

Model Governance Policy

1. Model Development:

- 1.1. Definition of a model.
- 1.2. Definition of model risk.
- 1.3. Types of models.
- 1.4. Steps of Model Development and Best Practices:
 - 1.4.1. Model Objective:
 - 1.4.2. Requirements:
 - 1.4.3. Data Preparation:
 - 1.4.4. Exploratory Data Analysis:
 - 1.4.5. Variable Selection:
 - 1.4.6. Missing Value Treatment:
 - 1.4.7. Variable Derivation:
 - 1.4.8. Feature Engineering:
 - 1.4.9. Removal of Highly Associated Variables:
 - 1.4.10. Normalization of Numerical Variables:
 - 1.4.11. Outlier Removal:
 - 1.4.12. Sampling Data for Modeling:
 - 1.4.13. Train-Test Split:
 - 1.4.14. Selection of Baseline Models and Hyper-parameters Estimation:
 - 1.4.15. Variance Estimation of Baseline Models:
 - 1.4.16. Model Training:
 - 1.4.17. Output Evaluation:
 - 1.4.18. Documentation: [Credit Card Cross Sell: Customer Segmentation](#) **AND** [Modeling Documentation](#)
 - 1.4.19. Python file templates:

jupyter creditcardcrossell_custsegmentation_RF_LR_24thOct19_v3 Last Checkpoint: 01/16/2020 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 C

Run Markdown

Credit Card Crossell CustSegmentation_RF_LR_Stacked (ProductUse_Model Use_AlgosStacked/Used)

Model ID: 123456

Product: Credit Card

Model Use: The Model will segment the customers on the basis of 5 parameters: A, B, C, D, E. 4 segments were found.

Algorithms Used: Random Forest, Logistic Regression

Business Owner: Jane Doe

Model Developer: John Doe

Model User: Gina Doe

Model Validator: Jacob Doe

Model Validation Frequency: Every 6 months

Dependencies:

Upstream Models: CreditCardCrossell_MarketBasket_Reg_XGB, CreditCardCrossell_Propensity_LR_RF, CreditCardCrossell_RecommendationEngine_XGB_AdaB

Downstream Models: None

Creation Date: 24th October, 2019

In [5]: `##Pulling the data from the source`

In [6]: `##Preparing data for use`

In [7]: `##Creating model`

In [8]: `##Train-test split`

In [9]: `##Running Model`

In [10]: `##Model Evaluation`

In [11]: `##Tabulation of Results`

2. Model Implementation:

2.1. Template: [Model Implementation](#)

3. Model Use:

3.1. Defined use:

3.2. Assumptions and limitations:

3.3. Controls and approvals for stakeholders:

4. Model Validation:

4.1. Elements and scorecards:

4.1.1. Data Import:

4.1.2. Discrimination metrics: Python templates

- 4.1.3. PSI metrics:
- 4.1.4. Variable quality metrics:
- 4.1.5. Collated Results:
- 4.1.6. Compare development vs validation cohort:
- 4.1.7. Export metric datasets and graphs:
- 4.1.8. Visualisation:
<https://gargi.invisionapp.com/console/Risk-Monitoring-ck3sdy01i0d8f018b5shi9pda/ck3sdy76g07gh016hazghnfu2/play>

4.2. Benchmarks:

5. Model Inventory:

5.1. Template: [Model Inventory](#)

6. Model Governance:

- 6.1. Organisational Structure:
- 6.2. Roles and Responsibilities:
- 6.3. Policies and Procedures:
- 6.4. Escalation Matrix:
- 6.5. Reporting Templates:

7. Model Materiality and Risk Appetite Setting:

- 7.1. Materiality setting: [Model Materiality](#)
- 7.2. Prioritization and validation frequency:
- 7.3. Risk Appetite Setting:

FURTHER:

- 1. Model maturity framework.
- 2. Organisational structure, visualisation, some integrated framework.