How to build and deploy machine learning projects

Litan Ilany, Advanced Analytics

litan.ilany@intel.com
AGENDA

• Introduction
• Machine Learning: Exploration vs Solution
• CRISP-DM
• Data Flow considerations
• Other key considerations
• Q&A
INTRODUCTION - LITAN ILANY

litan.ilany@intel.com

- Data Scientist Leader at Intel’s Advanced Analytics team.
- Owns a M.Sc. degree in Information-Systems Engineering at BGU (focused on Machine-Learning and Reinforcement-Learning)
- Married + 2, Live in Kiryat Motzkin
MACHINE LEARNING

• Statistics
• Pattern recognition
• Generalization / Inductive Inference

• Types of learning:
  • Supervised vs Unsupervised Learning
  • Passive vs Active & Reinforcement Learning
  • Batch vs Online Learning
ML – ALGORITHM VS SOLUTION

• “Given a data matrix...” – does not exist in real life

• Pareto Principle (80/20 rule)
  • Technical aspects
  • Business needs
  • Extreme cases
<table>
<thead>
<tr>
<th>ML PROJECT - GO / NO-GO DECISION</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BUSINESS FEASIBILITY</strong></td>
</tr>
<tr>
<td><strong>DATA FEASIBILITY</strong></td>
</tr>
<tr>
<td><strong>EXECUTION FEASIBILITY</strong></td>
</tr>
</tbody>
</table>
CRISP-DM

Cross-Industry Standard Process for Data Mining

- A structured methodology for DM projects
- Based on practical, real-world experience
- Conceived in 1996-7
CRISP-DM

- **Business Understanding**: what is the problem we are dealing with?
- **Data Understanding**: what is the data we are working with?
- **Data Preparation**: what are the transformations and extractions to be done on the Data?
- **Modeling**: what is the data model we should use?
- **Evaluation**: does the model meet the project goals?
- **Deployment**: how should we use the model we developed?
CRISP-DM: BUSINESS UNDERSTANDING

- Determine business objective
- Assess situation
- Determine data mining goals and success criteria
- Determine project plan
Example: Smart CI

- Each git-push is integrated with the main repository – after tests series passes
- Multi git-push (can't check one-by-one)
- Bug in code causes entire integration to fail
Example: Smart CI

- Each git-push is integrated with the main repository – after tests series passes
- Multi git-push (can't check one-by-one)
- Bug in code causes entire integration to fail
CRISP-DM: BUSINESS UNDERSTANDING - EXAMPLE

• Goals and success criteria:
  • Reduce Turnaround Time (TAT)
  • At least 20% time reduction

• Project plan

Model's I/O flow is reasonable
Reminder 😊
CRISP-DM: DATA UNDERSTANDING

- Collect initial data
- Describe data
- Explore data
- Verify data quality

Example:

- Git-log files (unstructured data):
  - Commits – numerical / binary
  - Files, Folders – numerical / binary
  - Lines – numerical
- Git DB (structured data):
  - Users – categorical
  - Timestamps, etc.
- Historical tests results (labels)
• Integrate data from multi sources
• Format data
• Feature extraction
• Clean data
• Construct data
  • Derive attributes – transformation
  • Reduce imbalance data
  • Fill in missing values
• Feature selection

Example:
• Generate features from log
• Generate and clean user-features
• Normalize counters
• Thousands of features – remove unnecessary ones
• Data balancing (if needed)
CRISP-DM: MODELING

- Select modeling technique
  - Consider computer resources, computation time, number of features, business needs
- Generate test design
  - Train/Test split, Cross validation
  - Simulation (chronological order)
- Build model
- Assess model

Example:
- We’ll check various ML models with various hyperparameters
- Simulation, weekly training phase
CRISP-DM: MODELING – EXAMPLE (SMART CI)

• Model assessment:
  • Which model to choose?
  • How can we measure it?

<table>
<thead>
<tr>
<th>Predicted \ Actual</th>
<th>push Passed</th>
<th>push failed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted pass</td>
<td>55</td>
<td>18</td>
<td>73</td>
</tr>
<tr>
<td>Predicted fail</td>
<td>15</td>
<td>12</td>
<td>27</td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>30</td>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Predicted \ Actual</th>
<th>push Passed</th>
<th>push failed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted pass</td>
<td>35</td>
<td>5</td>
<td>40</td>
</tr>
<tr>
<td>Predicted fail</td>
<td>35</td>
<td>25</td>
<td>60</td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>30</td>
<td>100</td>
</tr>
</tbody>
</table>
CRISP-DM: MODELING - EXAMPLE (SMART CI)

- Model assessment:
  - Which model to choose?
  - How can we measure it?

<table>
<thead>
<tr>
<th>Measure</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>(55+12)/100 = 67%</td>
<td>(35+25)/100 = 60%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model A</th>
<th>Predicted \ Actual</th>
<th>push Passed</th>
<th>push failed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted pass</td>
<td>55</td>
<td>18</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>Predicted fail</td>
<td>15</td>
<td>12</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>30</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model B</th>
<th>Predicted \ Actual</th>
<th>push Passed</th>
<th>push failed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted pass</td>
<td>35</td>
<td>5</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Predicted fail</td>
<td>35</td>
<td>25</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>30</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
CRISP-DM: MODELING – EXAMPLE (SMART CI)

- Model assessment:
  - Which model to choose?
  - How can we measure it?

<table>
<thead>
<tr>
<th>Measure</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>(55+12)/100 = 67%</td>
<td>(35+25)/100 = 60%</td>
</tr>
<tr>
<td>Precision</td>
<td>55/73 = 75%</td>
<td>35/40 = 87%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model A</th>
<th>Predicted \ Actual</th>
<th>push Passed</th>
<th>push failed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted pass</td>
<td>55</td>
<td>18</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>Predicted fail</td>
<td>15</td>
<td>12</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>30</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model B</th>
<th>Predicted \ Actual</th>
<th>push Passed</th>
<th>push failed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted pass</td>
<td>35</td>
<td>5</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>Predicted fail</td>
<td>35</td>
<td>25</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>30</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
**CRISP-DM: Modeling – Example (SMART CI)**

- Model assessment:
  - Which model to choose?
  - How can we measure it?

<table>
<thead>
<tr>
<th>Measure</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>(55+12)/100 = 67%</td>
<td>(35+25)/100 = 60%</td>
</tr>
<tr>
<td>Precision</td>
<td>55/73 = 75%</td>
<td>35/40 = 87%</td>
</tr>
<tr>
<td>Recall</td>
<td>55/70 = 76%</td>
<td>35/70 = 50%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Predicted \ Actual</th>
<th>Model A</th>
<th></th>
<th>Model B</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>push</td>
<td>push failed</td>
<td>push</td>
<td>push failed</td>
</tr>
<tr>
<td>Predicted pass</td>
<td>55</td>
<td>18</td>
<td>Predicted pass</td>
<td>35</td>
</tr>
<tr>
<td>Predicted fail</td>
<td>15</td>
<td>12</td>
<td>Predicted fail</td>
<td>35</td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>30</td>
<td>Total</td>
<td>70</td>
</tr>
</tbody>
</table>

Total: 100
CRISP-DM: MODELING – EXAMPLE (SMART CI)

- Model assessment:
  - Which model to choose?
  - How can we measure it?

<table>
<thead>
<tr>
<th>Measure</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>(55+12)/100 = 67%</td>
<td>(35+25)/100 = 60%</td>
</tr>
<tr>
<td>Precision</td>
<td>55/73 = 75%</td>
<td>35/40 = 87%</td>
</tr>
<tr>
<td>Recall</td>
<td>55/70 = 76%</td>
<td>35/70 = 50%</td>
</tr>
<tr>
<td>FPR*</td>
<td>18/30 = 60%</td>
<td>5/30 = 17%</td>
</tr>
</tbody>
</table>

*Lower is better

<table>
<thead>
<tr>
<th>Predicted \ Actual</th>
<th>push Passed</th>
<th>push failed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted pass</td>
<td>55</td>
<td>18</td>
<td>73</td>
</tr>
<tr>
<td>Predicted fail</td>
<td>15</td>
<td>12</td>
<td>27</td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>30</td>
<td>100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Predicted \ Actual</th>
<th>push Passed</th>
<th>push failed</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted pass</td>
<td>35</td>
<td>5</td>
<td>40</td>
</tr>
<tr>
<td>Predicted fail</td>
<td>35</td>
<td>25</td>
<td>60</td>
</tr>
<tr>
<td>Total</td>
<td>70</td>
<td>30</td>
<td>100</td>
</tr>
</tbody>
</table>
CRISP-DM: MODELING – EXAMPLE (SMART CI)

Business ease of explanation

- SVM
- Random Forest
- GBT
- KNN
- Decision Tree
- Regression

Complex to Simple
CRISP-DM: MODELING - EXAMPLE (SMART CI)

- SVM
- NN
- Random Forest
- GBT
- Decision Tree
- Regression
- KNN

Expected value (in simulations) vs. Business ease of explanation.

Expected value

Business ease of explanation
**CRISP-DM: EVALUATION**

- Evaluate results
  - In terms of business needs
- Review Process
- Determine next steps

**Example:**

<table>
<thead>
<tr>
<th>Predicted \ Actual</th>
<th>push Passed</th>
<th>push failed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predicted pass</td>
<td>TP</td>
<td>FP</td>
</tr>
<tr>
<td>Predicted fail</td>
<td>FN</td>
<td>TN</td>
</tr>
</tbody>
</table>

**TAT reduction:**

- TP = 50% reduction (X2 faster)
- FN = 0% reduction
- FP = -500-5000% reduction (X5-50 slower)
CRISP-DM: DEPLOYMENT

- Plan and deploy the model
- Plan monitoring and maintenance process

Example:
- Integrate with existing CI system
- Weekly automatic process that will train the model
- Weekly automatic process that will monitor the model’s performance and suggest better hyper parameters (if needed)
CRISP-DM: DATA FLOW

- BUSINESS UNDERSTANDING
  - Data Flow Architecture

- DATA
  - Data Flow Implementation
  - Data Schema Architecture

- EVALUATION
  - Data flow validation

- DEPLOYMENT

- DATA PREPARATION
  - Data Flow Architecture

- MODELING
OTHER KEY CONSIDERATIONS

• Use Git (or other version control platform)
• Automate the research process (trial-and-error)
• Use Docker containers
• TEST YOUR CODE (don’t think of it as black box)
• ML Technical Debt – code and data
REFERENCES

CRISP-DM (Wikipedia)

4 things DSs should learn from software engineers

Machine Learning: The High Interest Credit Card of Technical Debt