reality and Opportunities

Digitizing transactional accounting processes
- Reduction in data entry personnel and optimizing organization’s need

• Automated fiscal period-end accounting closing
  - Financial Reporting and Tax Policy

• Auditing
  - Systems using AI can scan 100 percent of the transactions, instead of using a sample

• Business process outsourcing (BPO)
  - Costs of accounting processes can be reduced by outsourcing due to economies of scale

• Regulatory filings
  - Transmit financial statements to government agencies.
Embedding Controls in Blockchain and Smart Contracts

Segregation of duties
– Designate conflicting accounting functions to different individuals to minimize fraud

• Access controls
  – Restrict system access and use appropriately

• Required approvals
  – Require approvals of senior employees or managers based on algorithms

• Asset audits
  – Assure that system information is consistent with physical assets

• Templates
  – Standardize financial documents for comparison of transactions

• Trial balances
  – Use trial balance with double-entry accounting to reveal errors or fraud

• Reconciliations
  – Reconcile external information with information in system

Blockchain: Human Context- Remove Bias?

Human BIAS Examples:

• Anchoring
• Conservatism
• Recency
• Availability
• Information Salience
• Bandwagon/Groupthink
• Selective perception
• Blind spot
• Salience (Your focus)
• Survivorship
• Clustering
• Placebo
• Confirmation
• Pro-innovation
• Overconfidence
• Stereotyping
• Outcome
Automated Continuous Auditing – Smart Contracts

Automated Auditing
• Distributed Ledger & Smart Contracts enable Continuous auditing by automating feedback process, monitoring, and risk analysis more frequent.
• Blockchain technology & Smart Contracts make some accounting functions real-time.

Benefits:
• More timely fixes and reduced errors
• Lower audit expenses
• Better analytics - population testing instead of sampling limitations
• More effectiveness of internal audit
• Better identification of fraud via trend analysis
• Better management of risk impact
• Independence

Blockchain and Smart Contracts Potential Impacts

The Future
• Algorithm audit and Bias Detection
• Algorithmic Coding of Accounting Entries
• Analysis of Credit Card Statements and coding
• Bank reconciliations
• Distributed Input Payables
• Distributed Payroll Administration
• Distributed Transparency to Improve fraud detection
• Distributed Self-assessment of internal control systems
• Automated Analysis (e.g., financial results, tax return data)
“Educating the mind without educating the heart is no education at all.”
– Aristotle

“The potential of greater good goes right along with the potential for greater evil.”
– Larry Wall

THANK YOU!

QUESTIONS?