Spring 2020 – Final Project
Predicting Airline Delay

Team 13:
Thomas Goter
Douglas Xu
Marcial Nava
Hong Yang

Question Formulation

- Delays cost the economy more than $33 billion per year, from which half is borne by the customers*
- Nearly 20% of domestic flights experienced a delay every year (Source: Bureau of Transportation Statistics)
- About 25-50% of delays are caused by weather (Source: Bureau of Transportation Statistics)
- Can we predict with a reasonable degree of certainty whether a flight will be delayed at least one hour before departure, and for how long?

*Sources: Michael Ball, Cynthia Barnhart, Martin Dresner, *et.al.* (2010). "Total delay impact study: a comprehensive assessment of the costs and impacts of flight delay in the United States." NEXTOR. October. Available at: https://rosap.ntl.bts.gov/view/dot/6234

EDA & Discussion of Challenges

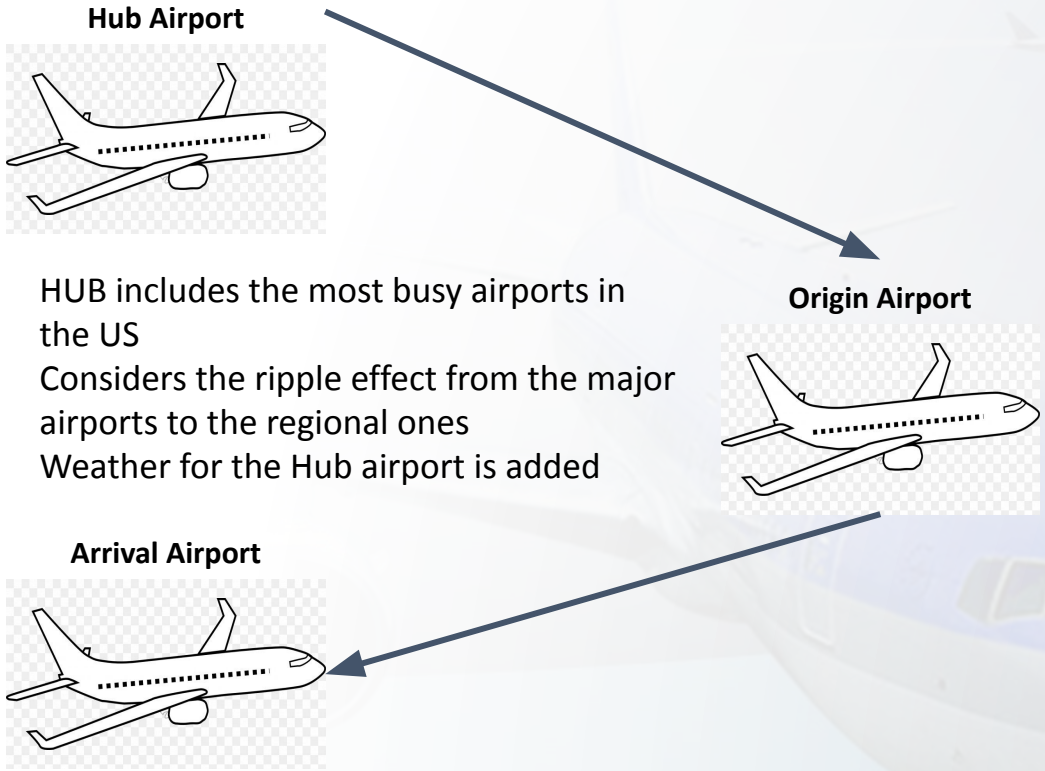
	Examples	
EDA	**Airlines**	**Weather**
Dimensionality	Benefit from most observations	More observations than what was actually needed, significant degree of sparcity
Features	Timestamps could be better treated as categorical variables	Schemas are not consistent across files. More than 170 potential fields. Features measured in different scales
Target variable	Continuous, not very practical for our purposes. Skewed	
Relationship between target variable and features	Seasonality and fixed effects , multicolinearity	Difficult to assess due to the presence of extreme observations, multicolinearity

EDA & Discussion of Challenges

EDA Challenges	
Dimensionality reduction	Discriminate among features, filter only the observations that we need (airports)
Sparsity	Decide between replacing null observations or leaving them as they are
Outliers	Eliminate them or transform the target variable to make them irrelevant
Transformations	How to make our features more suitable to our models
New features	New features were created to boost predictive power (e.g. origin, hub, destination)
Merging the two datasets	Merge the two datasets based on hour and location
Balance	Much of the data is labeled as no delay, thus, subsampling, resampling or weights?

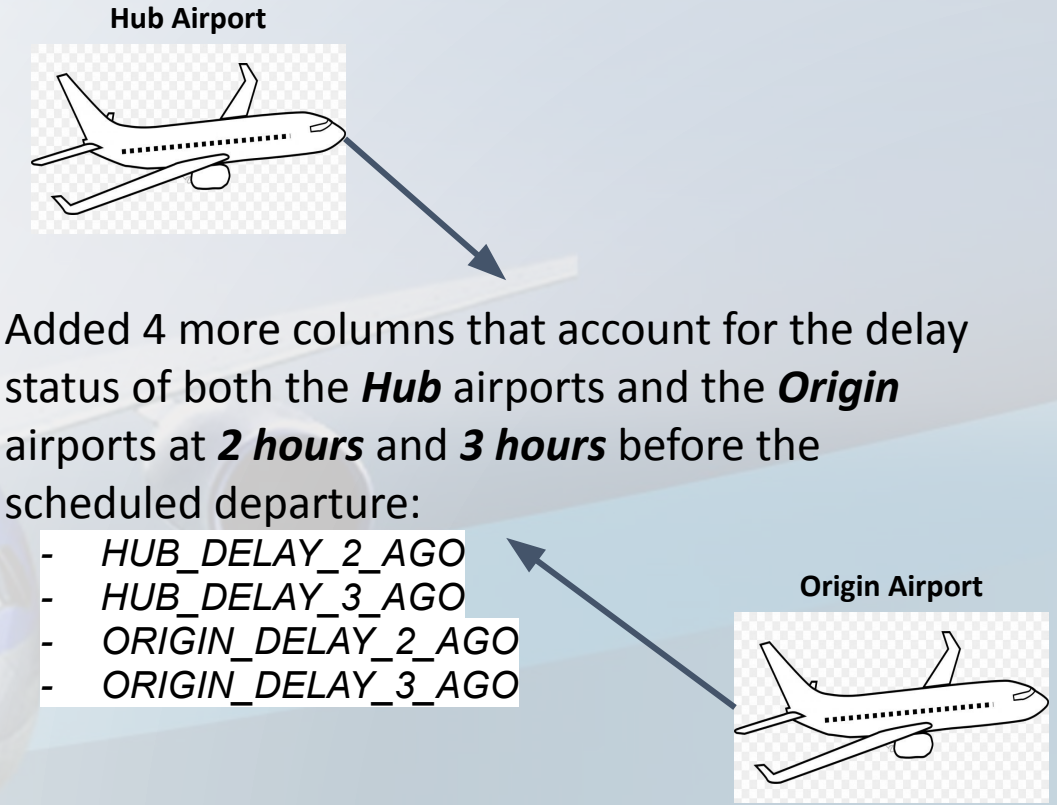
Highlights of Feature Engineering

Delay due to delayed previous flights - HUB



- HUB includes the most busy airports in the US
- Considers the ripple effect from the major airports to the regional ones
- Weather for the Hub airport is added

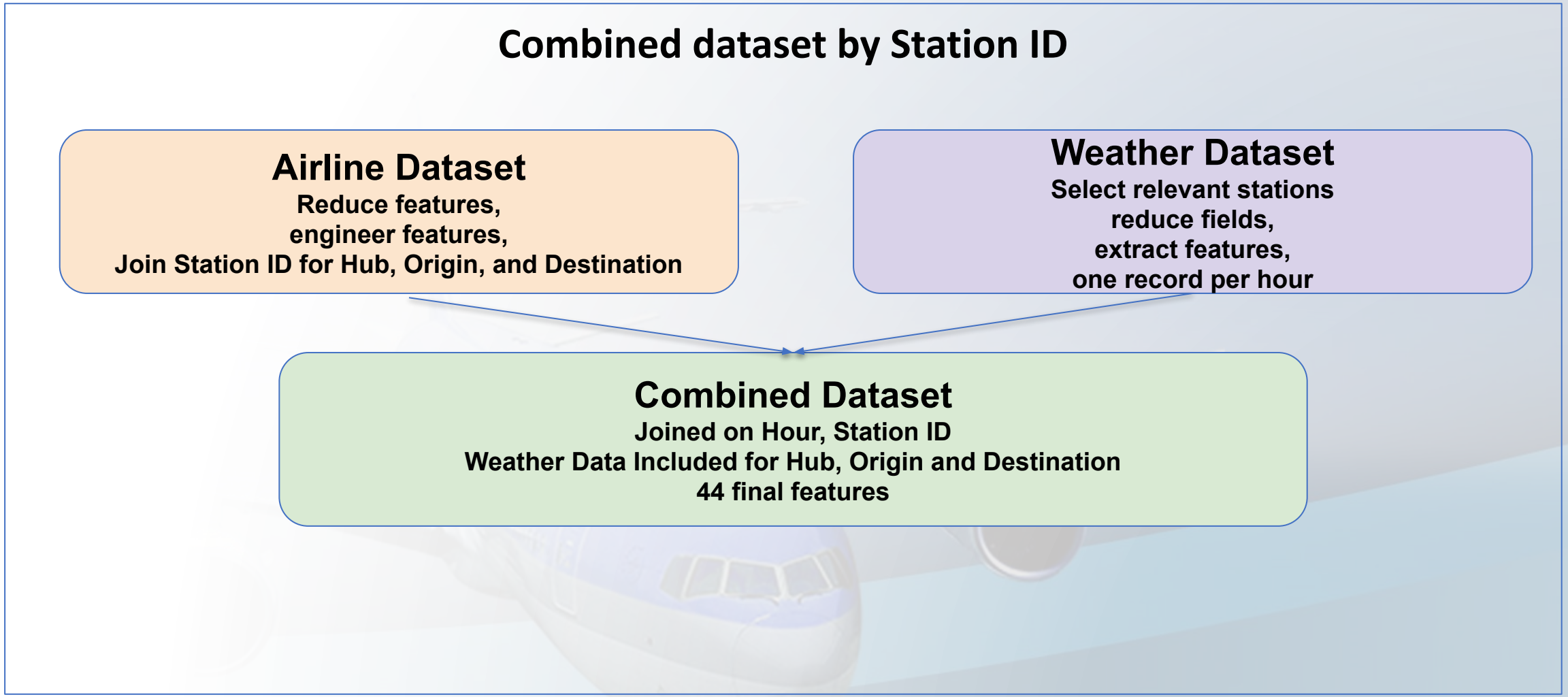
Considering the delay pattern throughout the day



Added 4 more columns that account for the delay status of both the ***Hub*** airports and the ***Origin*** airports at ***2 hours*** and ***3 hours*** before the scheduled departure:

- *HUB_DELAY_2_AGO*
- *HUB_DELAY_3_AGO*
- *ORIGIN_DELAY_2_AGO*
- *ORIGIN_DELAY_3_AGO*

Highlights of Feature Engineering cont.



Feature Engineering - Cont.

- **Treating missing value**
  - Target column
  - Continuous variables
  - Categorical variables

- **Final feature transformation**
  - Bucketized based on hand selected cutoff
  - Applied one-hot encoding

Algorithm Exploration

Naive Bayes:

Pros:

1. fast implementation
2. easy to parallelize
3. fast in training

Cons:

1. assume feature independence while features might be correlated

Random Forest:

Pros:

1. minimum feature transformations
2. high potential for fine-tuning

Cons:

1. slower in training
2. PySpark ML does not allow weights to adjust for class imbalance

Logistic Regression:

Pros:

1. considers weights to adjust class imbalance
2. efficient

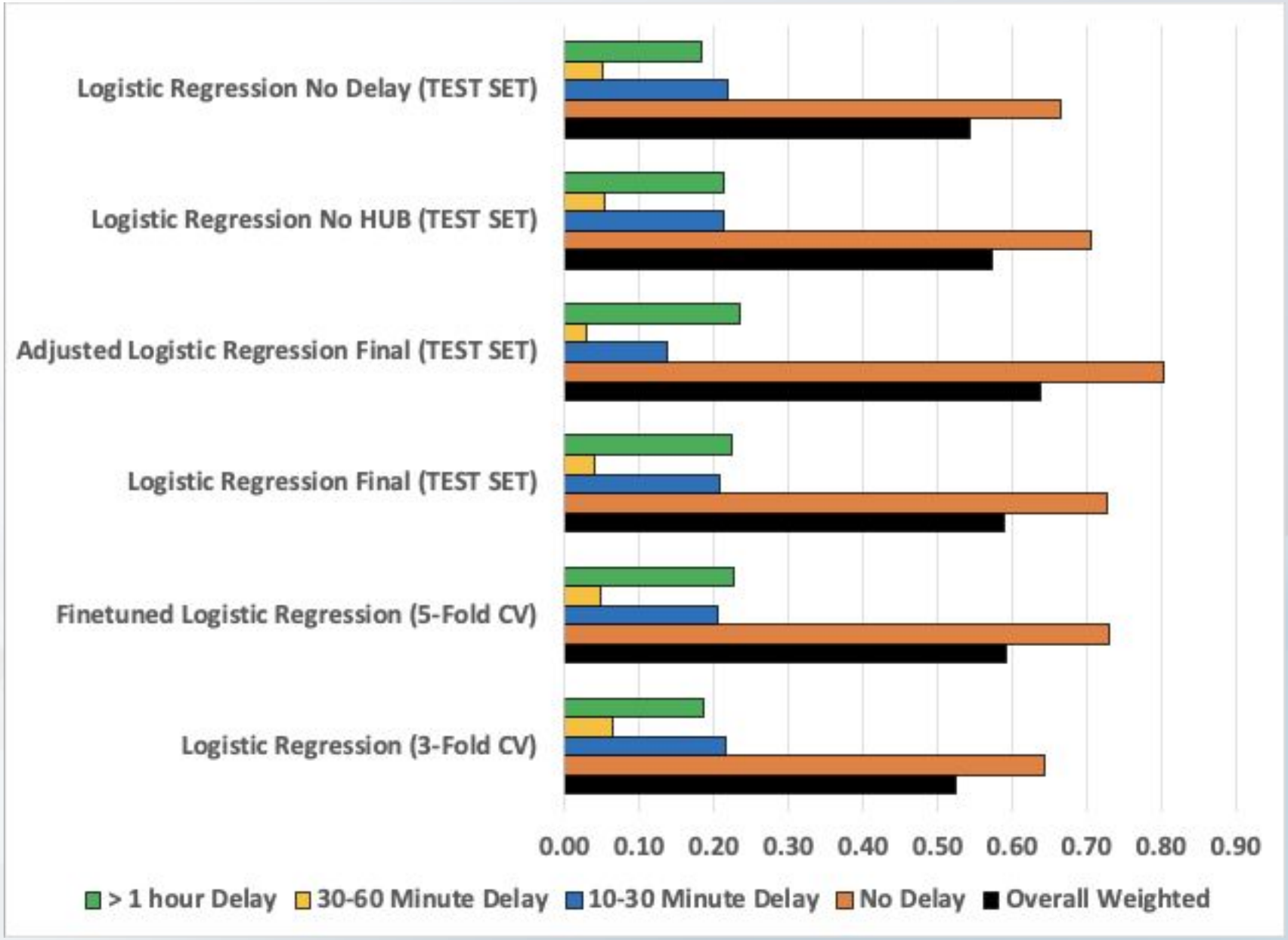
Cons:

1. not as sophisticated as random forest in complex and highly non-linear problems

	Naive Bayes	Random Forest	Logistic Regression
Weighted F1 Score	0.510	0.556	0.525
F1 label 0 [0-10 mins]	0.620	0.683	0.642
F1 label 1 [10-30 mins]	0.210	0.219	0.217
F1 label 2 [>60 mins]	0.186	0.193	0.187
F1 label 3 [30 - 60 mins]	0.086	0.064	0.065
Time to train (1 model w/o CV)	2 mins	10-20 mins	2 mins

Final Results

F1 Score - Weighted and by Class



Conclusions

Lesson & Learns:

01	Regression or Classification	• depends on the dataset • depends on the problem domain
02	Null handling for continuous variables	• Method 1: replace Null with mean • Method 2: bucketize variable into bins, Null will be its own bin
03	Feature engineering can contribute significantly to accuracy	• We have seen the majority of performance gain from feature engineering
04	Adjusting for class imbalances	• Importance for having the feature to adjust for class imbalances

Future Work:

- Fine-tuning target variable bins
- More specific route details
- Explore other algorithms
- Incorporating explanatory variables other than weather (e.g. security, maintenance, plane characteristics, etc.)

Thank you!
Q&A

Back-ups

Algorithm Implementation

Toy Example

Observation	Label	ORD	TEMP	RAIN	Rain_StdScl	Comment
1	0	0	50	0	-0.127	No delay
2	0	0	70	0	0.889	No delay
3	0	1	75	0	1.143	No delay
4	1	0	45	1	-1.651	Delay, raining, colder
5	1	1	20	1	-0.381	Delay O'hare and raining, cold
6	1	1	55	1	0.127	Delay O'hare and Raining

Observation	True Label	x_1 (ORD)	x_2 (Std. Scld. Temp)	x_3 (RAIN)	β_1	β_2	β_3	z	$\hat{y}$	Predicted Label
1	0	0	-0.127	0	0.118	0.261	-0.927	1.162	0.529 > 0.5	DELAYED - **Incorrect**
2	0	0	0.889	0	-0.824	0.261	-0.927	1.162	0.305 < 0.5	ONTIME - **Correct**
3	0	1	1.143	0	-0.799	0.261	-0.927	1.162	0.310 < 0.5	ONTIME - **Correct**
4	1	0	-1.651	1	2.692	0.261	-0.927	1.162	0.937 > 0.5	DELAYED - **Correct**
5	1	1	-0.381	1	1.776	0.261	-0.927	1.162	0.855 > 0.5	DELAYED - **Correct**
6	1	1	0.127	1	1.212	0.261	-0.927	1.162	0.771 > 0.5	DELAYED - **Correct**

Implementation & Pipeline

Categorical Features

1. **String Indexer**: convert categorical classes into strings
2. **Vector Assembler**: vectorize the categorical feature rows
3. **One-Hot Encoder (optional)**: optional step to one-hot encode the categorical columns, depending on the algorithm

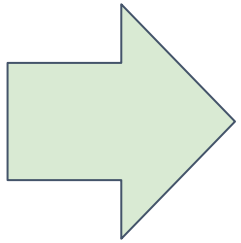
Continuous Features

1. **Imputer**: replace Null values with mean value of the column
2. **Vector Assembler**: vectorize the continuous feature rows
3. **Standard scaler (optional):** standardize the continuous feature columns

EDA & Discussion of Challenges

EDA Summary

1. Determining dimensions
2. Explore datasets separately
3. Evaluate the type of data available
4. Evaluate the target variable
5. Investigate distributions of potential features
6. Investigate relationships between target and predictive variables



Challenges

1. Dimensionality reduction
2. Missing values
3. Sparsity
4. Outliers
5. Additional transformations to the target variable
6. Cleaning and merging the two databases