Independent Model Validation on Enterprise Risk Management Model Documentation Requirements

Machine Learning Model Validation

March 15, 2019

State Farm Insurance – EDA – AI-MRM

Table of Contents

Introduction

We will use *the classic Titanic* dataset. The dataconsists of demographic and traveling information for 1,309 of the Titanic passengers, and the goal is to predict the survival of these passengers. It contains all the facts, history, and data surrounding the Titanic, including a full list of passengers and crew members.

Design

Business Problem

- How the model is used and how the model fits into overall strategy?
  Business problem solution aims to use machine learning techniques on the Titanic data to analyze the data for binary classification and to predict the survival of the Titanic passengers by using data-mining algorithms. The prediction and efficiency of these algorithms depend greatly on data analysis and the model.
- What is the current approach to solving the problem?
  To predict the survival of the passengers using the Titanic dataset binary classification models are considered as current approach
  - Why do we need to (re)build the model?
    <If the model is rebuild it needs to be explained as it's not required for current model>
    Binary Target
  - What alternatives were considered (both modeling and non-modeling)?
    We could considers many other models even Automatic Machine Learning model

Modeling Methodology

- Description of the conceptual approach and quantification techniques of the model, including model theory with references and model logic.
  Listing the methodology that fits to the specific dataset and then using the qualification techniques to identify the model.
  1. logistic regression,
  2. random forest
  3. Adaboosting
  4. ensemble voting classifier
  5. SVM linear kernel
- How does the model address the business problem?
  Any model address the business problem in providing the accuracy in the outcomes. Further understanding the business scenario and make it to work for certain business purpose.
- Provide rational for the methods chosen.
  Reason for choosing the model is to get the
- Documentation of the target variable including how the target relates to the business problem.

Sl.No.	Survived	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare
0	0	3	male	22	1	0	A/5 21171	7.25
1	1	1	female	38	1	0	PC 17599	71.2833
2	1	3	female	26	0	0	STON/O2. 3101282	7.925
3	1	1	female	35	1	0	113803	53.1
4	0	3	male	35	0	0	373450	8.05

- Supporting evidence from research/industry practice to substantiate the model theory and design.

- Modeling processes should include comparison to alternative theories and approaches.
- Discuss metrics that are used in assessing the model performance.
    - Accuracy
    - AUC ROC
    - Precision
    - Recall
    - F1 Score
    - Lift Chart
    - Fitting time
    - Score time
- Documentation of changes to the methodology that were made based on the results from the modeling.
  <If applicable>

Assumptions

This repo is able to automate part of the 'Model Development' and 'Outcome Analysis' documentation work for Binary Classification model by automatically build SR reviewed and recommended challenger models and outcome metrics comparison across models.

Limitations

•Any non-binary-classification model
•This repo does not address other parts of the model validation work yet(possibly in the future):e.g. Model Design', 'Data EDA and Preprocessing', 'Performance Monitoring' ,a complete sections of model validation components can be found in Model Validation Guidance developed by SR
•The automation feature DOES NOT intend to replace model developers' review and analysis on the pros and cons of different models, which is a critical piece of model validation work that requires data science professional's judgment and rational considering the business problem and application.

Downstream and upstream model dependencies

Data

Data Sources

Sl. No.	Name	*Value*
1	*Sources*	*<Hadoop>*
2	*Location*	*<path>*
3	*Owner*	*<Data Scientist>*
4	*Process used for pulling data*	*<Process>*
5	*Code used to pull the data*	*<Code>*

- The data is suitable to use, accurate, and complete.
    - Data time frames are provided.
    - The rational for the selection of the data is documented.
    - List of exclusions (Waterfall from the source data) are included.

Data Quality Assessment

- Items that we look for from the documentation:
  - EDA (including univariate assessment of variables and how they relate to the dependent variable)
  - Variable trends over time
  - Stability of the variables in the build relative to the use time period
  - Documentation of all the data anomalies.
  - List of all input variables with basic definitions.
  - Output from EDAs
    - Valid ranges
    - Missing values/special values
  - List of variables excluded and included and rational why.
  - Discussion of approach used to detect outliers and methods used to deal with missing data.
  - Basic documentation of the transformations, derived variables, and the code used.
    - Missing handling
    - Capping and flooring
  - Documentation of the data source and the relevance of the data source to the data used during implementation.
  - Rational for choice of the model build time period and choice of out of time validation set (or lack thereof).
  - Graphical views of the data.

Code Review

- Documentation of the code used to pull the data is provided.

```
************* Module main
AI-Risk-Management-Automation-master\main.py:7:0: C0301: Line too long (158/100) (line-too-long)
AI-Risk-Management-Automation-master\main.py:12:0: C0301: Line too long (113/100) (line-too-long)
AI-Risk-Management-Automation-master\main.py:35:0: C0301: Line too long (107/100) (line-too-long)
AI-Risk-Management-Automation-master\main.py:86:0: C0301: Line too long (112/100) (line-too-long)
AI-Risk-Management-Automation-master\main.py:1:0: C0111: Missing module docstring (missing-docstring)
AI-Risk-Management-Automation-master\main.py:18:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\main.py:26:-1: W0105: String statement has no effect (pointless-string-statement)
AI-Risk-Management-Automation-master\main.py:29:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\main.py:47:-1: W0105: String statement has no effect (pointless-string-statement)
AI-Risk-Management-Automation-master\main.py:50:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\main.py:61:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\main.py:68:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\main.py:68:15: W0622: Redefining built-in 'dir' (redefined-builtin)
AI-Risk-Management-Automation-master\main.py:73:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\main.py:73:0: R0914: Too many local variables (26/15) (too-many-locals)
AI-Risk-Management-Automation-master\main.py:77:32: W0612: Unused variable 'xtest' (unused-variable)
AI-Risk-Management-Automation-master\main.py:73:0: R0915: Too many statements (72/50) (too-many-statements)
************* Module metrics
AI-Risk-Management-Automation-master\metrics.py:28:0: C0301: Line too long (102/100) (line-too-long)
AI-Risk-Management-Automation-master\metrics.py:40:0: C0301: Line too long (101/100) (line-too-long)
AI-Risk-Management-Automation-master\metrics.py:1:0: C0111: Missing module docstring (missing-docstring)
AI-Risk-Management-Automation-master\metrics.py:23:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\metrics.py:52:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\metrics.py:75:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\metrics.py:94:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\metrics.py:104:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\metrics.py:105:4: W0612: Unused variable 'prediction' (unused-variable)
AI-Risk-Management-Automation-master\metrics.py:111:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\metrics.py:111:0: R0913: Too many arguments (7/5) (too-many-arguments)
AI-Risk-Management-Automation-master\metrics.py:111:0: R0914: Too many local variables (28/15) (too-many-locals)
AI-Risk-Management-Automation-master\metrics.py:145:0: C0111: Missing function docstring (missing-docstring)
AI-Risk-Management-Automation-master\metrics.py:145:0: R0913: Too many arguments (7/5) (too-many-arguments)
AI-Risk-Management-Automation-master\metrics.py:145:0: R0914: Too many local variables (28/15) (too-many-locals)
************* Module models
AI-Risk-Management-Automation-master\models.py:1:0: C0111: Missing module docstring (missing-docstring)


------------------------------------------------------------------
Your code has been rated at 8.85/10 (previous run: 8.85/10, +0.00)
```

- Code, files, logs, data sets. This can be documented using a link to GitLab or some other respository
- Choice of the software
- Good documentation of the code

Development

Model Selection Process

- Discussion of the statistical methods
  - Document the decisions made during model development.
  - Document any judgment calls made during the model build.
  - Give details on the methods used for variable reduction.
  - Provide rational for choice of the metrics used.
  - Document the choice of the final model.

Model Output (e.g. ANOVA Table for linear regression)

- Discussion of the variable transformations
  - Capping flooring
  - Creation of variables
  - Non-linear transformations
  - Discuss how method chosen helps alleviate need for transformations.
- Were changes made to the planned modeling approach or was the planned model used in light of lessons learned during development?
- Assess the model to ensure the model works as intended and challenged by the Model Owner and other Business Partners.

Code Review

- Documentation of the code used for model development is provided.
  - Code, files, logs, data sets. This can be documented using a link to GitLab or some other respository
  - Choice of the software
  - Good documentation of the code

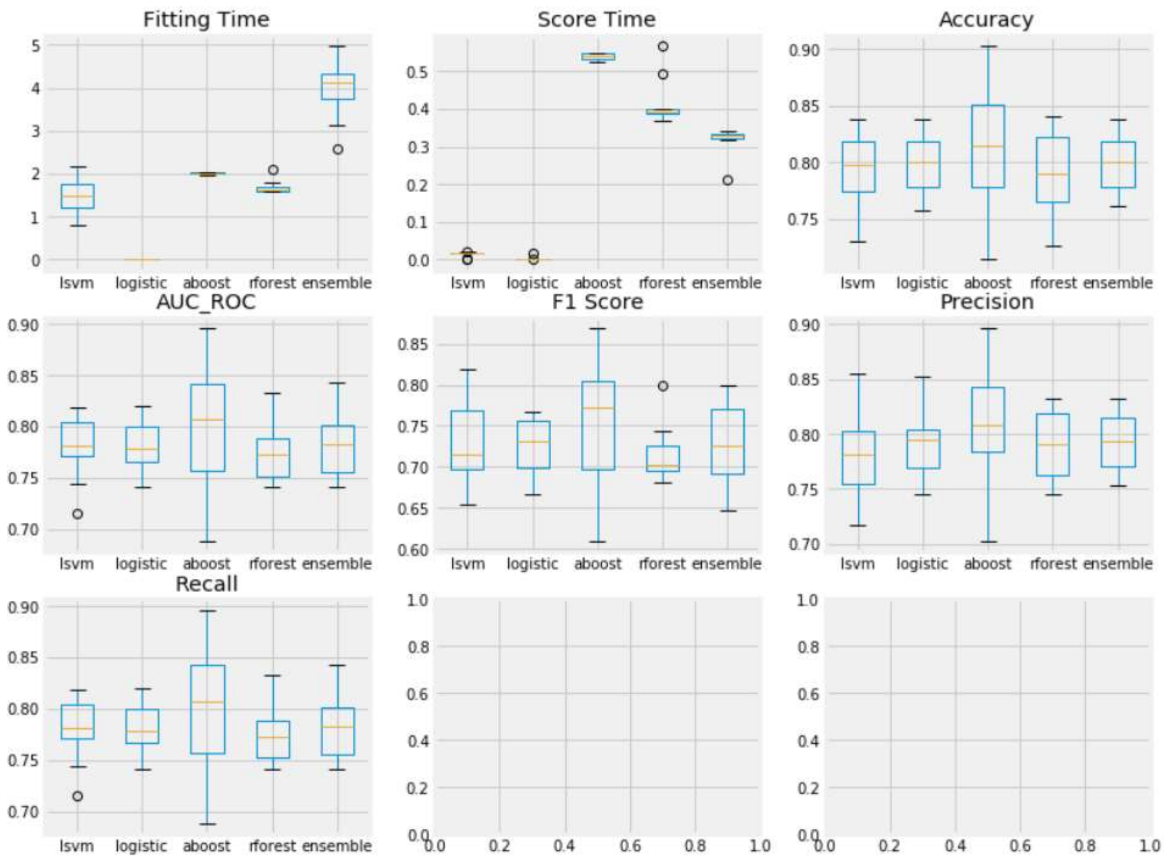
Outcome Analysis

Model performance assessment

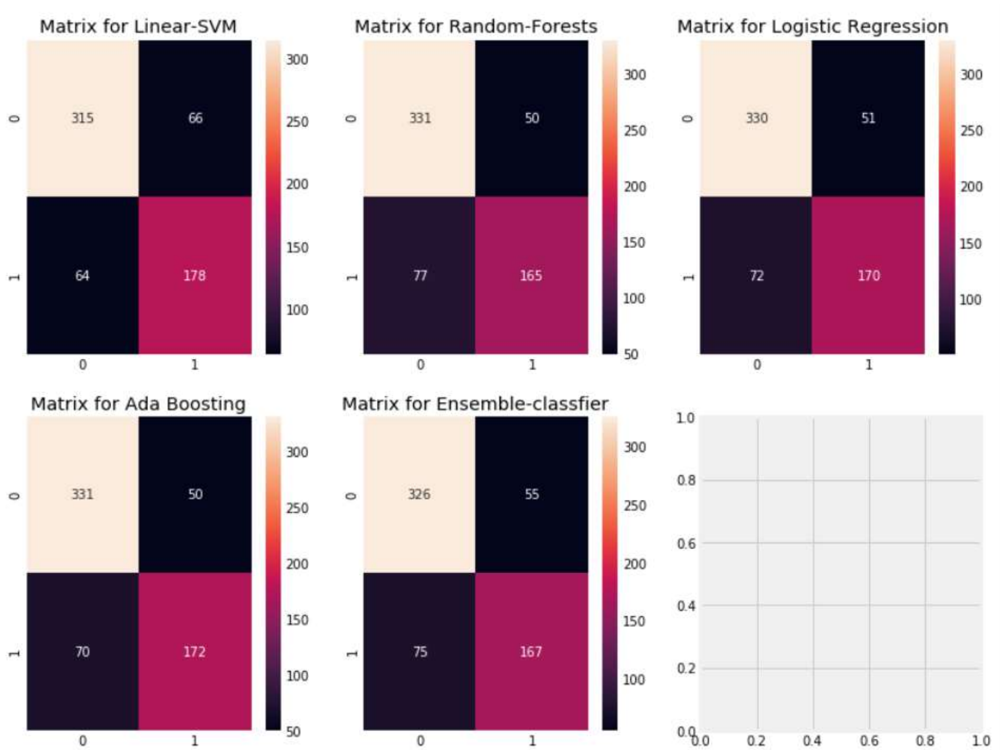
Scoring metrics:

	fit_time	score_time	test_acc	test_auc	test_f1	test_prec	test_rec
Linear Svm	1.476583	0.014687	0.791398	0.780230	0.728721	0.779961	0.780230
Logistic Regression	0.006967	0.002099	0.797798	0.780353	0.725140	0.790021	0.780353
Ada Boosting	2.001520	0.540055	0.813953	0.799265	0.751914	0.806855	0.799265
Random Forest	1.681545	0.418293	0.792934	0.774355	0.715185	0.790159	0.774355
Ensemble	3.952127	0.317602	0.801024	0.784808	0.728686	0.793039	0.784808

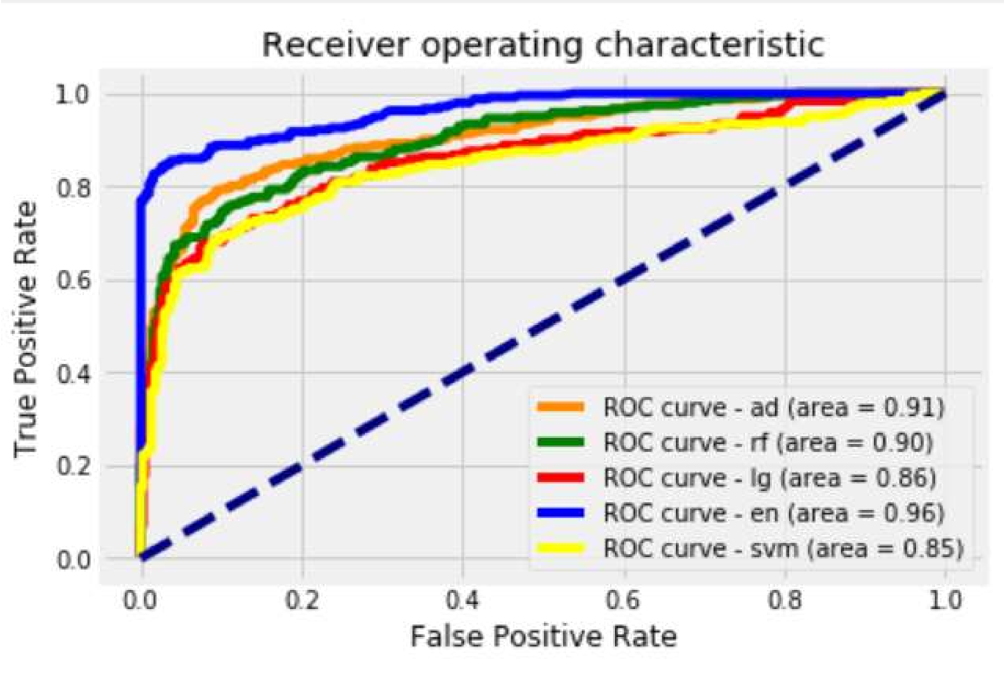
BoxPlot:



Confusion Metrics



ROC Curve:



Additionally Lift chart and Gain chart for metrics

Demonstrating the Model is robust and stable.

Assessing potential limitations.

Assessing the impact of assumptions.

Evaluating Model behavior over a range of input values.

- Do model effects match expectations of business partners? (Effects monotonic? Properly signed?)
- Identifying situations where the Model performs poorly or becomes unreliable.
- Actual circumstances under a variety of market conditions
- Scenarios outside the range of ordinary expectations

A variety of products or applications for which the Model is intended.

Sensitivity analysis

- This is done when the model is used to forecast
- For predictive models used to rank order this is often done by assessing the coefficients

Assessment of the model on an out of time sample.

Swap set analysis

Residual analysis

- Note this is sometimes called Posterior EDA

Implementation

Dual Controls

- Ensure that the data used is in implementation matches the data used for development. (Table 1)
- Control and test that the model is correctly implemented.
  - Code used for comparison
  - Clearly outline the process and the results
- Explicitly adjusting Model inputs or calculations to produce more severe or adverse output in the interest of conservatism.

Overrides

- Overrides of the Model's output by Model Users should be:
  - Documented
  - Tracked to determine rate of occurrence
  - Analyzed for performance
- User access and Security
  - State the users who have access to the model, and how access is managed and controlled.
  - Is production output easily accessible to Model Developers and business partners?

Documentation of the contingency plans including processes for when the model can't be scored.

Table 1 – Process ensuring the data used is in implementation matches the data used for development

Production data	Model build data	Transformations and model scores	Scoring code	Implementation
Data used in production to score the model	Data that was used to build the model either from the production environment, an analytical copy, or an archive	Data set with the scores from the model build.	SAS code that can be used to score a analytical data set based on the output from the model	IT implementation of the model based on the specifications from the modelers.
An exact comparison is not possible for going forward so other methods should be employed to ensure that the data is on the same scale, same variable definitions.				
	The scoring code is created at the end of the model build. The Model Developer should then score out the raw data using the scoring code to ensure that the scoring code gives the same scores as the model build data set. This may not be possible if there is binning or round being done as part of the scoring code.			
			An exact match on the same data is required prior to roll out. Testing of the exception handling is required as well.	

Reporting

Performance Monitoring

Change Control

- Does model require regular refreshes of parameters and auxiliary inputs?
- A documented change control process should be followed for all model changes to ensure:
    - Changes are understood
    - Approved by the Model Owner
    - Evidence of approval is preserved
    - Validation of the change occurs, where appropriate
- Change control procedures should be used to ensure the code implementing the model:
    - Remains accurate and correct
    - Is only altered by approved parties
    - Records evidence of changes and can be audited