Stroke Prediction Modeling

Springboard Capstone 2

Problem Statement

Strokes are the 2nd leading cause of death globally; account for 11% of total deaths (per WHO)

Using machine learning or AI could be indispensable for predicting strokes

We want to answer:

- Can we predict strokes effectively from patient's physical and personal attributes?
- What attributes are most important to model prediction?
- What algorithms will be most effective?

The Data

Stroke Prediction Dataset, is publicly available on Kaggle

The data contains 11 describing attributes for patients

Categorical	Numerical
gender	age
hypertension	avg glucose level
heart disease	bmi
ever married	
work type	
residence type	
smoking status	

Response variable: stroke

Class Imbalance

The data contains 5110 observations (patients)

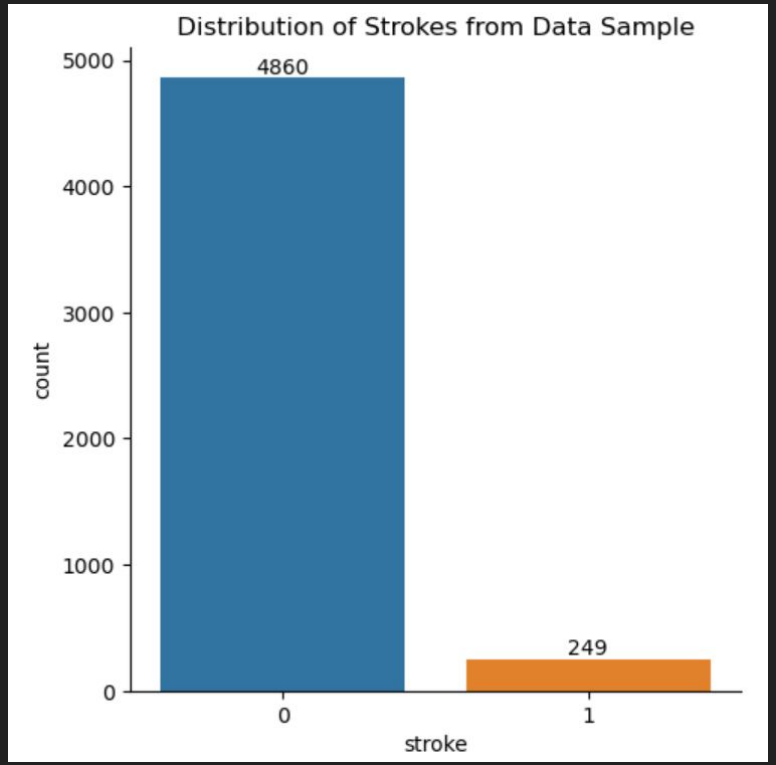
Only 249 patients had a stroke

- <5% stroke-rate

Class imbalance creates a challenge for machine learning algorithms

Accuracy will not be the best metric to evaluate model performance

Class Imbalance



Data Cleaning

Identify outliers

Remove singularities (e.g. ‘Other’ in gender column)

Replace NaN values in bmi column with median

- Appeared to be normally distributed

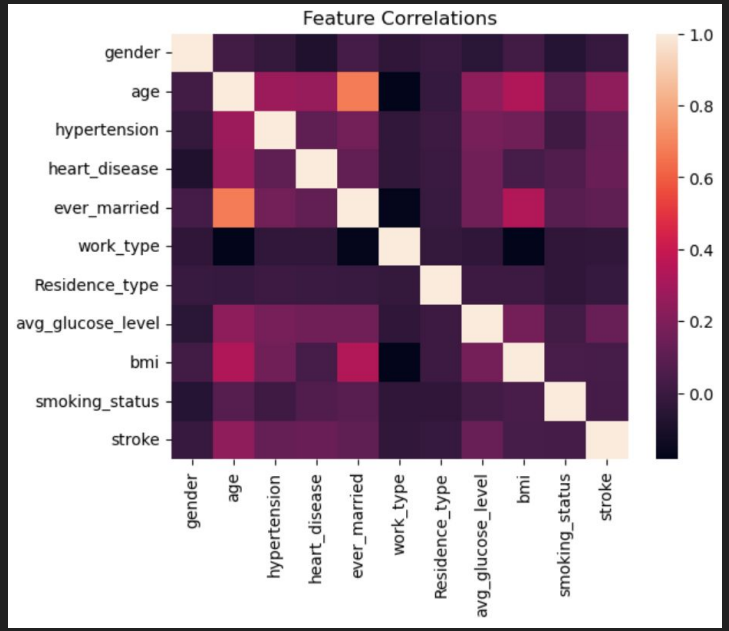
Encode the data

- One-hot encoding for binary features
- Dummy encoding for categorical features

Exploratory Data Analysis

Investigate the correlations between features

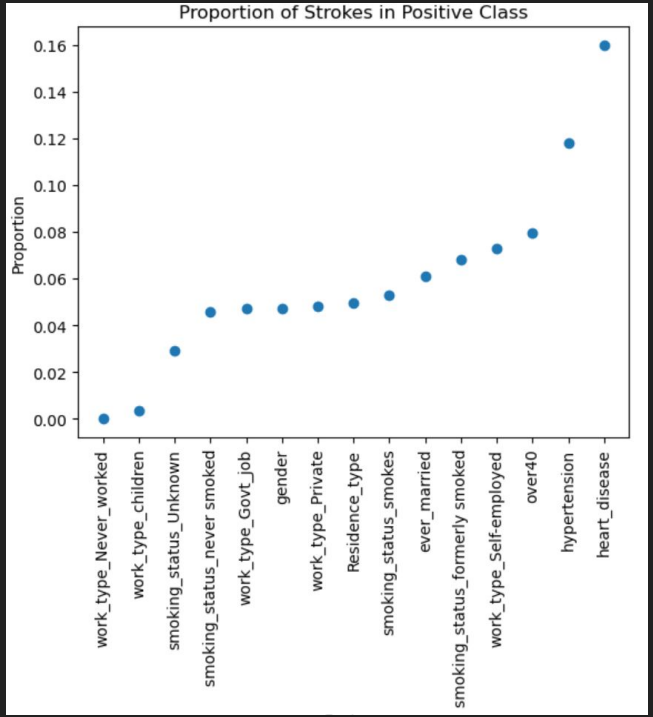
Features are not strongly correlated with strokes



Exploratory Data Analysis

Investigate the proportion of strokes per class

Will these features be most important for our models?



Data Preprocessing

Split the data into testing and training sets using 80:20 split

- X_train, X_test, y_train, y_test

Training set includes 4086 observations

Testing set includes 1022 observations

Modeling

Given the class-imbalance our metric choices are important

- Area under the curve (AUC)
- F1-score
- F1.5-score: higher emphasis on recall; minimize false-negatives

The algorithms we trained and tested:

- Random Forest Classifier
- Support Vector Machine (SVM)
- Logistic Regression
- XGBoost Classifier
- CatBoost Classifier

Modeling

Parameter tuning using grid search cross-validation and scoring by recall

Initial models had poor performance and almost all had 0 recall

Use synthetic minority oversampling technique (SMOTE) to upsample data to remove class-imbalance from the training set

Train and test algorithms again…

Results

	Model Name	AUC	F1 Score	F1.5 Score
0	RandomForest	0.823824	0.000000	0.000000
1	SVM	0.583401	0.000000	0.000000
2	LogisticRegression	0.864012	0.000000	0.000000
3	XGBoost	0.667490	0.098765	0.082019
4	CatBoost	0.829965	0.000000	0.000000
5	RandomForest_scaled	0.818893	0.120482	0.101246
6	SVM_scaled	0.661744	0.164948	0.148997
7	LogisticRegression_scaled	0.810333	0.201439	0.210162
8	XGBoost_scaled	0.732745	0.025316	0.020767
9	CatBoost_scaled	0.831048	0.141414	0.128895

Best models: Logistic Regression, SVM, and CatBoost

Results

Identify key features in top models

- never smoked
- currently smokes
- formerly smokes
- unknown smoking status
- private work type
- age

Recall that most of these features did not have the highest proportions of strokes

Results

Perform threshold tuning on top models

By default, classification models have a threshold of 0.5 to predict labels of probability predictions

For data with class-imbalance, 0.5 may not be suitable

Tested values between (0,1) and optimized for both F1-score and F1.5-score

Results

Threshold tuning is a key parameter for this data

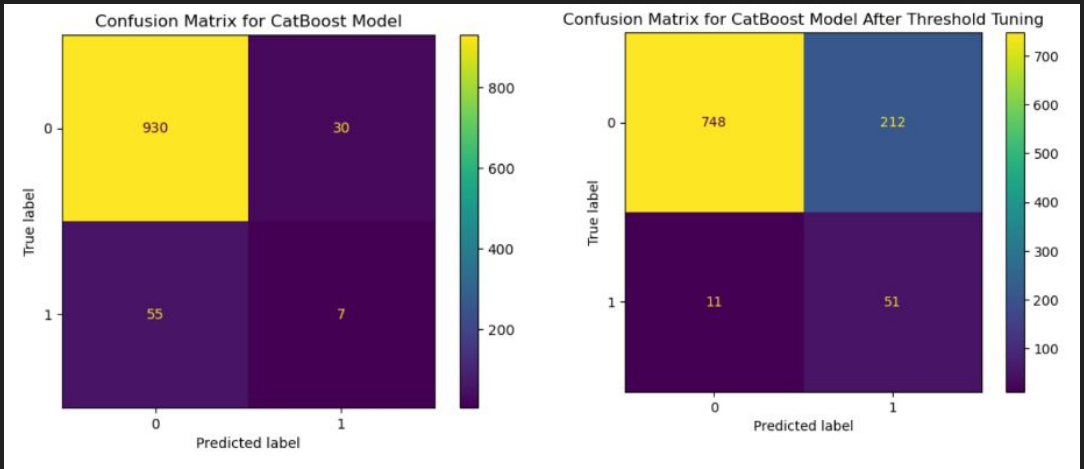
Revealed CatBoost to be most effective

- Best model was chosen with F1.5-score threshold

	Model	F1-Score	F1.5-Score	AUC (w/ F1 Threshold)	AUC (w/ F1.5 Threshold)	F1 Threshold	F1.5 Threshold
0	SVM	0.243137	0.303008	0.665625	0.665625	0.010	0.010
1	Logistic Regression	0.313167	0.398884	0.763693	0.763693	0.266	0.266
2	CatBoost	0.319728	0.411801	0.782678	0.800874	0.138	0.122

Results

Confusion matrices for CatBoost models before and after threshold tuning:



This parameter greatly improved the recall of the model

Conclusion and Future Work

We tested a variety of ML models, identified the most important features to predicted strokes, and optimized our top models with parameter tuning

Future work:

- Refine models with additional observations
- Include additional features
- Try more algorithms (e.g. deep learning)