Data Science Competitions - Flu Shot Learning: Predict H1N1 and Seasonal Flu Vaccines

Students: Vanilton Paulo and Nhi Nguyen
Supervisor: Giuseppe Casalicchio
24/07/2025

Acknowledgements

Dear Mr. Giuseppe,

Thank you for your expert guidance and insightful feedback, which elevated the rigor, clarity, and impact of our data science presentation. Your mentorship and support have been invaluable.

Agenda

1. Introduction to the Project
2. Descriptive and Exploratory Data Analysis
3. Introduction to Model Approach
4. Final Model
5. Future Ideas

Introduction to the Project

- A Data Science competition on DrivenData.org which was launched back when Covid-19 vaccines were still under development.
- **Flu Shot Learning**: Predict H1N1 and Seasonal Flu Vaccines
- Research Question:

Can we predict whether people got H1N1 and seasonal flu vaccines using information they shared about their backgrounds, opinions, and health behaviors?

History of the Data

- Beginning in spring 2009, a **pandemic was caused by H1N1 influenza virus** (swine flu) with an estimate of 151k to 575k deaths globally in the first year
- A public vaccine was released in October 2009
- A **phone survey** has been conducted in late 2009/early 2010
  - Has the person received H1N1 and seasonal flu vaccines
  - Questions about themselves

Information about the Data

- The data comes from the **National 2009 H1N1 Flu Survey (NHFS)**
- NHFS was list-assisted **random digit dialing telephone survey of households** monitoring influenza immunization coverage in 2009-2010
- **Target population**: all people, 6 months or older living in the US at the time of the interview
- Were used to produce timely **estimates of vaccination coverage rates** for both monovalent pH1N1 and trivalent seasonal influenza vaccines

Data Use Restrictions

- Public Health Service Act provides the data collected by National Center for Health Statistics (NCHS) only to be used for **health statistical reporting and analyses.**
- National Center for Health Statistics (NCHS) ensures **anonymity** by removing all direct identifiers.
- Do not make use of the identity of any person or establishment discovered inadvertently and advise the Director, NCHS, of any such discovery.
- Do not link these data files with individually identifiable data from other NCHS or non-NCHS data files.

Descriptive and Exploratory Data Analysis

- Get an overview of the data we are working with
- Explore missing data
- Investigate distribution of target variables
- Analyze correlation of the outcomes

Our Data

- Relevant data:
    - Training Features
    - Training Labels
    - Test Features
- 26.707 observations
- 35 features
- 2 target variables

Top 3 Features have around 50% of missing values

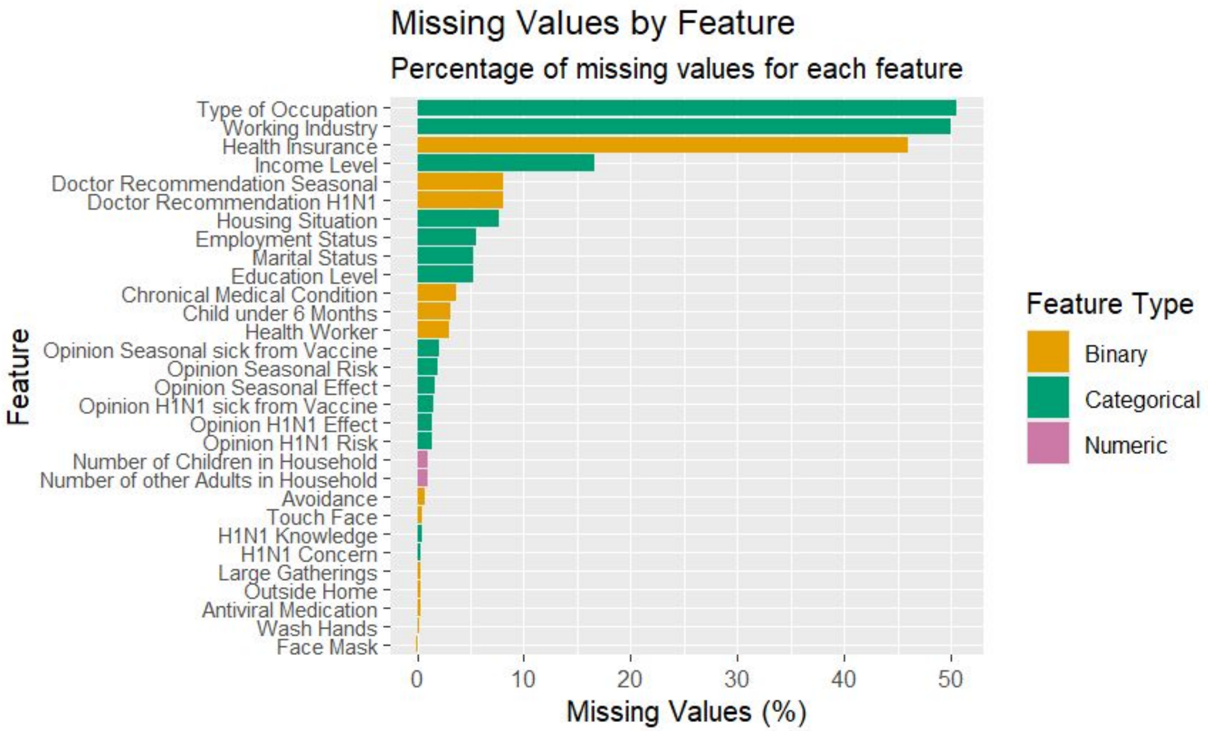


Fig. 1 Plot portraying missing values for every feature

More non-vaccinated than vaccinated

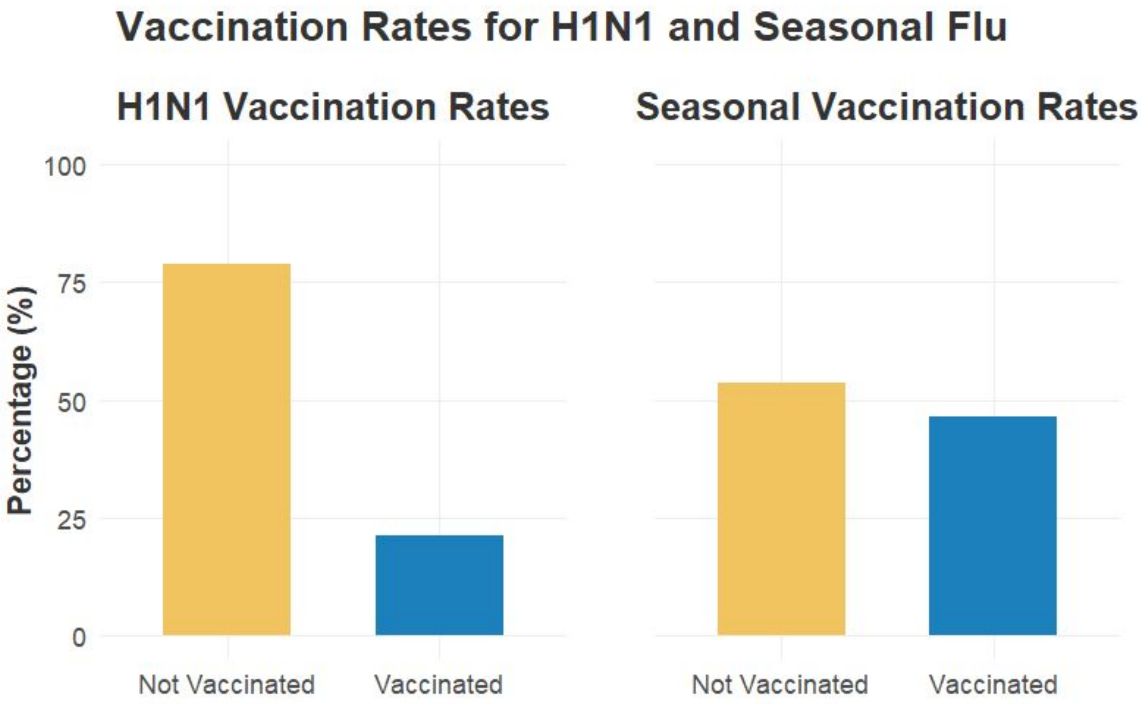


Fig. 2 Plot portraying vaccination rates for H1N1 and Seasonal Flu

Moderate positive correlation between H1N1 and Seasonal Vaccine

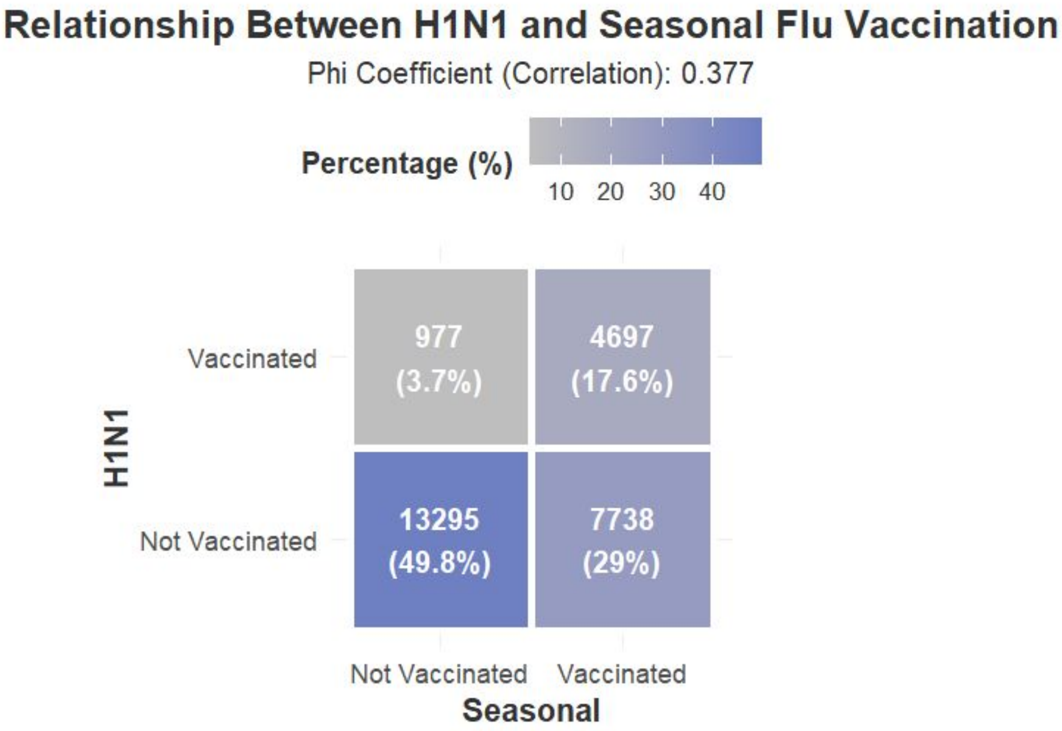


Fig. 3 Heatmap portraying correlation between target variables

Factors influence H1N1 Vaccinations only to a small extent

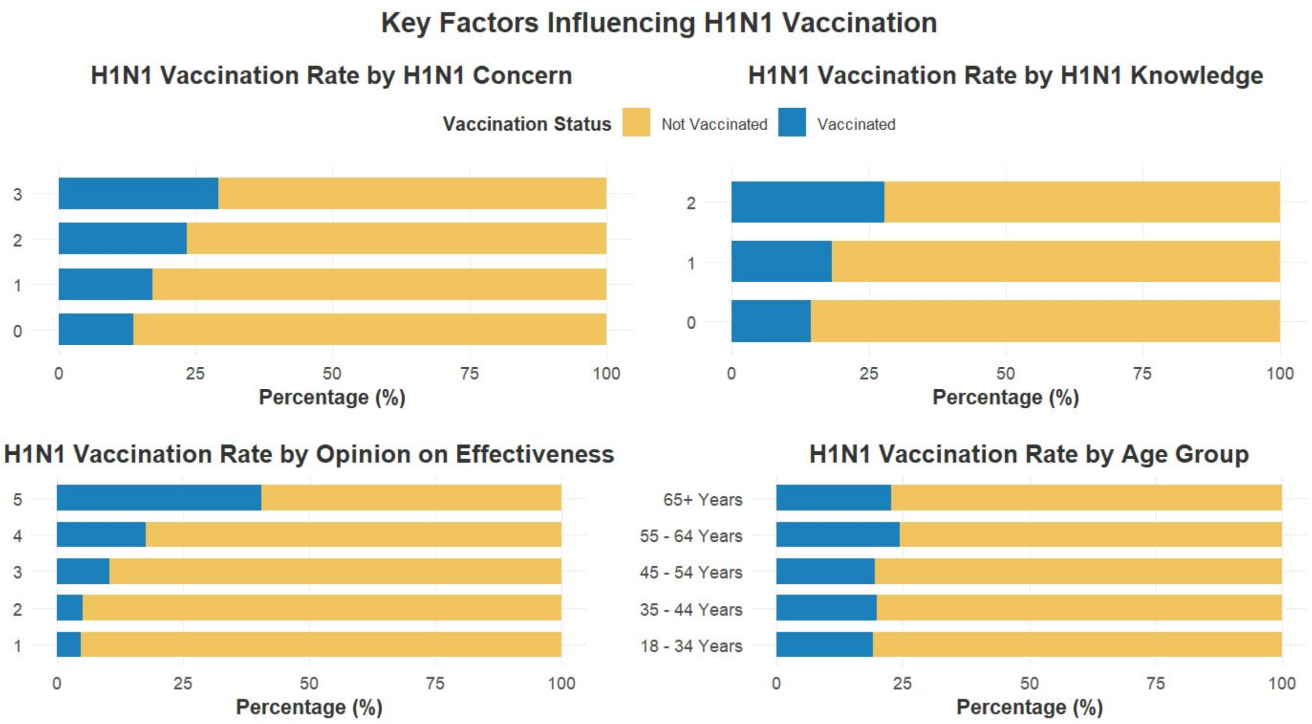


Fig. 4 Plot portraying distribution of factors that influence H1N1 Vaccination

Factors have a substantial influence on Seasonal Vaccinations

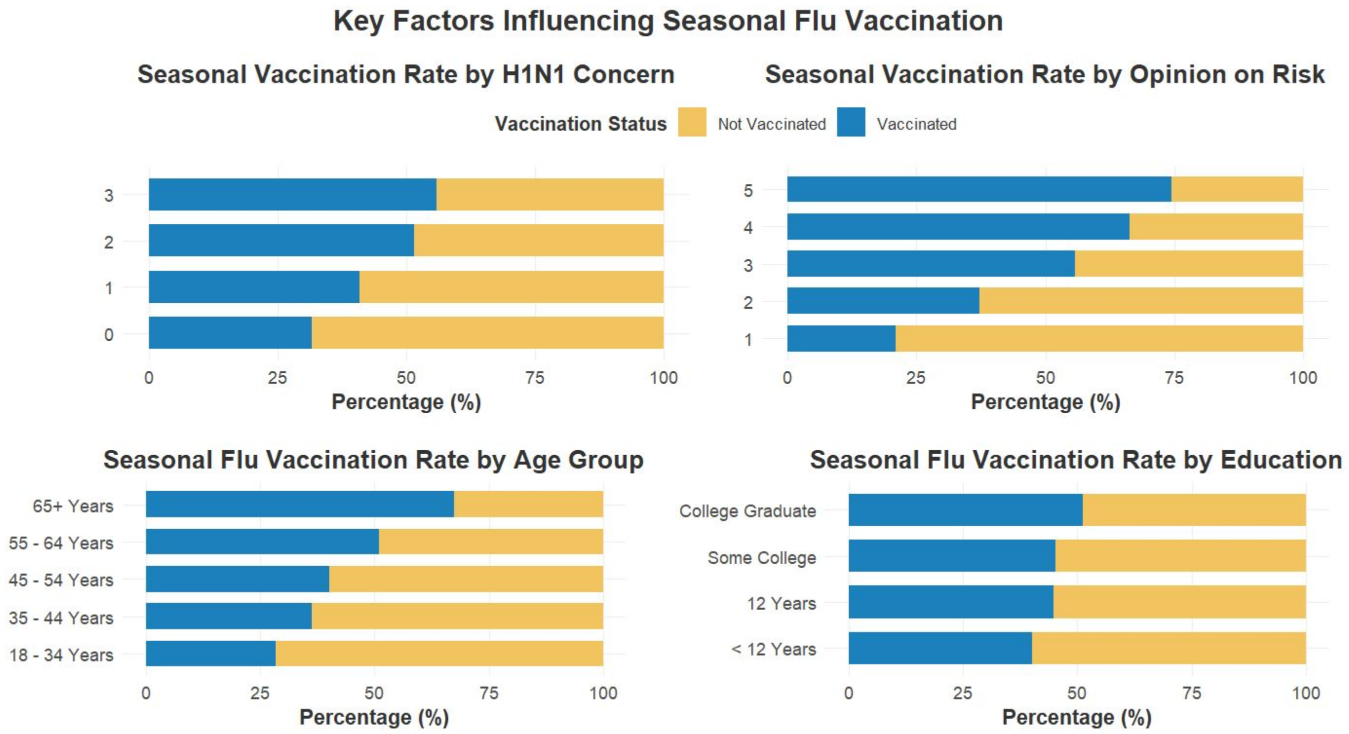


Fig. 5 Plot portraying distribution of factors that influence Seasonal Vaccination

Conclusion

- Knowledge- and opinion-based questions show strong predictive power for both vaccinations.
- **Age group** shows clear positive association with Seasonal Flu vaccination but no similar association with H1N1 vaccination.

Modeling Approach

Introduction to Model Approach (with tidymodels)

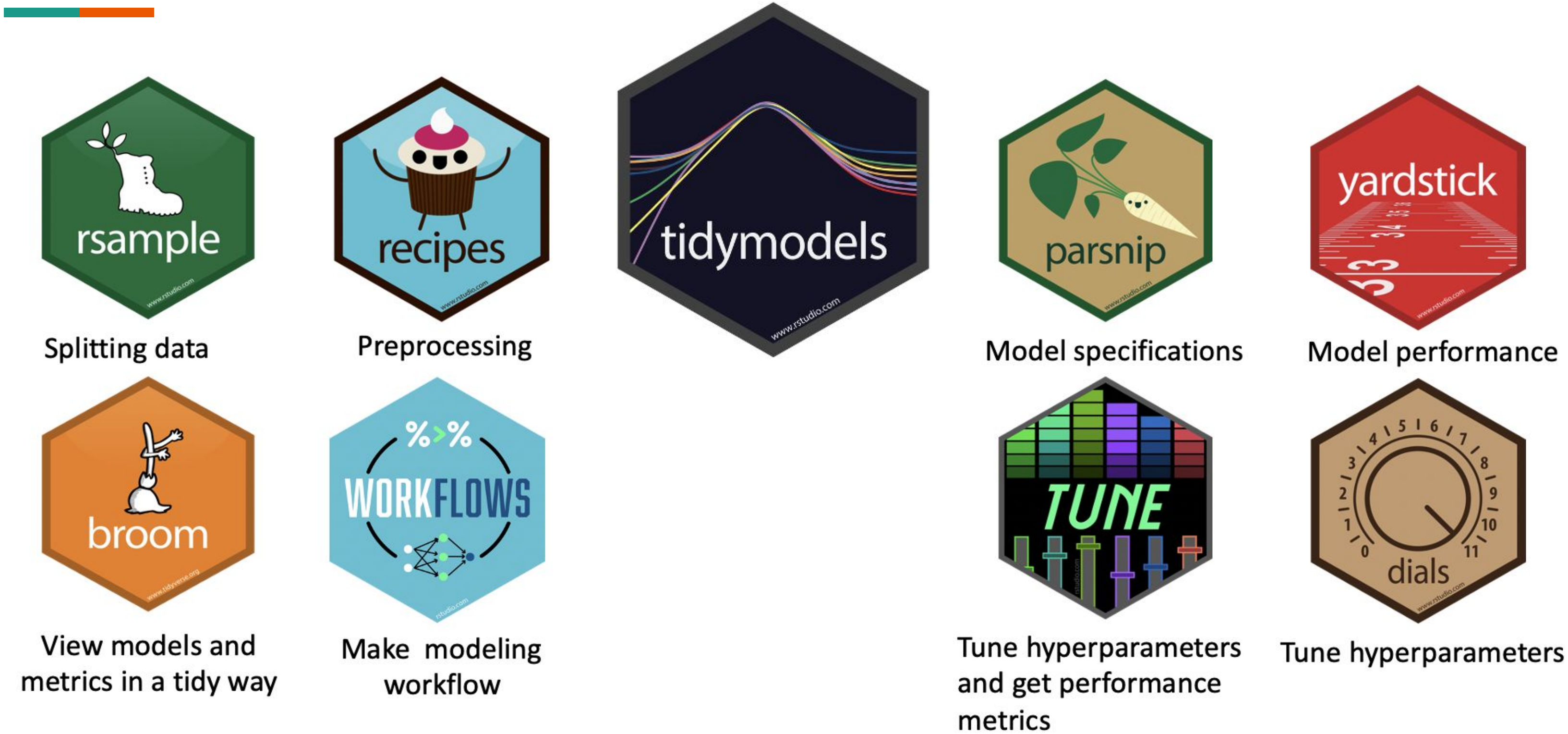


Fig. 6 Chen Xing, “Tidymodels Ecosystem Tutorial”, Overview of relevant Packages

Tidymodels Ecosystem

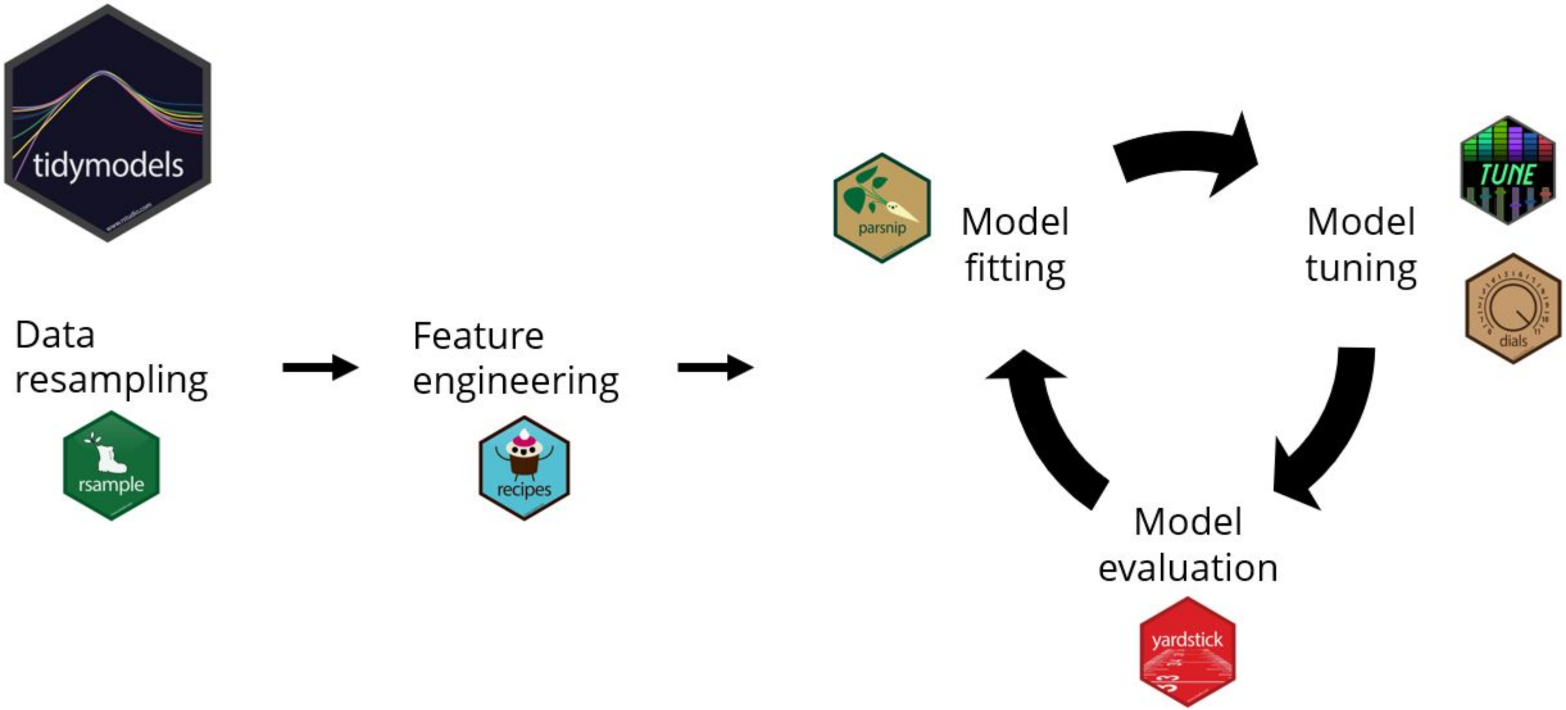


Fig. 7 Chen Xing, “Tidymodels Ecosystem Tutorial”, Basic Tidymodels Ecosystem

Supervised Machine Learning (ML)

A branch of ML that uses **labeled data** for model fitting.

Classification:

- Predicts **binary** outcomes
- Whether someone was vaccinated with the **H1N1 and/or Seasonal Flu Vaccine**

employment_industry	employment_occupation	h1n1_vaccine
NA	NA	0
pxcmvdjn	xgwztkwe	0
rucpziij	xtkaffoo	0
NA	NA	0
xicduogh	xtkaffoo	0

Fig. 8 Glimpse of Dataset

Tidymodels **variables**

- ***h1n1_vaccine and/or seasonal_vaccine*** is the **outcome** variable
- All other variables are **predictor variables**

Our Base Model: Logistic Regression - Example

Purchased	Total time	Total visits
yes	800	3
yes	978	7
no	220	4
no	124	5
yes	641	4

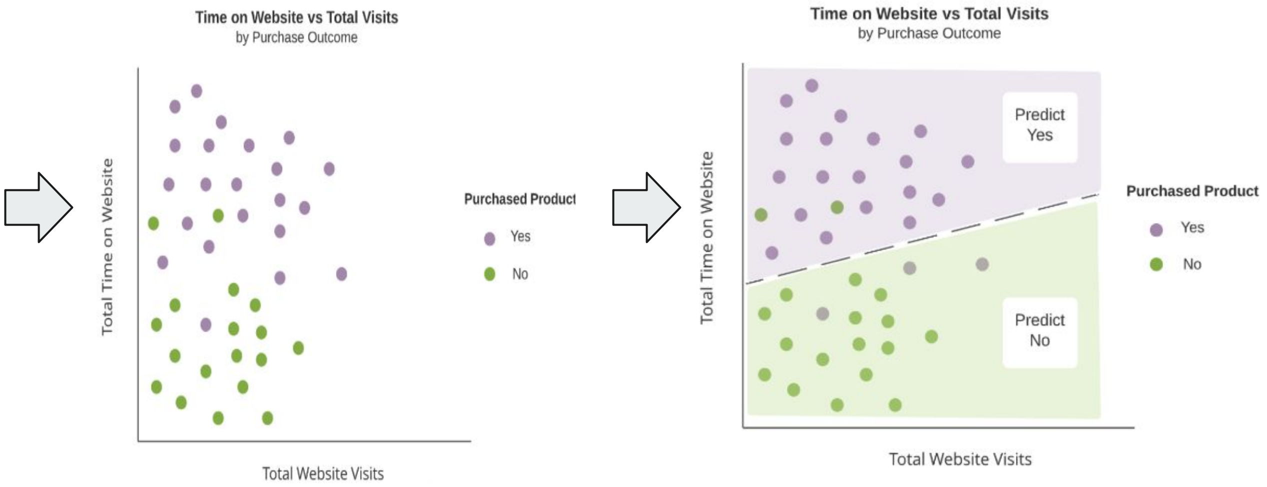


Fig. 9 Data Camp, Example of a Logistic Regression

Why Logistic Regression as a Base Model?

- Logistic Regression is fairly easy to:
  - Implement,
  - Interpret and
  - Train
- Quick
- Makes no assumptions about predictor variable distributions

The WORKFLOW

Data Resampling

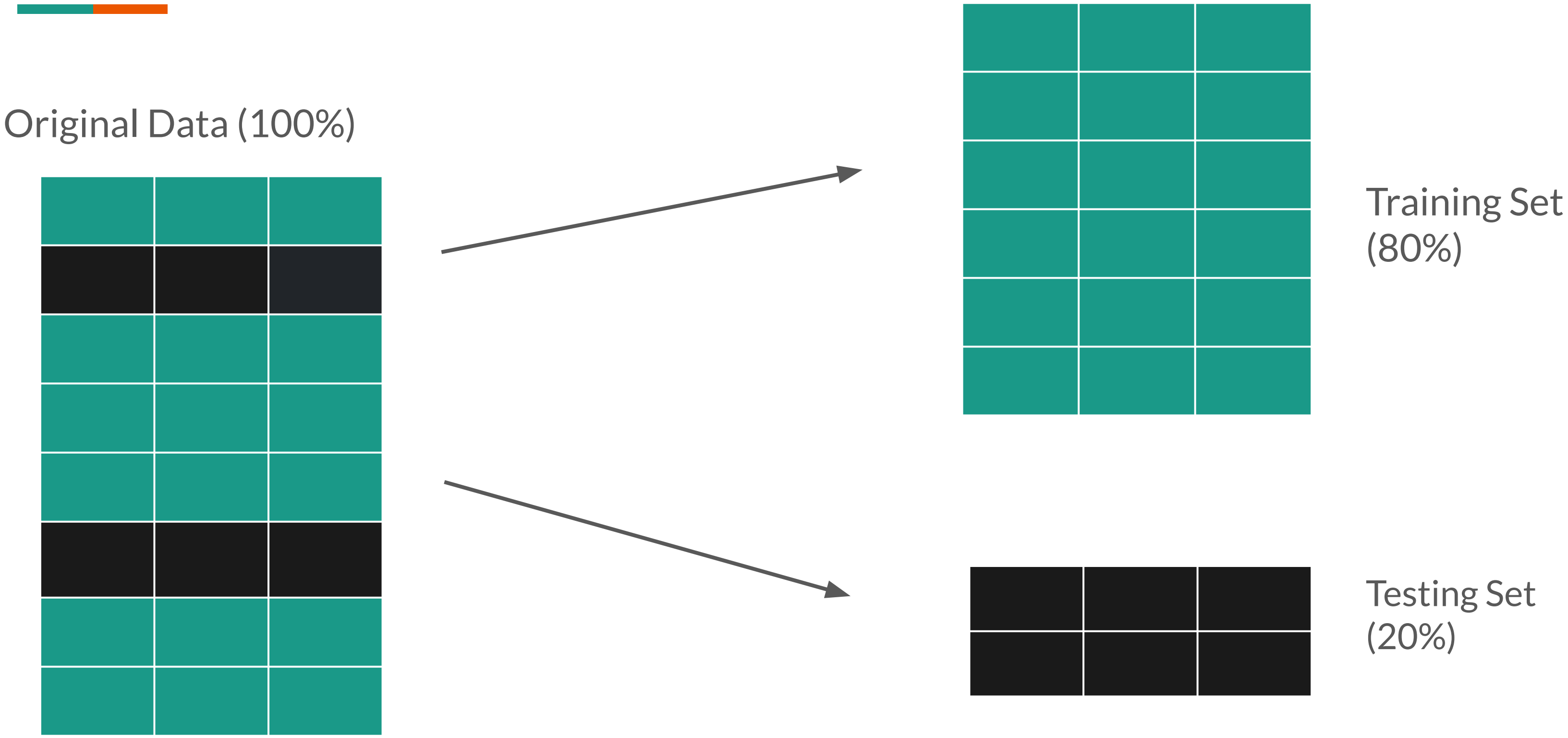


Fig. 10 Data splitting

Model Specification for both vaccinations

```
══ Workflow ════════════════════════════════════════════════
Preprocessor: Recipe
Model: logistic_reg()

── Preprocessor ────────────────────────────────────────────
11 Recipe Steps

• step_rm()
• step_impute_median()
• step_unknown()
• step_dummy()
• step_interact()
• step_interact()
• step_interact()
• step_interact()
• step_interact()
• step_zv()
• ...
• and 1 more step.

── Model ───────────────────────────────────────────────────
Logistic Regression Model Specification (classification)

Main Arguments:
  penalty = 1
  mixture = 0

Computational engine: glmnet
```

Fig. 11 Model Specifications

Specify a logistic regression model using parsnip

Feature Engineering

Application of regularization to the model to prevent overfitting
Ridge Regression (L2 penalty only)

Fit and train Workflow

- Bundle together:
    - Data Preprocessing,
    - Modeling and

-> No need to keep track of separate objects in the workspace

- Train the workflow

Evaluate the Model

- Use earlier created **test set**
- Predict class probabilities
- This competition evaluates the models performance using **Receiver Operating Characteristic (**ROC) curves and Area under the Curve (AUC)
- Compute **Confusion Matrix and ROC curve**

Assessing Performance Metrics

Confusion Matrix

A matrix that summarizes the **performance of a classification model** by comparing its **predicted labels to the true labels**

		Actual	
		Positive (+)	Negative (-)
Predicted	Positive (+)	True Positives (TP)	False Positives (FP)
	Negative (-)	False Negatives (FN)	True Negatives (TN)

Fig. 12 Confusion Matrix

The majority has been identified as non-vaccinated

Confusion matrix of Predicted vs. Actual
(H1N1 Vaccine)

Predicted \ Actual	1	0
1	226	42
0	909	4165

Confusion matrix of Predicted vs. Actual
(Seasonal Vaccine)

Predicted \ Actual	1	0
1	1756	513
0	731	2342

Fig. 13 Confusion Matrix of predicted vs. actual values

Data Metrics - Calculating Metrics from the confusion matrix

Accuracy

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Sensitivity

$$\frac{TP}{TP + FN}$$

Specificity

$$\frac{TN}{TN + FP}$$

Fig. 14 Custom Metrics

Results

Custom Metrics Summary for H1N1 Vaccine Predictions

Metric	Estimator	Estimate
Accuracy	binary	0.822
Sensitivity	binary	**0.199**
Specificity	binary	0.990

Custom Metrics Summary for Seasonal Vaccine Predictions

Metric	Estimator	Estimate
Accuracy	binary	0.767
Sensitivity	binary	0.706
Specificity	binary	0.820

Table 1 Custom Metrics for both outcomes

Visualizing Model Performance

- Visual representation of models performance across thresholds.
- Generated by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various classification thresholds

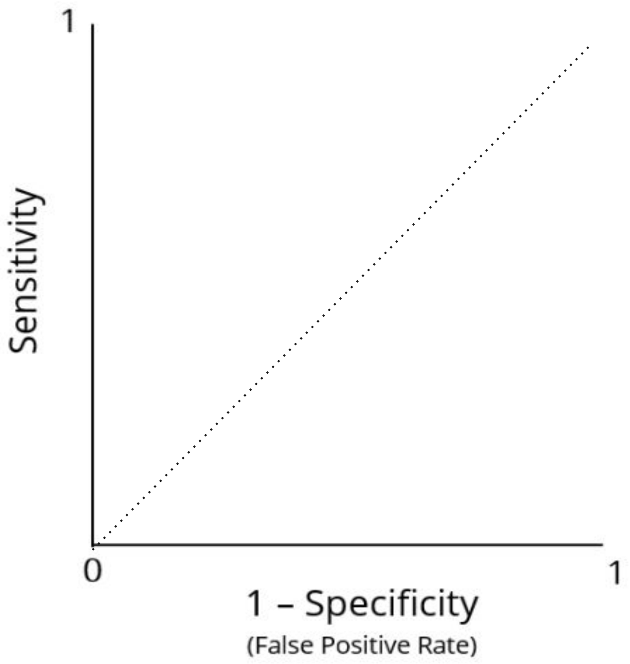
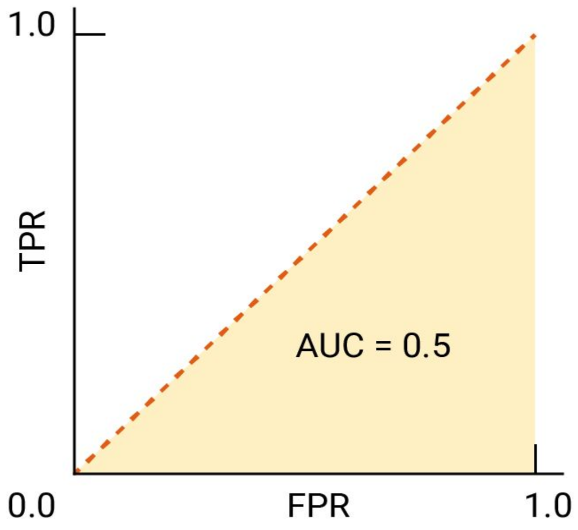


Fig. 15 ROC curve

AUC and ROC for choosing Model and Threshold

Random



Perfect

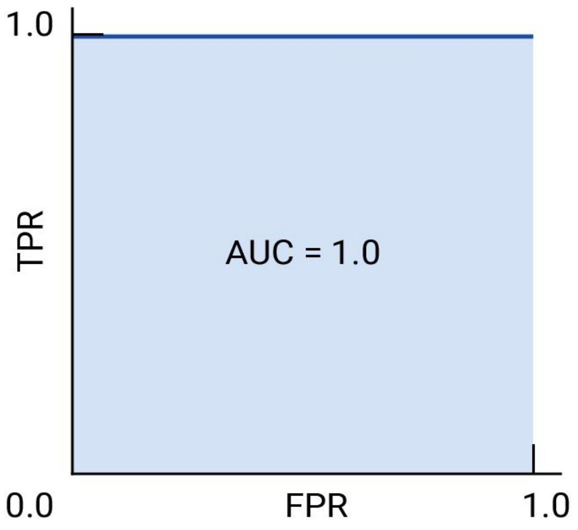
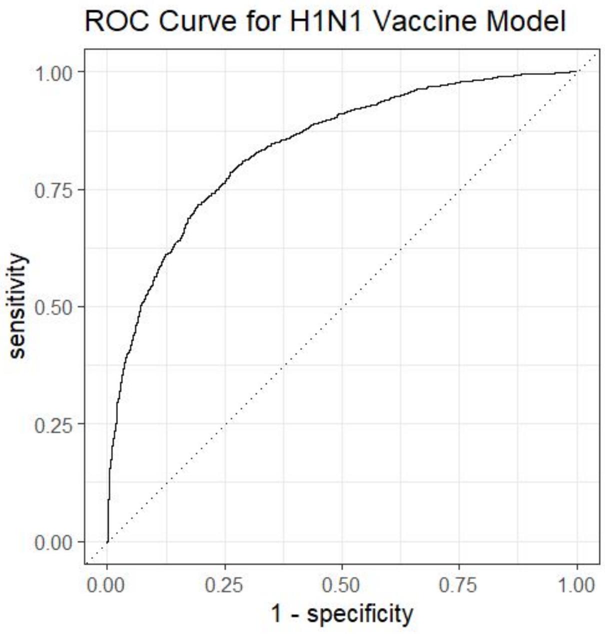
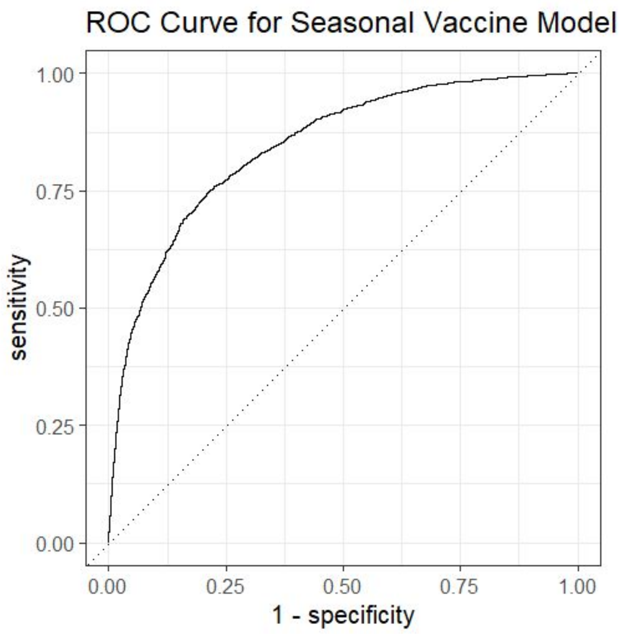


Fig. 16 Google Developers, “Classification: ROC and AUC”, Random and perfect ROC curves

ROC curves and area under the ROC curve



AUC : 0.839



AUC : 0.846

Fig. 17 ROC curves of Logistic Regression Model pre-tuning

Cross Validation

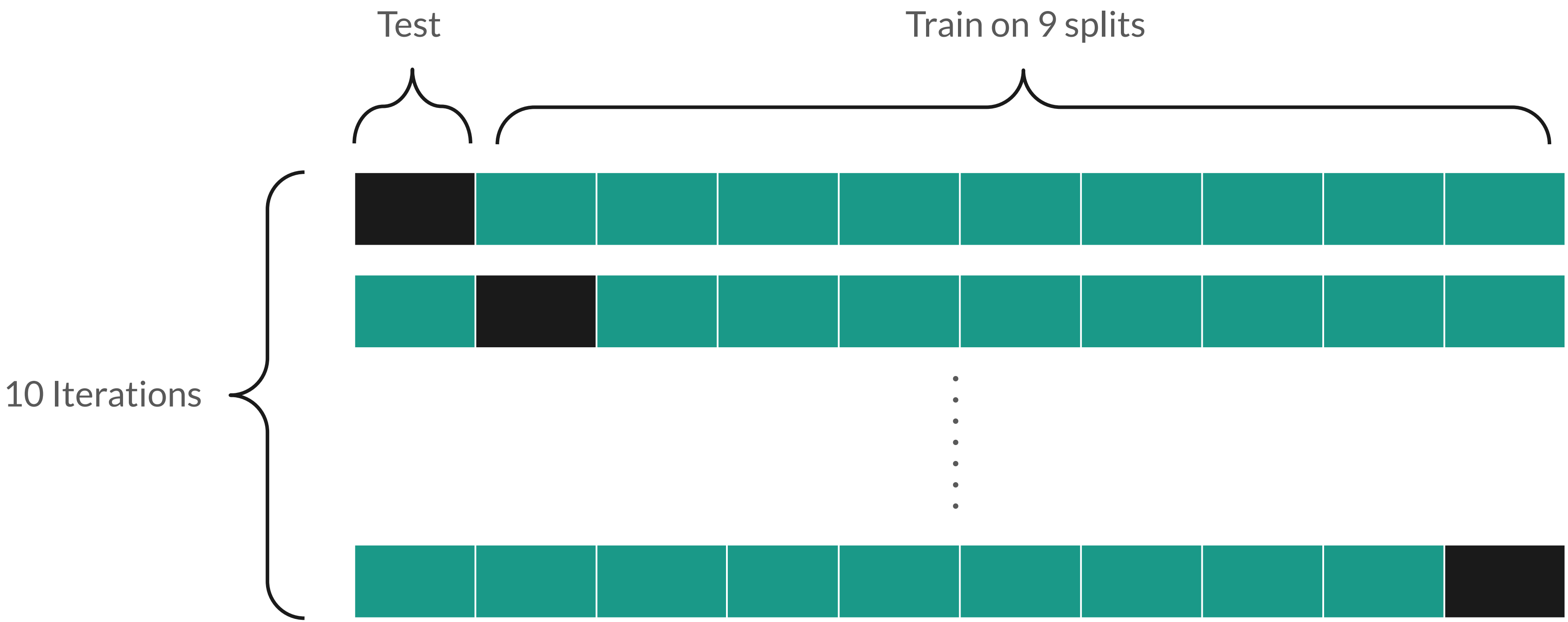


Fig. 18 Cross Validation theory

CV with Logistic Regression - Measuring performance

10 folds cross validation for H1N1 Vaccine Predictions

Metric	Min	Median	Max	Standard Deviation
Accuracy	0.814	0.819	0.829	0.00475
AUC	0.832	0.849	0.865	0.0115
Sensitivity	0.150	0.193	0.227	0.0208
Specificity	0.987	0.989	0.993	0.00241

Table 2 Cross Validation results for H1N1 Vaccine Predictions

CV with Logistic Regression - Measuring performance

10 folds cross validation for Seasonal Vaccine Predictions

Metric	Min	Median	Max	Standard Deviation
Accuracy	0.747	0.776	0.788	0.0105
AUC	0.822	0.849	0.862	0.0101
Sensitivity	0.706	0.725	0.747	0.0137
Specificity	0.784	0.815	0.827	0.0130

Table 3 Cross Validation results for Seasonal Vaccine Predictions

Hyperparameter Tuning

- The process of using cross validation to find the optimal set of hyperparameter values for a model.
- Lays the **groundwork for model's structure, performance and training efficiency**
- Balance **variance-bias tradeoff**
- Improve **different Hyperparameters for each model**
- Aim to **minimize the loss function** of the model

  -> train its performance to be as **accurate as possible**

Hyperparameter Tuning - Random Grid

- Generate **random combinations** of hyperparameter values
- With a grid size of **500**
- Random sampling covers a wider range of values than **systematic selection**
- **Increases the chance** of finding optimal hyperparameter settings

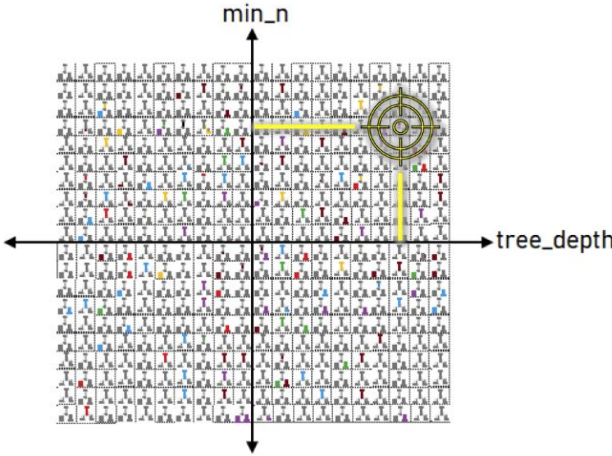


Fig. 19 Data Camp, Random Grid

Usage of mlr3 tuning spaces

- Ready-to-use search configurations for top Machine Learning algorithms
- Based on peer-reviewed research for broad dataset applicability
- Seamlessly integrates with the tidymodels framework for automated tuning

often called **λ** (lambda), the overall penalty strength

the mixing parameter between Ridge (α = 0) and LASSO (α = 1)

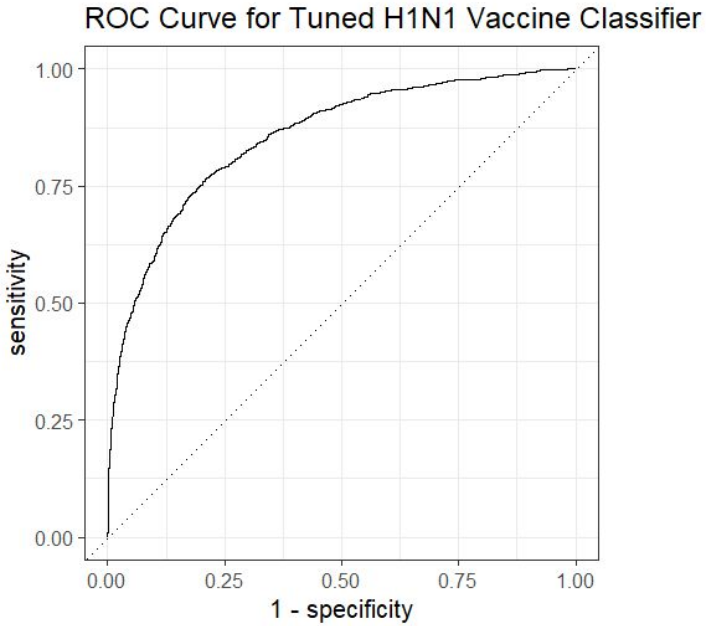
Description

Tuning spaces from the Bischl (2023) article:

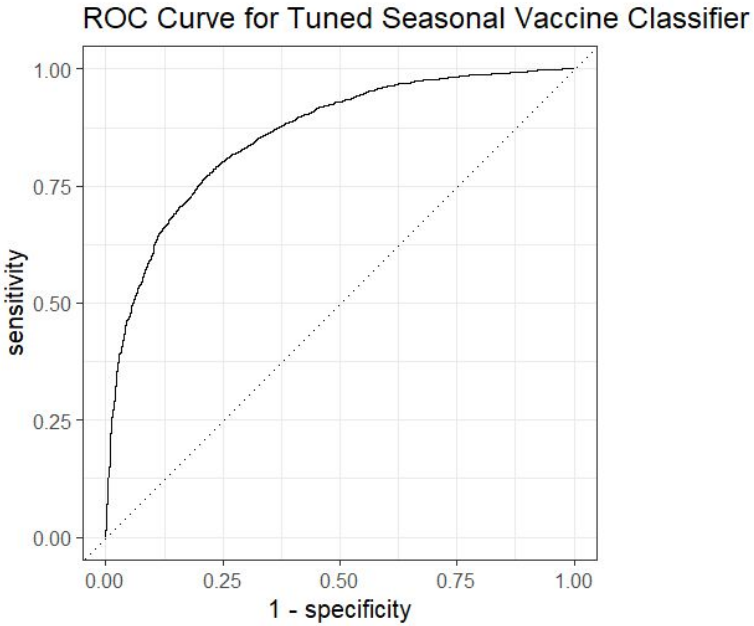
Glmnet tuning space

- s [1e - 04, 10000] Log-scale
- Alpha [0, 1]

Plots of post hypertuned ROC



AUC: 0.854



AUC: 0.859

Fig. 20 ROC curves of Logistic Regression Model post-tuning

Final Steps

- Select the best model **using ROC AUC**
- Perform a **last fit on held out splits**
- Evaluate model using **different metrics and visualize ROC curve**
- **Train model on the full training data and make predictions on the test data**
- Score and rank for Logistic Regression Model

Score:	Current Rank:
0.8542	836

Different Approaches

Random Forest

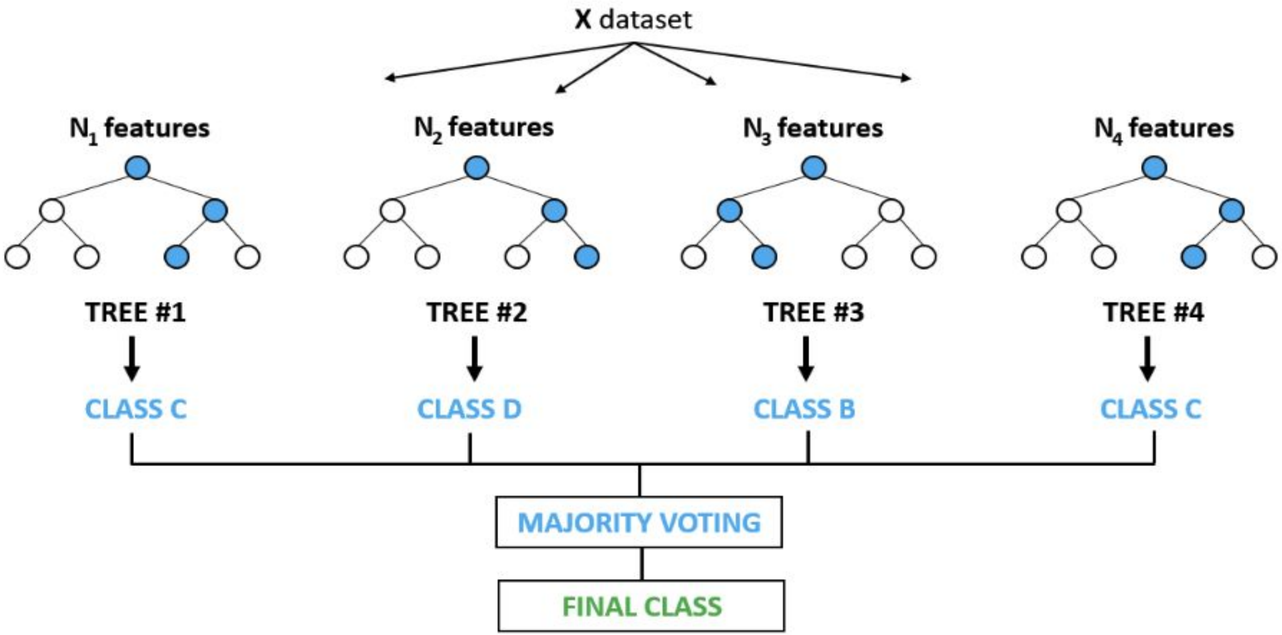


Fig. 21 Data Camp, Random Forest workflow

Random Forest - Feature Engineering

- Used **ranger**
- Dropped one-hot encoding
- Mode imputation for categorical and median imputation for numerical
- Retained our engineered interactions
- Wrapped it into workflows
- Unlike penalized logistic regression models, random forest models do not require **dummy** or **normalized** predictor variables.

Random Forest - Hyperparameter tuning

- mtry
- min_n
- tree

Ranger Tuning Space (mlr3)

- mtry.ratio [0, 1]
- sample.fraction [0.1, 1]
- num.trees [1, 2000]

LightGBM

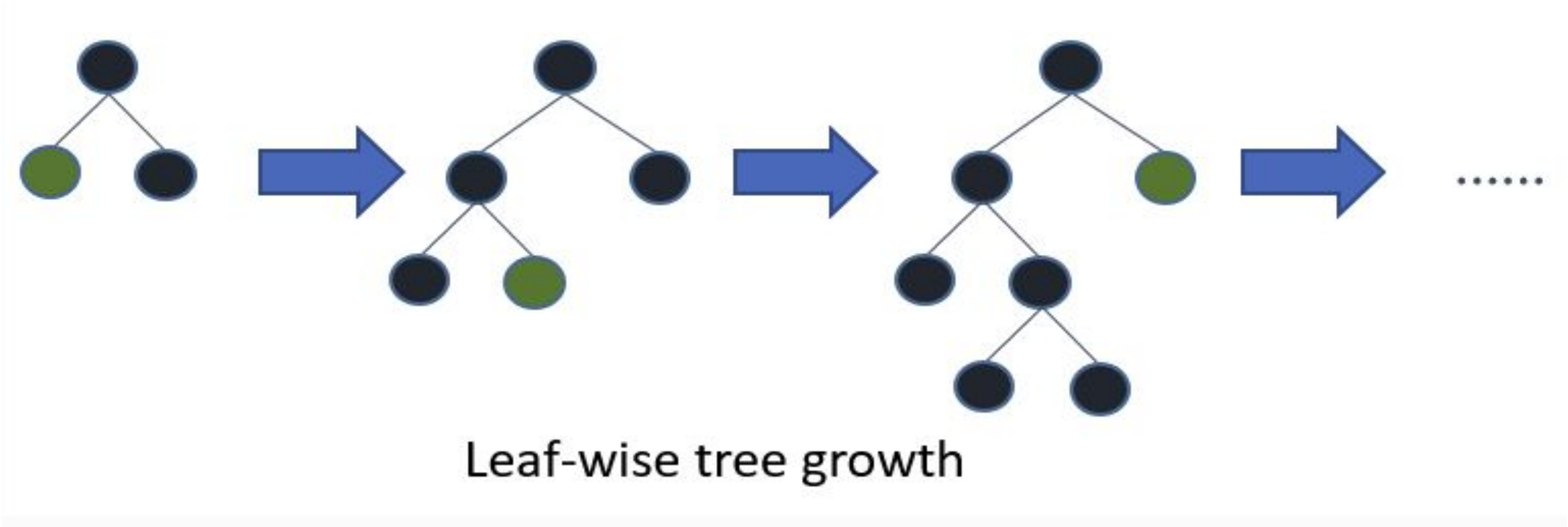


Fig. 22 LightGBM Documentation, LightGBM workflow

LightGBM - Feature Engineering

- Used one-hot encoding
- Median imputation
- Unknown-level handling
- Interaction engineering
- Removes zero-variance

LightGBM - Hyperparameter Tuning

- Number of trees
- Tree depth
- Learn rate
- Sample size
- Switched to **finetune**’s racing method

LightGBM performs best out of all models

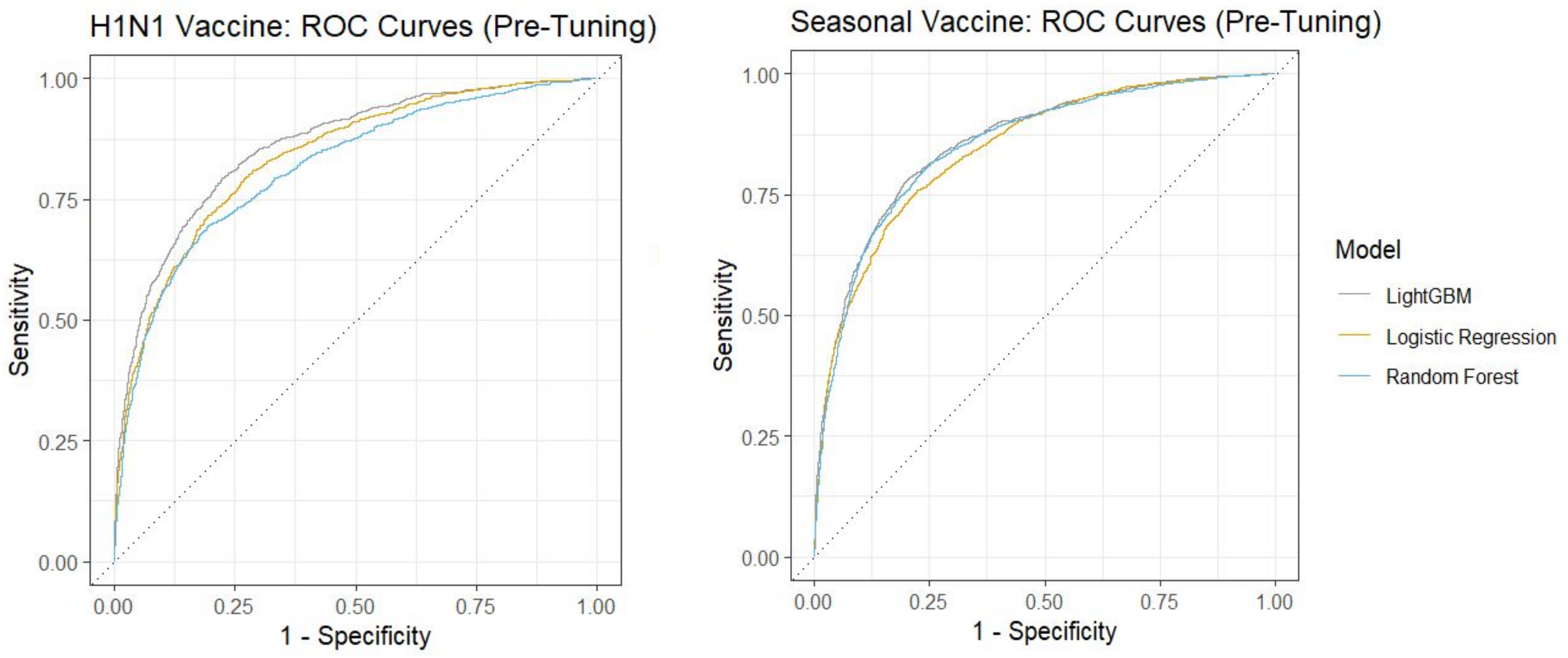


Fig. 23 Comparison ROC curves pre-tuning

LightGBM performs best out of all models

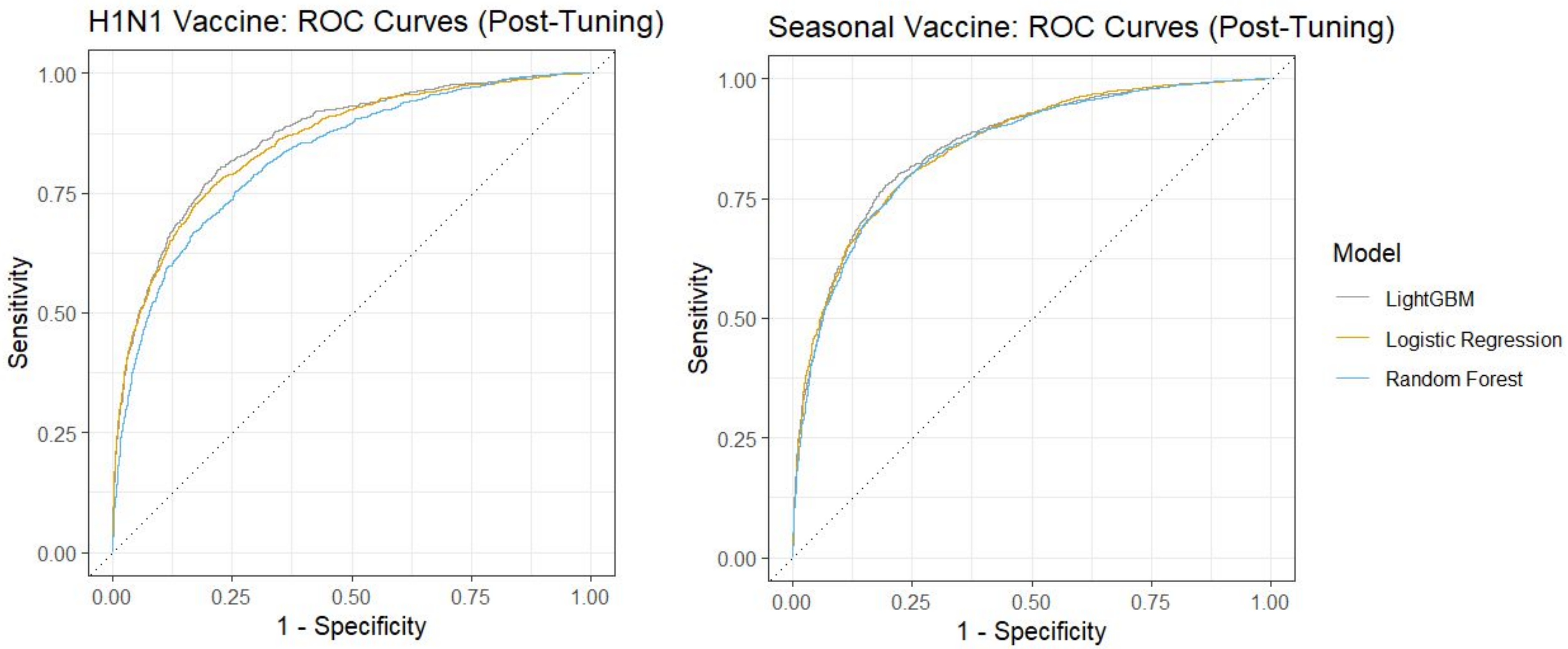


Fig. 24 Comparison ROC curves post-tuning

Leaderboard Score and Rank Comparison

Model	Score	Rank
Logistic Regression	0.8542	836
Random Forest	0.8392	-
LightGBM	0.8625	230

Table 4 Public Scores and Ranks

Best Score and Final Rank

Score:	Current Rank:
0.8625	230

-> LightGBM performed best out of all models we fitted
-> Top 2.8% of participants

Future Ideas

- For future work, we recommend Model Ensembling and Stacking. By training multiple models, we can build a meta-model using, i.e. CatBoost and LightGBM and combine their predictions.

- Another approach would be to use Voting/Averaging. Although simple, this can be effective by simply averaging predictions from multiple strong models.

Resources

- DrivenData, Flu Shot Learning - Predict H1N1 and Seasonal Flu Vaccines, https://www.drivendata.org/competitions/66/flu-shot-learning/page/213/
- R. Elliot, What is Random Digit Dialing?, GeoPoll, https://www.geopoll.com/blog/what-is-random-digit-dialing/
- Google Developer's, Datasets: Imbalanced datasets, https://developers.google.com/machine-learning/crash-course/overfitting/imbalanced-dataset
- Z. Bobbitt, Phi Coefficient: Definition and Examples, Stratology, https://www.statology.org/phi-coefficient/
- A. Heiss, Little's missing completely at random (MCAR) test, R Documentation, https://search.r-project.org/CRAN/refmans/naniar/html/mcar_test.html
- M. C. M. de Goeij, M. van Diepen, K. J. Jager, G. Tripepi, C. Zoccali, F. W. Dekker, Multiple imputation: dealing with missing data, Nephrology Dialysis Transplantation, Volume 28, Issue 10
- F. Lee, What is logistic regression?, IBM, https://www.ibm.com/think/topics/logistic-regression
- A. Chopra, Stratified Split: Why Data Splitting is crucial in Machine Learning, LinkedIn, https://www.linkedin.com/pulse/ml-6-stratified-split-why-data-splitting-crucial-machine-chopra-zdv
- M. Kuhn, K. Johnson, Feature Engineering and Selection: A Practical Approach for Predictive Models, https://bookdown.org/max/FES/
- A. Rojo-Echeburúa, What Is One Hot Encoding and How to Implement It in Python, DataCamp, https://www.datacamp.com/tutorial/one-hot-encoding-python-tutorial
- scikit-learn, Cross-validation: evaluating estimator performance, https://scikit-learn.org/stable/modules/cross_validation.html
- scikit-learn, Tuning the hyper-parameters of an estimator, https://scikit-learn.org/stable/modules/grid_search.html
- M. Becker, mlr3tuningspaces: Search Spaces for 'mlr3', R package version 0.6.0, https://mlr3tuningspaces.mlr-org.com
- scikit-learn, confuision_matrix, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html
- Google Developer's, Classification: ROC and AUC, https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Resources

- https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
- https://www.ibm.com/think/topics/logistic-regression
- https://www.ibm.com/think/topics/random-forest
- https://www.drivendata.org/competitions/66/flu-shot-learning/page/213/
- https://www.tmwr.org/recipes
- https://www.tidymodels.org/find/parsnip/
- https://ruslanmv.com/blog/The-best-binary-Machine-Learning-Model
- https://christophm.github.io/interpretable-ml-book/logistic.html
- https://christophm.github.io/interpretable-ml-book/tree.html
- https://medium.com/@data-overload/comparing-xgboost-and-lightgbm-a-comprehensive-analysis-9b80b7b0079b
- https://lightgbm.readthedocs.io/en/latest/Advanced-Topics.html

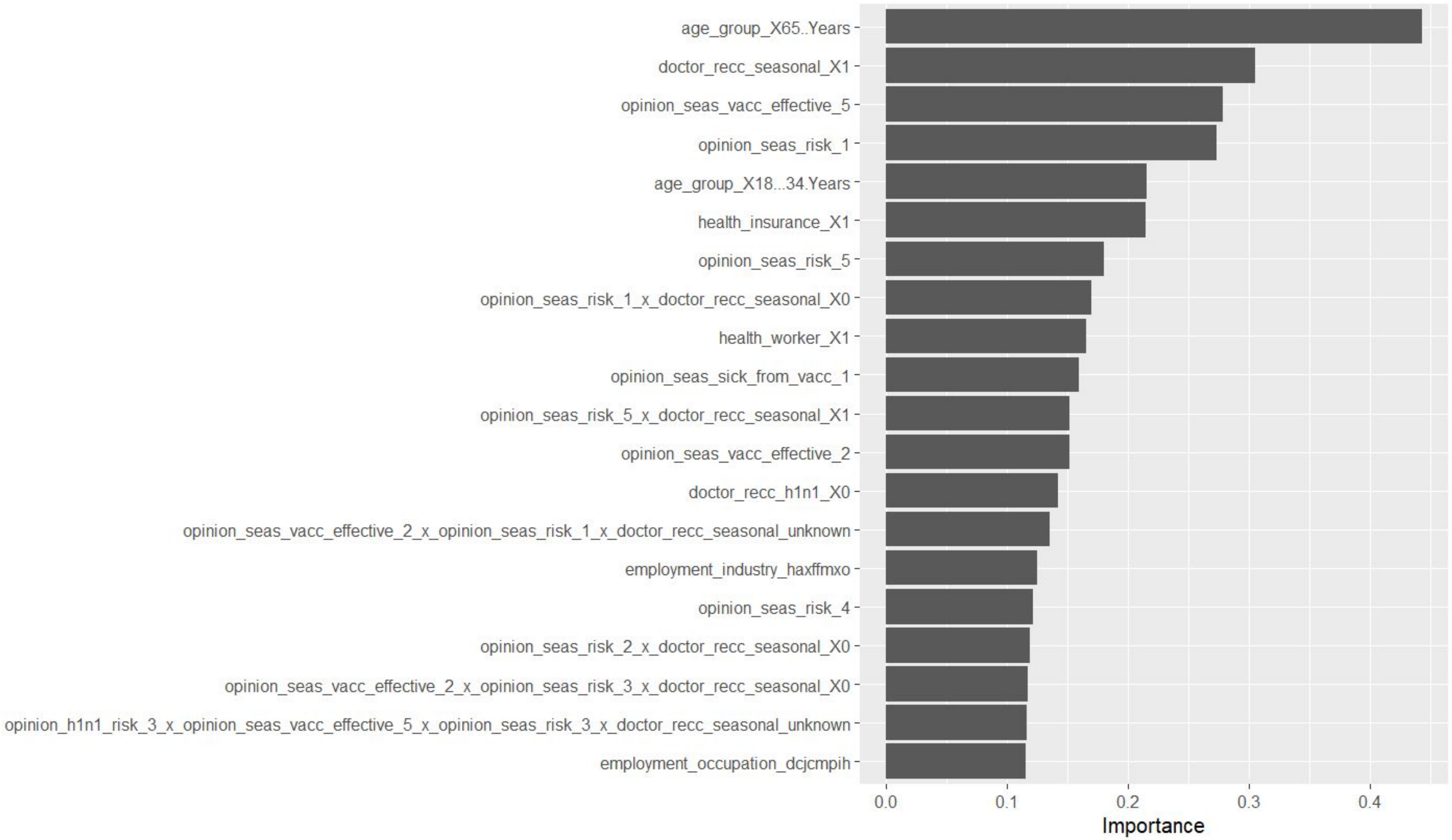
Discussion

ROC values

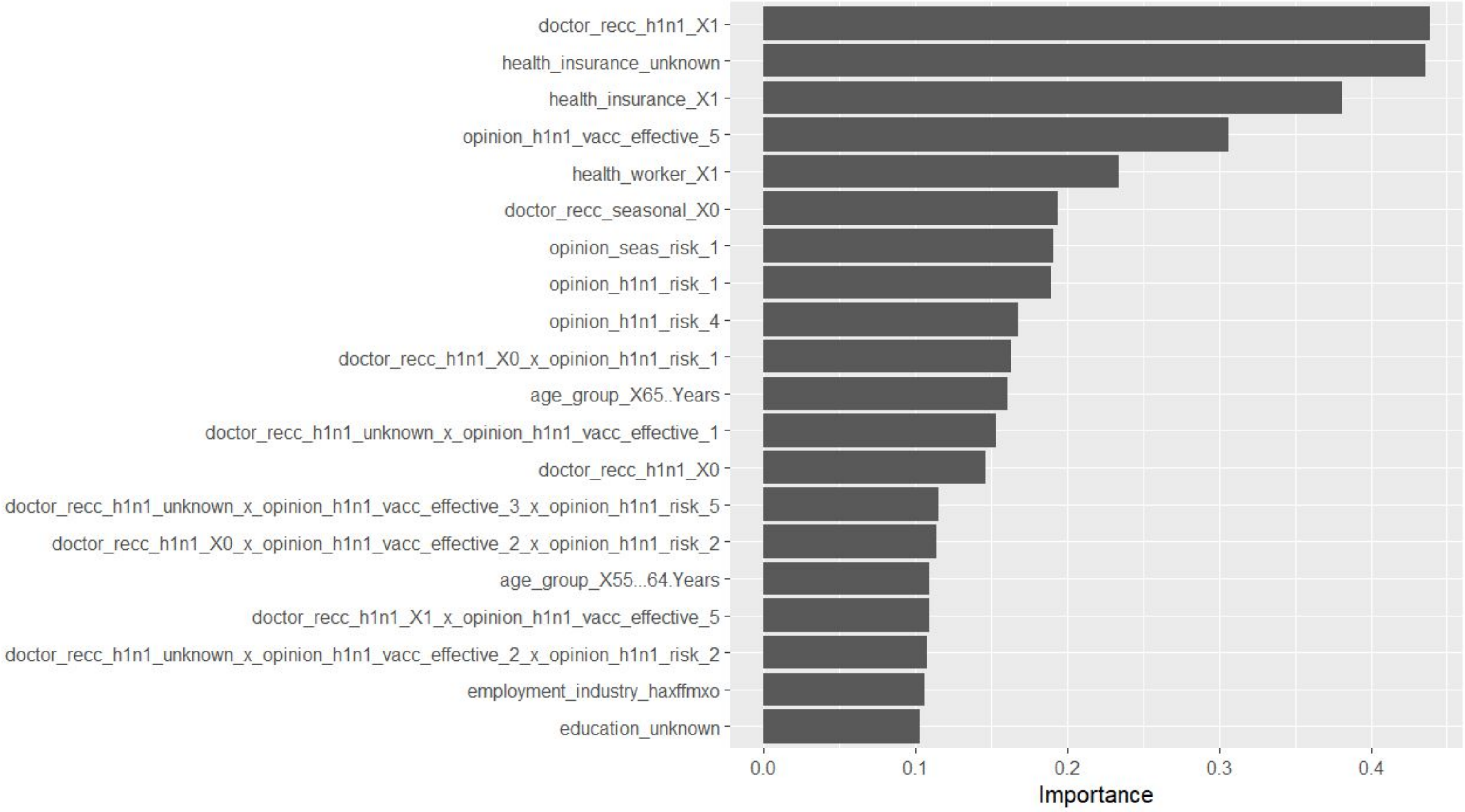
	Pre-tuning			Post-tuning		
Models	Logistic Regression	Random Forest	LightGBM	Logistic Regression	Random Forest	LightGBM
H1N1	0.839	0.826	0.861	0.854	0.828	0.864
Seasonal	0.846	0.853	0.860	0.859	0.854	0.862

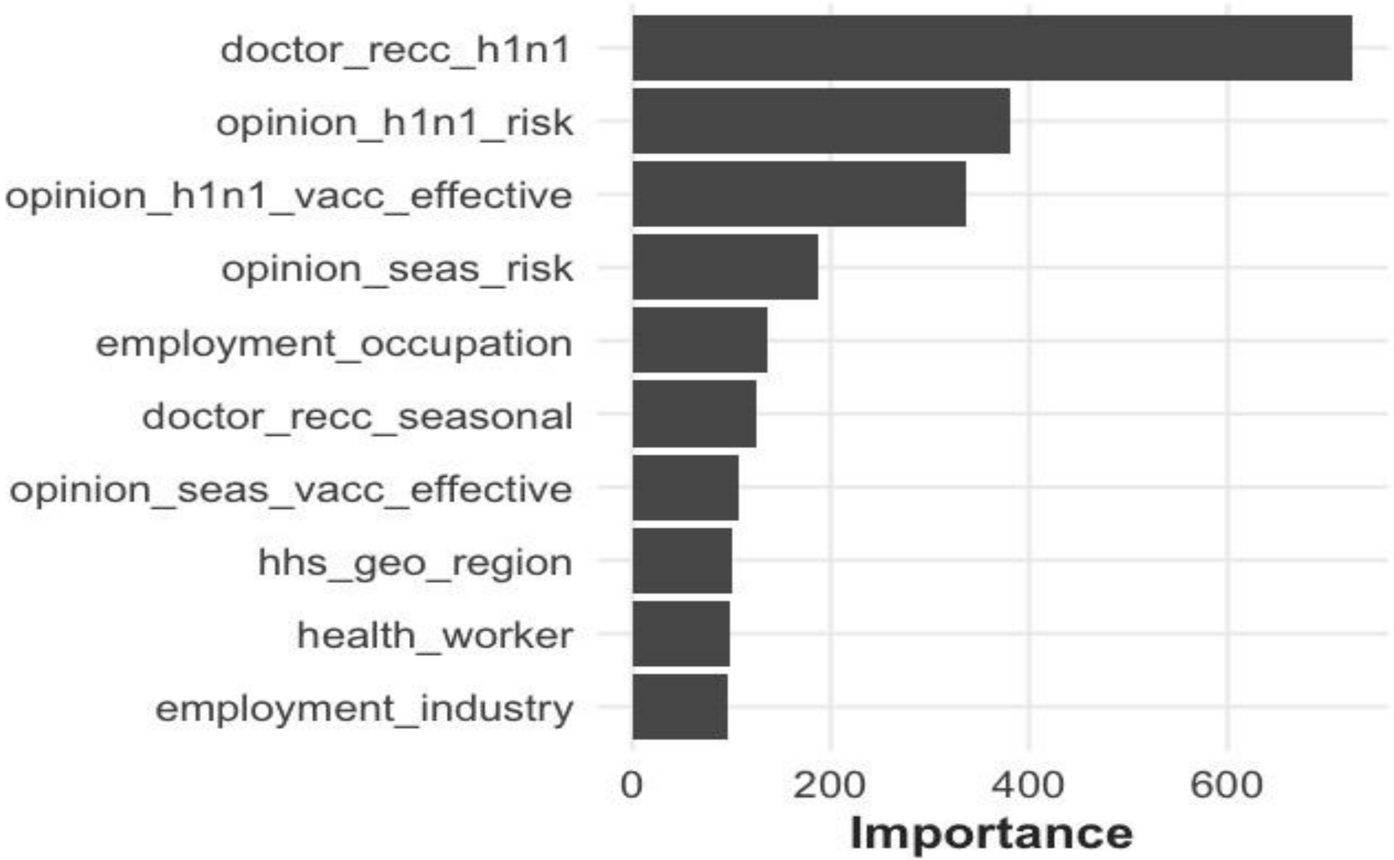
Table 5 Comparison results of pre-tuning and post-tuning

Feature Importance for H1N1 vaccine using Logistic Regression post-tuning

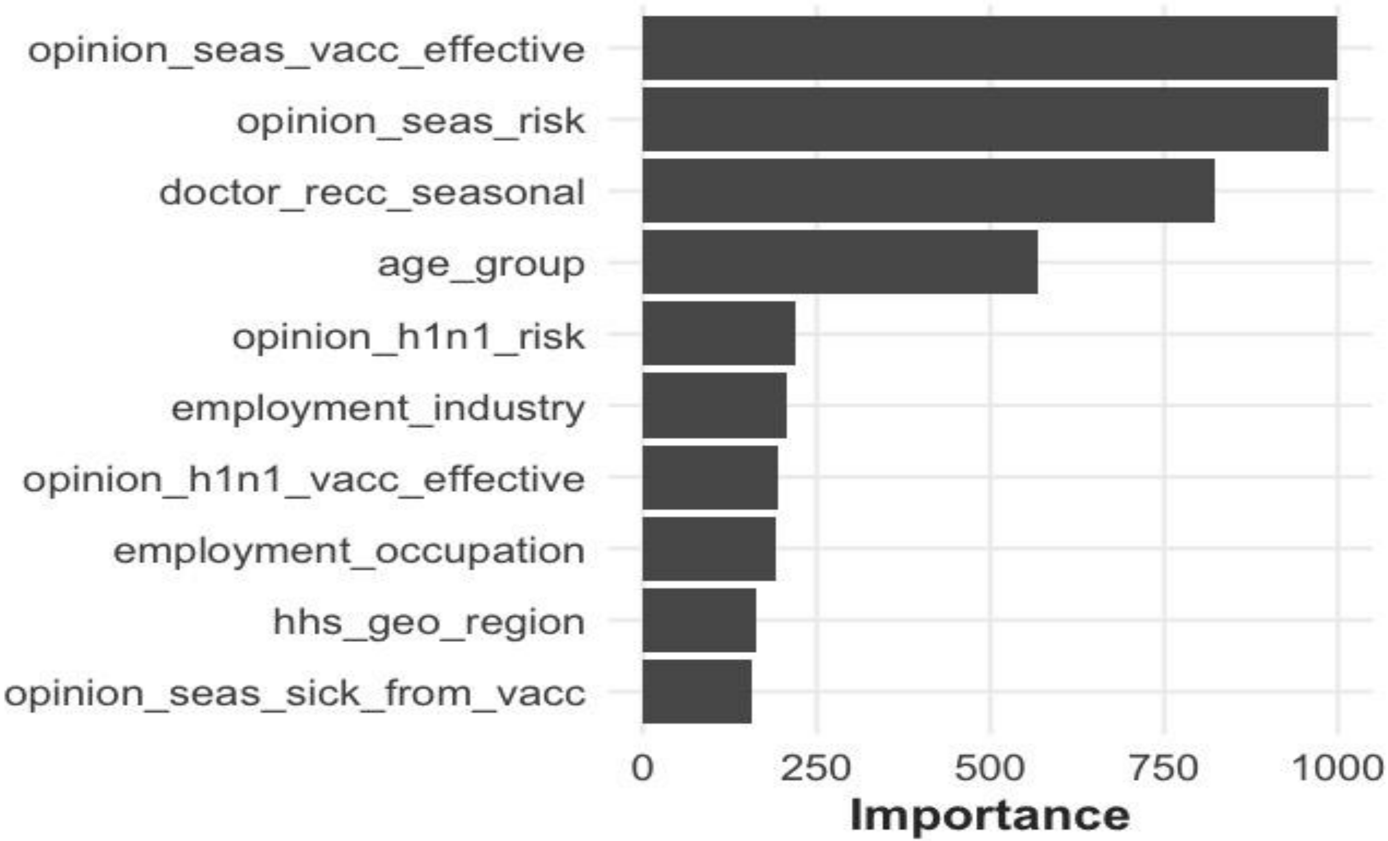


Feature Importance for Seasonal vaccine using Logistic Regression post-tuning

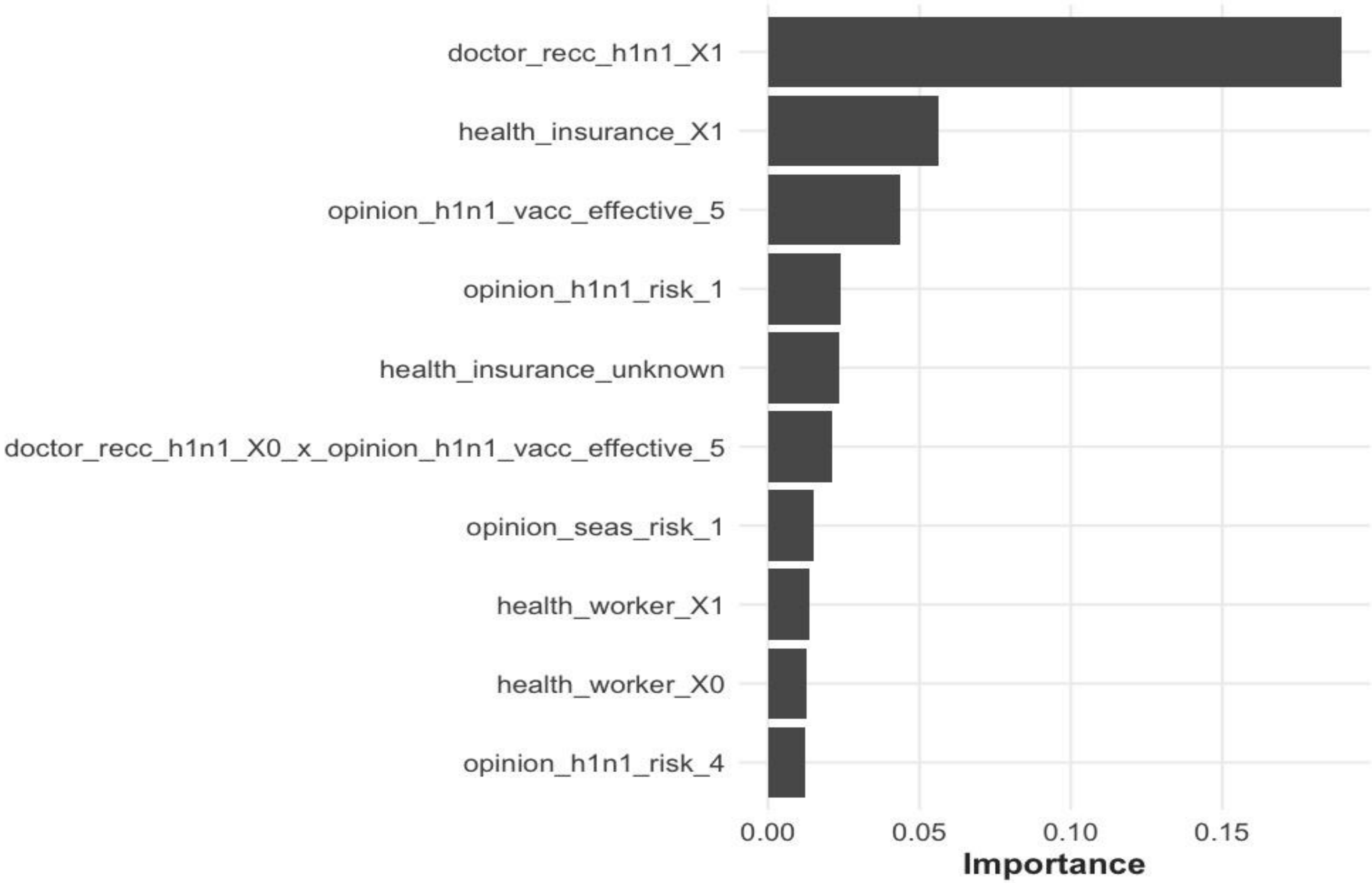



doctor_recc_h1n1
opinion_h1n1_risk
opinion_h1n1_vacc_effective
opinion_seas_risk
employment_occupation
doctor_recc_seasonal
opinion_seas_vacc_effective
hhs_geo_region
health_worker
employment_industry
0
200
400
600
Importance

Feature Importance for Seasonal vaccine using Random Forest post-tuning



Feature Importance for H1N1 vaccine using LightGBM post-tuning



Feature Importance for H1N1 vaccine using LightGBM post-tuning