Preparing Messy Real World Data For Supervised Machine Learning with `vtreat`

Nina Zumel
John Mount
Win-Vector LLC
http://www.win-vector.com/

Win-Vector LLC

Outline

- Why prepare data?
- Example: modeling with and without data treatment
- `vtreat`: Automating variable treatment in R and Python
- Overfit and Cross-frames
- References

Win-Vector LLC

Why Prepare Data at All?

- To facilitate modeling/analysis
  - Clean dirty data
  - Format data the way machine learning algorithms expect it
- Not a substitute for getting your hands dirty
  - But some issues show up again and again

Win-Vector LLC

Typical Data Problems

- "Bad" numerical values (NA, NaN, None)
- Categorical variables: missing values, missing levels
- Categorical variables: too many levels
- Invalid values
    - New/invalid category levels

Example Problem: KDD2009

- Data set for KDD DataCup 2009
    - Task: predict account cancellation (or “churn”) from supplied features.
- Very messy data
    - 230 columns/variables, 50000 instances
    - Numeric and Categorical values
    - High-cardinality categorical variables
    - Nonsense column and level names, and no data dictionary
    - Many missing values
    - Unbalanced task: churn rate around 7%

KDD model with `xgboost`

(without data treatment)

```
fitter = xgboost.XGBClassifier(
   n_estimators=10,
   max_depth=3,
   objective='binary:logistic')
fitter.fit(d_train, churn_train)
```

```
DataFrame.dtypes for data must be int, float or bool.
                Did not expect the data types in fields
Var191, Var192, Var193, Var194, Var195, Var196, Var197,
Var198, Var199, Var200, Var201, Var202, Var203, Var204,
Var205, Var206, Var207, Var208, Var210, Var211, Var212,
Var213, Var214, Var215, Var216, Var217, Var218, Var219,
Var220, Var221, Var222, Var223, Var224, Var225, Var226,
Var227, Var228, Var229
```

Win-Vector LLC

`xgboost` didn't accept string-values (categorical) variables

- Even if you fixed this you still run into the other issues
  - Missing values
  - High-cardinality categorical variables
  - Rare levels and novel levels

Win-Vector LLC

Grab-bag solution

- Combine
  - `sklearn.preprocessing.OneHotEncoder`
  - `sklearn.impute.SimpleImputer`
  - Maybe `keras.layers.Embedding` to deal with high-cardinality categorical variables

Issues

- Lots of steps to manage.
- May not be statistically efficient.
- Possibly subject to nested model bias/overfit.

What we are advocating

- `vtreat` all in one step
  - Easy to use
  - Known good performance
  - Citable documentation: arXiv 1611.09477 [stat.AP]

vtreat Solution

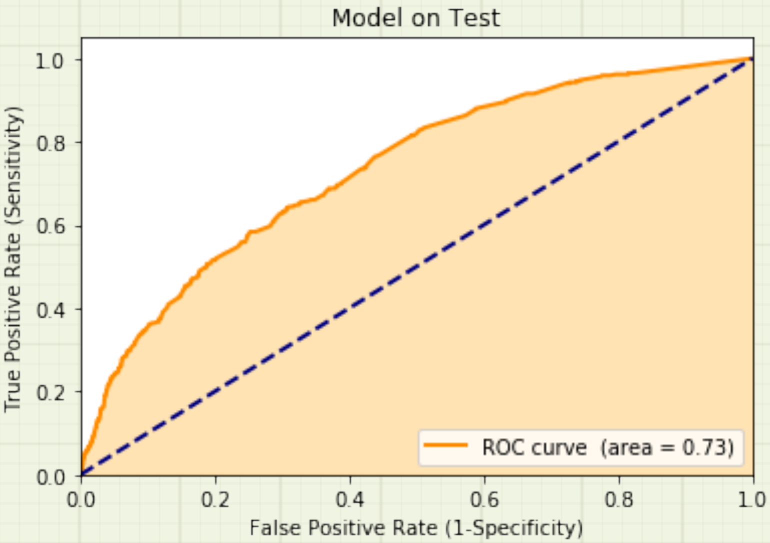
```
import vtreat

# Learn data encoding
plan = vtreat.BinomialOutcomeTreatment(outcome_target=True)
cross_frame = plan.fit_transform(d_train, churn_train)

# convert Pandas sparse representation to
# xgboost sparse representation
scipy.sparse.hstack([scipy.sparse.csc_matrix(cross_frame[[vi]])
                     for vi in model_vars])

# elided code to pick number of trees and select
# modeling variables
…

fitter = xgboost.XGBClassifier(n_estimators=ntree,
                               max_depth=3,
                               objective='binary:logistic')
model = fitter.fit(cross_frame.loc[:, model_vars], churn_train)
```



https://github.com/WinVector/pyvtreat/blob/master/Examples/KDD2009Example/KDD2009Example.ipynb

Result

- `vtreat` first try: AUC performance of 0.73
  - Off the shelf variable treatment
  - Off the shelf machine learning, no hyper-parameter tuning
- Contest-winners: AUC performance of 0.7467 (day one) to 0.7651 (end of contest)
- `vtreat` in the ballpark on first try, leaving time to tune/improve results.
  - A platform to build on

(contest ref: http://proceedings.mlr.press/v7/guyon09/guyon09.pdf)

`vtreat` goal

- Goal: faithfully and reliably convert data into a ready for machine learning data frame that is entirely numeric, and without missing values.
  - By *ready* we mean: the converted information is in a *simple* encoding compatible with linear models, tree models, and neural nets.
  - By *faithful* we mean: *most* of the relevant modeling information is preserved.
  - By *reliable* we mean: a number subtle over-fitting (or nested model bias) traps are avoided.

Win-Vector LLC

vtreat organization

- Built on top of Pandas
  - Assumes columns addressed by name

.score_frame_

variable	orig_variable	treatment	y_aware	PearsonR
x_is_bad	x	missing_indicator	FALSE	-0.01
x	x	clean_copy	FALSE	-0.01
x2	x2	clean_copy	FALSE	0.06
xc_is_bad	xc	missing_indicator	FALSE	-0.38
xc_logit_code	xc	logit_code	TRUE	0.83
xc_prevalence_code	xc	prevalence_code	FALSE	0.41
xc_lev_level_1.0	xc	indicator_code	FALSE	0.77
xc_lev__NA_	xc	indicator_code	FALSE	-0.38
xc_lev_level_0.5	xc	indicator_code	FALSE	0.19
xc_lev_level_-0.5	xc	indicator_code	FALSE	-0.34

From: https://github.com/WinVector/pyvtreat/blob/master/Examples/Classification/Classification.md

What does `vtreat` do?

Win-Vector LLC

1. Fix bad/missing numerical values

- Replace missing values with harmless stand-ins (the mean)
- Add an indicator variable to mark where that happened.
- Extra column makes the treatment "invertible"
  - Preserves maximum information in the data

MPG
50
Inf
20
36.4
NaN
NA

MPG	MPG_isBad
50	0
35.5	1
20	0
36.4	0
35.5	1
35.5	1

Win-Vector LLC

Missingness as signal

- In business analytics missing data is often an indicator of where the data came from and how it was processed.
- Consequently it is often one of your more informative signals when modeling!

2. Encode Categorical Levels as Indicators

- similar to
  `OneHotEncoder(drop=None, handle_unknown='ignore')`
  - novel levels in application data treated as "no level" (all zero indicator variables)
- difference: only **common** levels are dummy encoded

Training Data

Residen
CA
NV
OR
CA
CA
WA

Encoded Training Data

Res_CA	Res_NV	Res_OR	Res_WA
1	0	0	0
0	1	0	0
0	0	1	0
1	0	0	0
1	0	0	0
0	0	0	1

New Data

Residen	
OR	
CA	
WY	(novel level)

Encoded New Data

Res_CA	Res_NV	Res_OR	Res_WA
0	0	1	0
1	0	0	0
0	0	0	0

Win-Vector LLC

3. Manage high-cardinality categorical variables

ZIP	SalePriceK
94127	725
94564	402
90011	386
94704	790
95555	1195
94109	903
94124	625
94123	439
94562	290

- Don't want to have an unbounded number of dummy variables
- Too many levels is a computational problem for many machine learning algorithms.
- You will inevitably have a novel level during application

Manage high-cardinality categoricals (cont.)

- Only dummy encode levels that occur more frequently than some threshold (e.g. 0.02 = 1/50)
  - user-definable threshold
  - prevent "dummy variable blowup"
    - max 1/threshold indicators created (per variable)
- Also convert categorical variables to a single informative numerical variable
  - "impact coding"

“Impact Coding”

ZIP	avgPriceK	ZIP_impact
90011	386	-253.4
94109	903	263.6
94124	532	-107.4
94127	960	320.6
94564	346	-293.4
94704	790	150.6
globalAvg	639.4	0

Win-Vector LLC

Impact-coding the ZIP variable

ZIP
94127
94564
90011
94704
94127
94109
94124
94124
93401

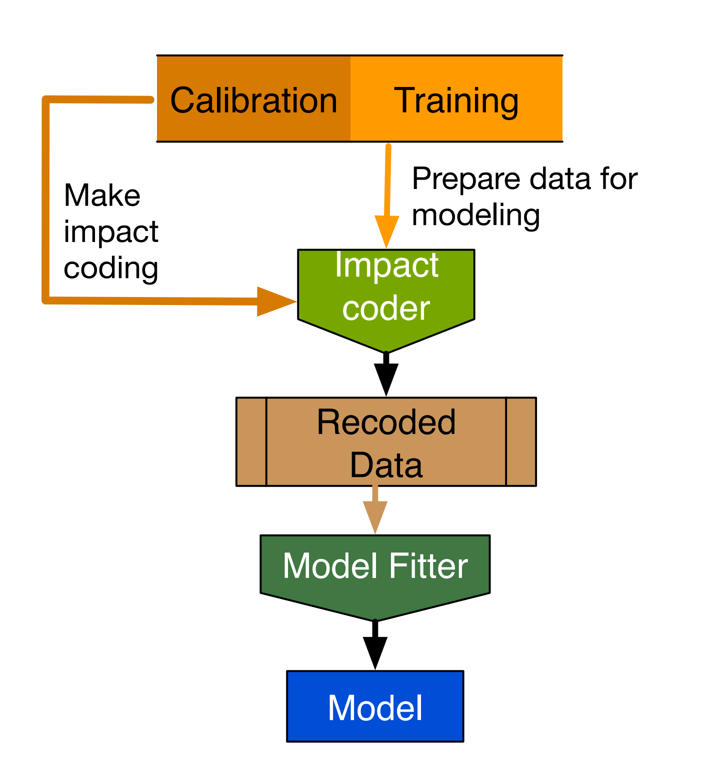
(novel level)

ZIP_impact
320.6
-293.4
-253.4
150.6
320.6
263.6
-107.4
-107.4
0

Win-Vector LLC

Don't Naively Use Training Data to Impact Code!

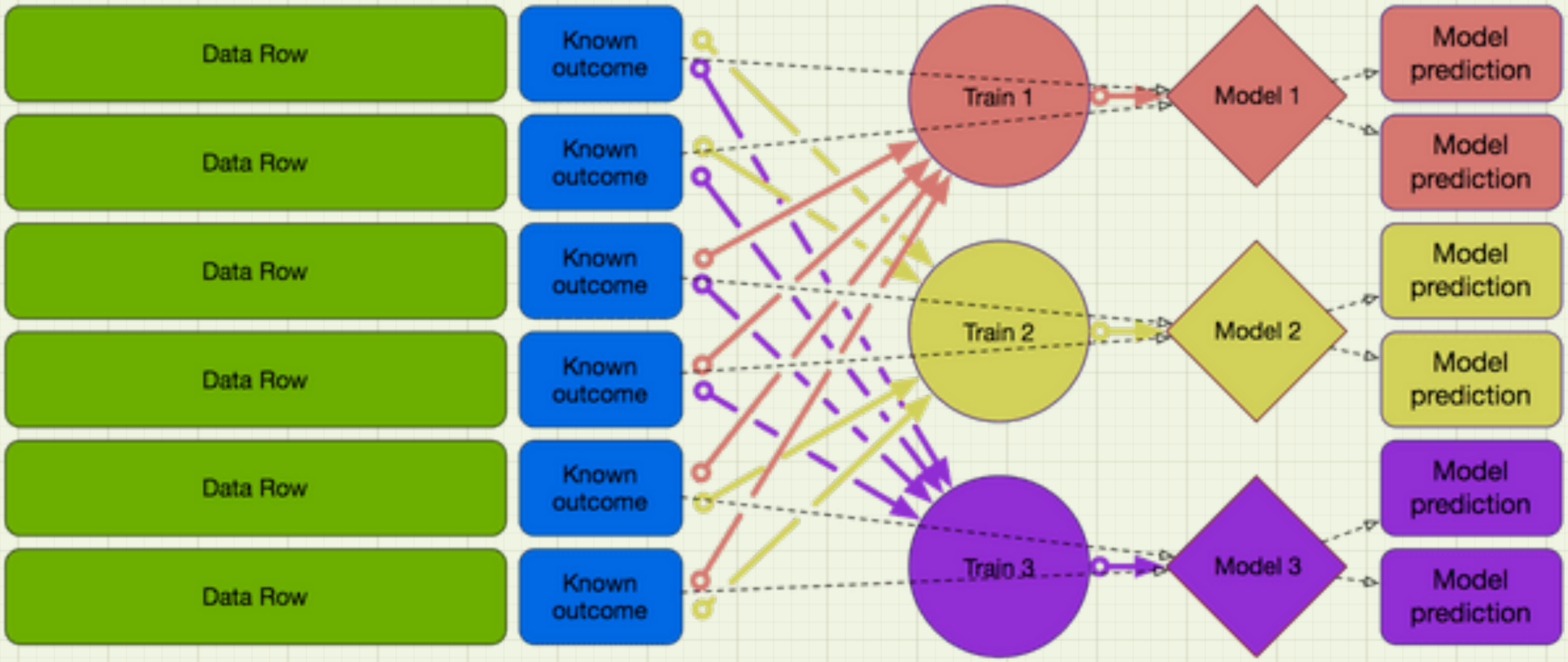
- Can introduce undesirable nested model bias
    - Full model may overestimate value of impact coded variable
    - Useless high-complexity variables cut in front of other variables.
    - Overall model is bad
- Use separate calibration data or cross-validation methods
    - vtreat has built-in cross-validation methods that avoid these issues



http://www.win-vector.com/blog/2019/11/when-cross-validation-is-more-powerful-than-regularization/

Win-Vector LLC

Cross-Frames



Idea:

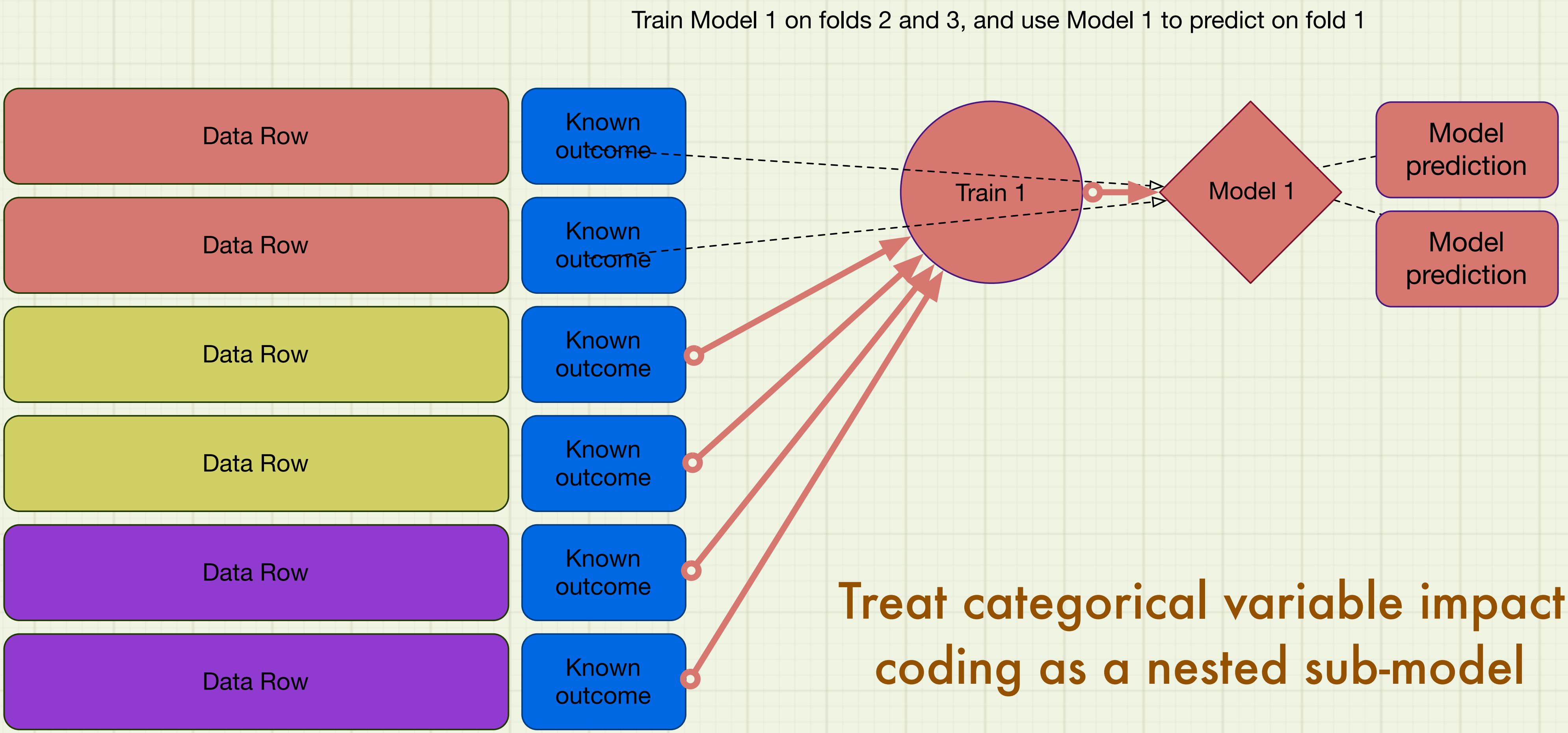
no row should be processed by a model it was included in the construction of.

Win-Vector LLC

Training data

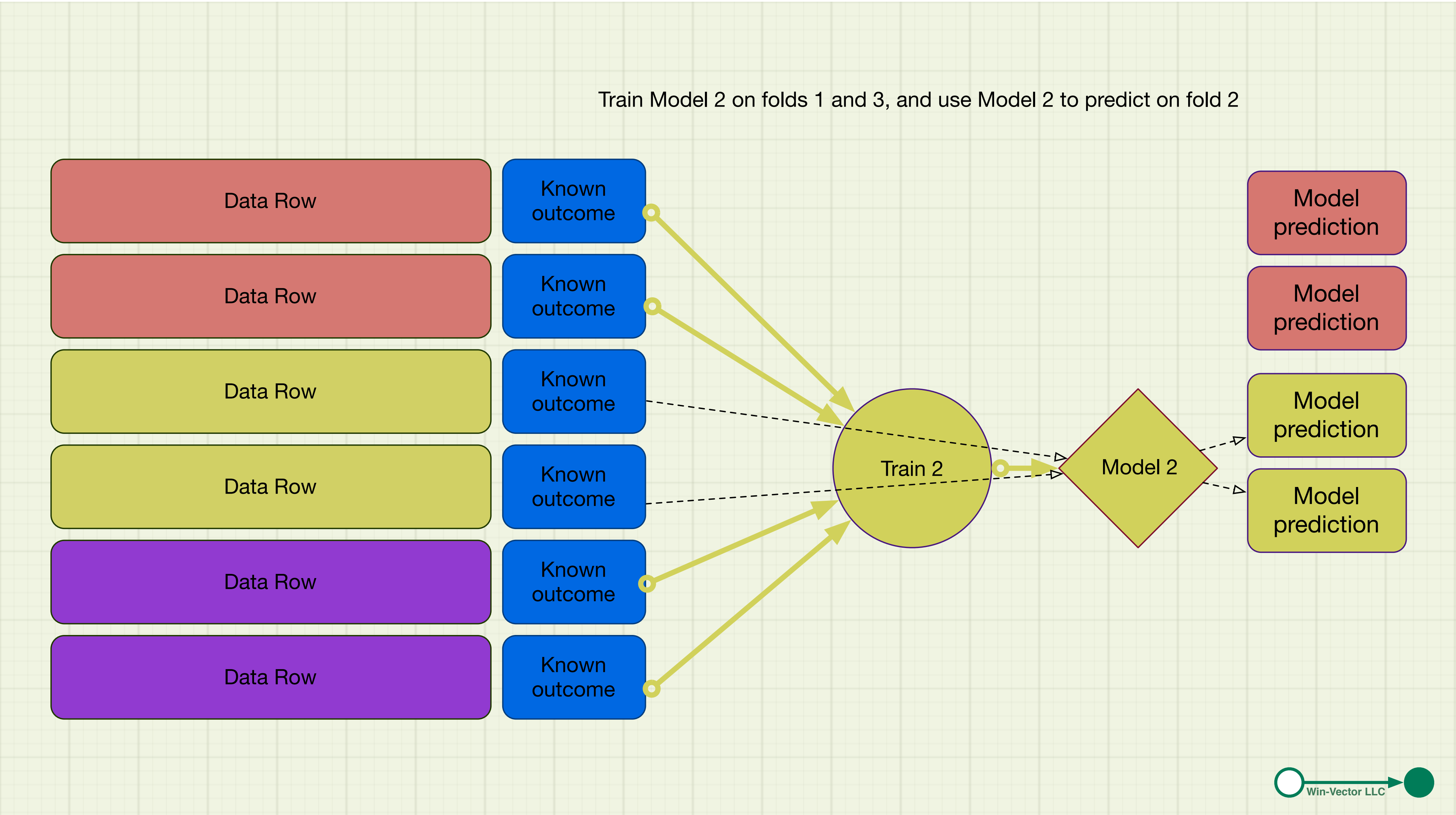
Partition the data into k folds (k = 3)

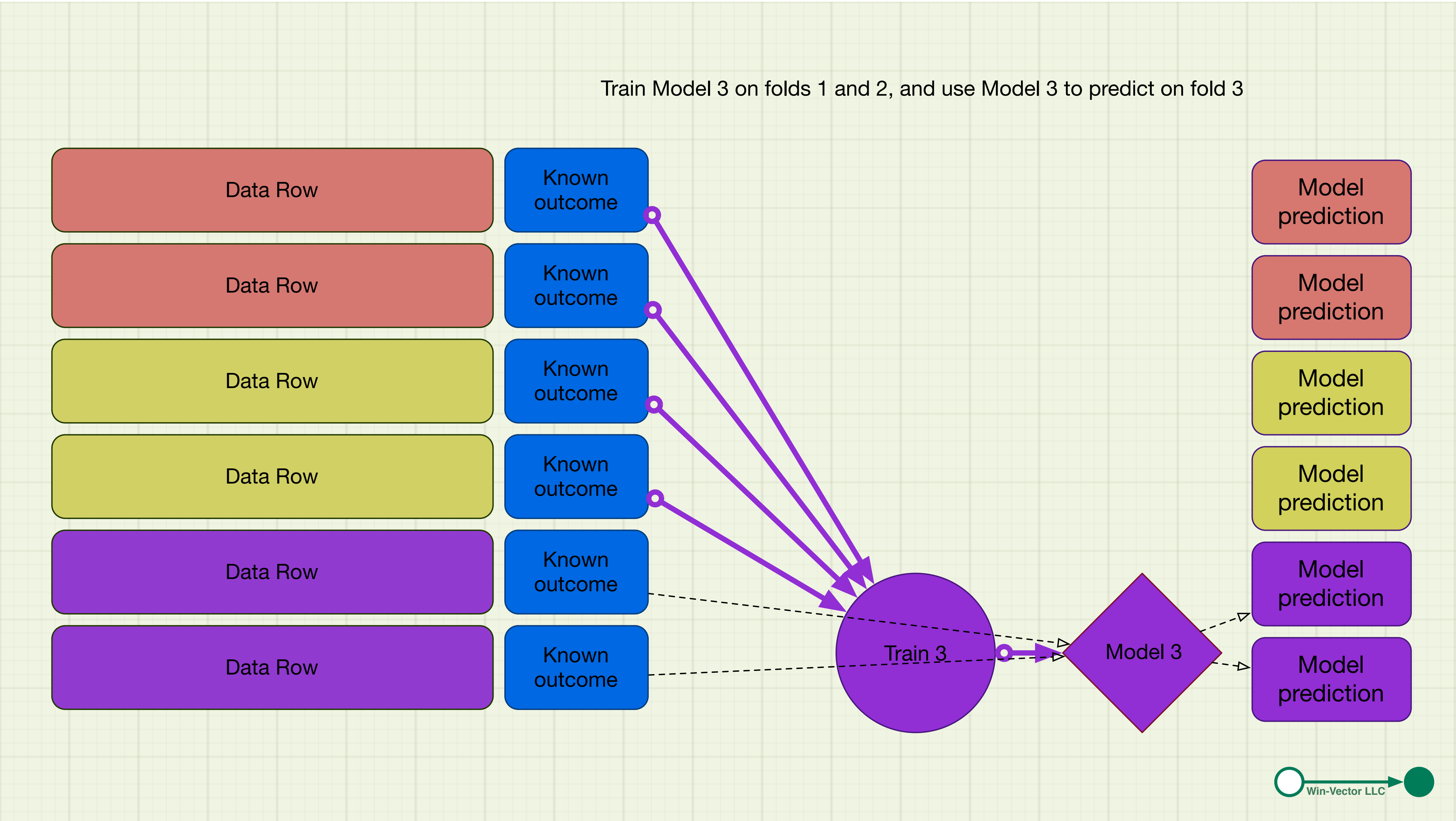
Win-Vector LLC



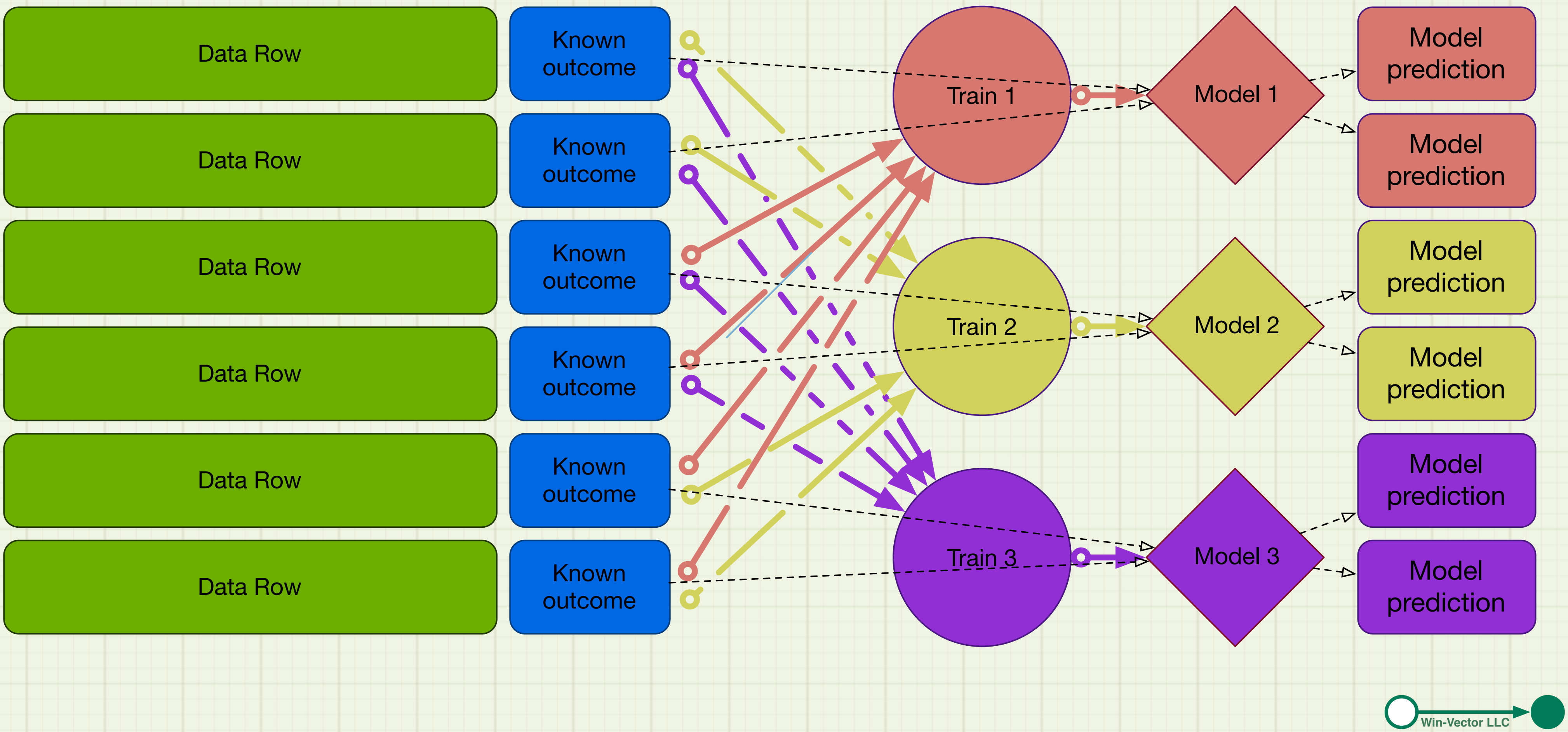
Treat categorical variable impact coding as a nested sub-model

Win-Vector LLC


Train Model 2 on folds 1 and 3, and use Model 2 to predict on fold 2
Data Row
Data Row
Data Row
Data Row
Data Row
Data Row
Known outcome
Known outcome
Known outcome
Known outcome
Known outcome
Known outcome
Train 2
Model 2
Model prediction
Model prediction
Model prediction
Model prediction
Win-Vector LLC


Train Model 3 on folds 1 and 2, and use Model 3 to predict on fold 3
Data Row
Known outcome
Data Row
Known outcome
Data Row
Known outcome
Data Row
Known outcome
Data Row
Known outcome
Data Row
Known outcome
Train 3
Model 3
Model prediction
Model prediction
Model prediction
Model prediction
Model prediction
Model prediction
Win-Vector LLC

Now every row has a predicted outcome based on a model that was not trained on that row.



The `vtreat` calling pattern on training data

- Naive call:

  ~~`.fit(X_train, y).transform(X_train)`~~

- Calibration set method:

  `.fit(X_train_cal, y).transform(X_train_model)`

- Cross-frame method:

  `.fit_transform(X_train, y)`

Win-Vector LLC

The `vtreat` calling pattern on test or application data

```
.transform(X_test)
```

Win-Vector LLC

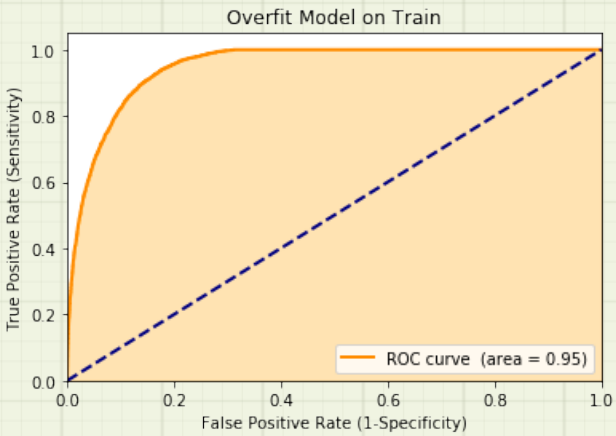
The value of cross-frames

`.fit(X, y).transform(X)`

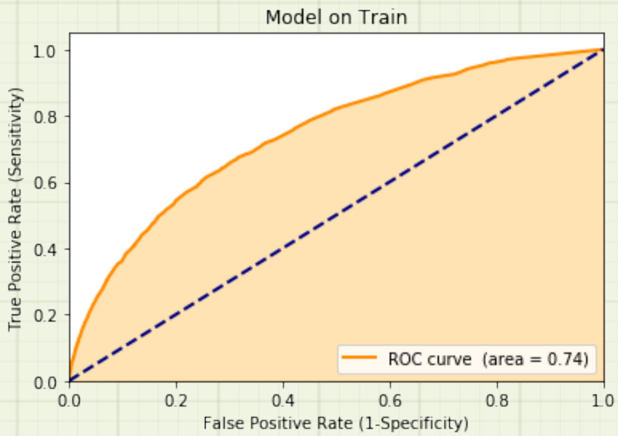
(naive, also what some ad-hoc methods will do)

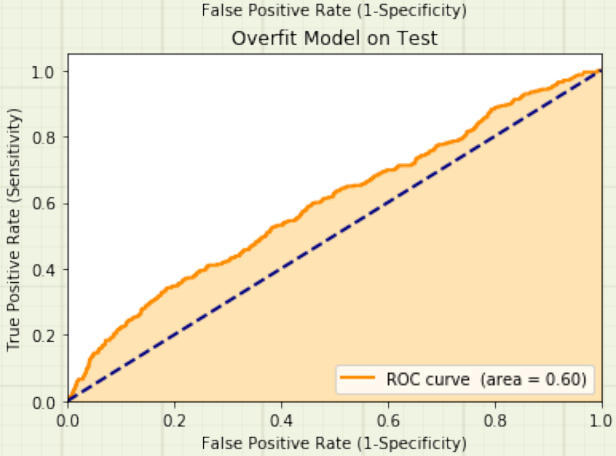
`.fit_transform(X, y)`

(cross-frames)

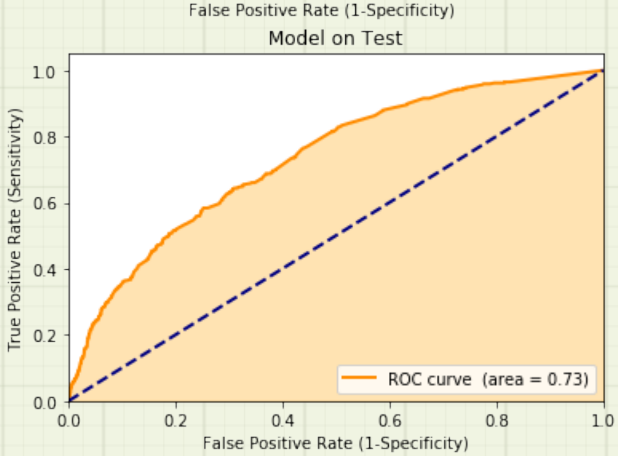


training set performance





test set performance



Win-Vector LLC

Conclusion

- `vtreat` is an easy-to-use, statistically sound process to prepare data for machine learning
  - prepared data is entirely numeric, and without missing values
- It's no substitute for getting your hands in the data
- Automating common data steps leaves the data scientist more time to develop high-value domain- or problem-specific ideas

References

- Impact Coding (Kagglers call this target coding)
  - http://www.win-vector.com/blog/2012/07/modeling-trick-impact-coding-of-categorical-variables-with-many-levels/
  - http://www.win-vector.com/blog/2012/08/a-bit-more-on-impact-coding/
  - “vtreat paper” https://arxiv.org/abs/1611.09477
- R vtreat:
  - https://github.com/WinVector/vtreat/
  - Install from CRAN with "`install.packages(vtreat)`"
- Python/Pandas vtreat:
  - https://github.com/WinVector/pyvtreat
  - Install from PyPi with "`pip install vtreat`"

Win-Vector LLC

Thank You

Win-Vector LLC