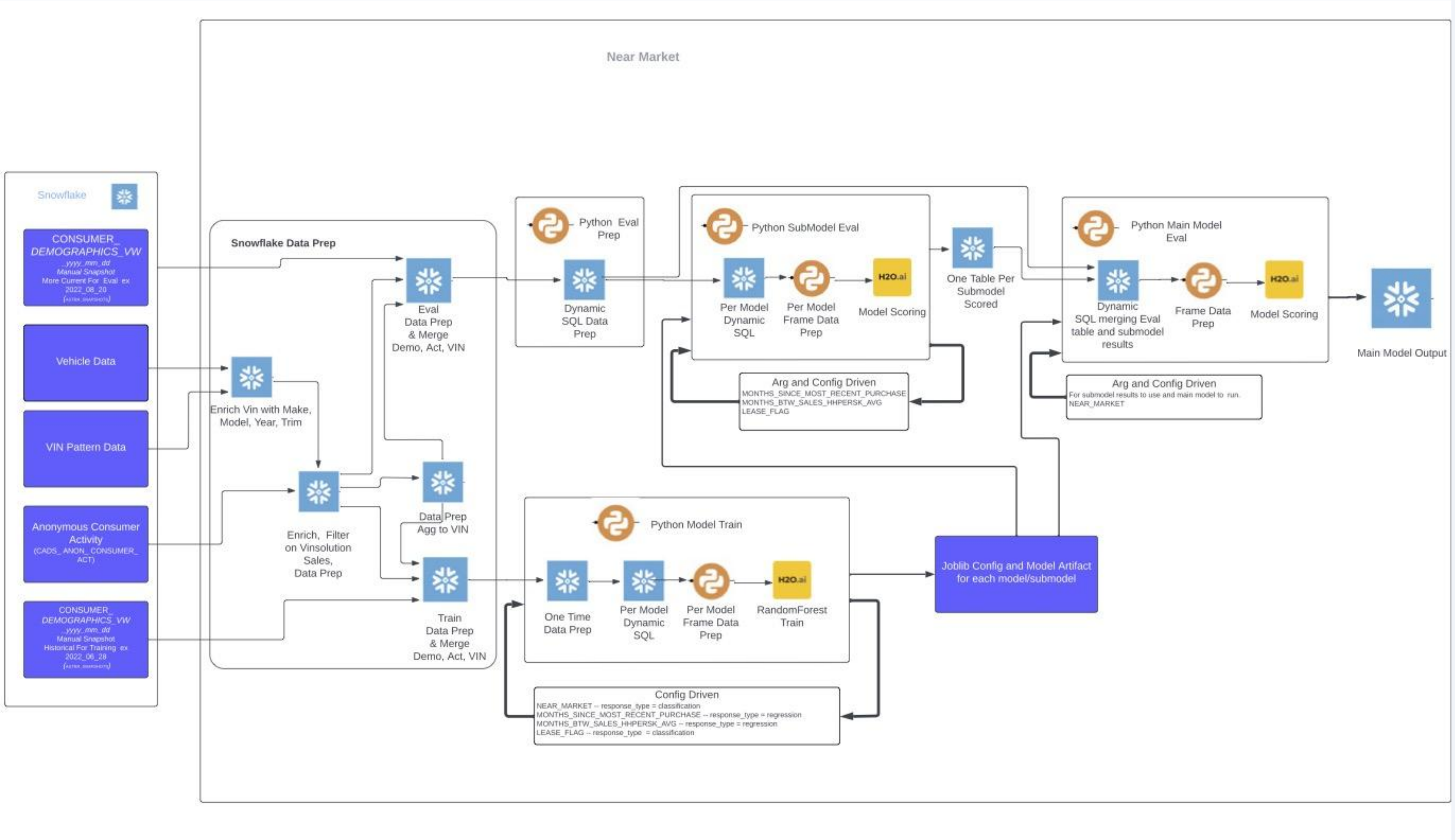
SUMMARY

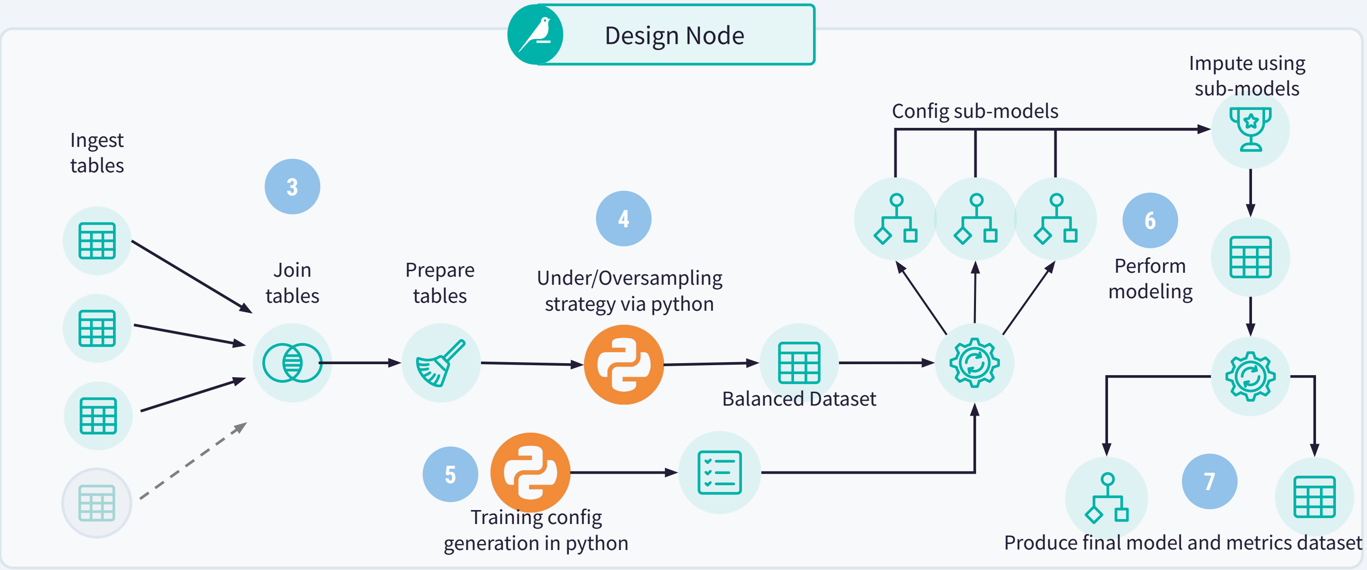
Context

Cox Automotive wants to create a targeted outreach program aimed at certain customers they've classified as "near market", or looking to purchase a vehicle in the next 6-12 months. The project is a lift-and-shift from Domino Data Lab into Dataiku and will utilize Dataiku's more traceable project flow, metrics and checks, and suite of available algorithms. Cox Auto doesn't care about false positives but doesn't want to miss out on potential positives (false negatives), so we recommend using recall as a preferred performance metric when selecting a candidate model.

	Current State	End State	Actions To Close Gap
Objective	The current implementation is extremely complicated and only comprehensible to experienced data scientists, consisting of pure python scripts only.	The new project will make use of Dataiku's more traceable assets like visual recipes whenever possible. The datasets are large so push-down compute to Snowflake will be a centerpiece. The project will be pushed to the automation node for scheduled execution.	1. Confirm the functionality of Dataiku Design and Automation nodes and establish all required data connections 2. Create python code environment with all required package dependencies 3. Join and clean the various Snowflake tables as designed on the Lucidchart diagram
Technical Solution	Domino is currently housing, training, and deploying the H2O random forest models. The project consists of multiple sub-models that effectively impute missing values in the testing set. The "main" model is a binary classifier for whether a customer is near market.	Want to try a series of different algorithms, including importing H2O models into the training loop. Will employ MLFlow to import existing, trained H2O random forest model if necessary. Will explore using app-as-recipes to contain repetitive, customizable tasks such as over/undersampling.	4. Create class balanced training data via under/oversampling with python and in the Lab 5. Use python to generate model training configurations and store as json in managed folder 6. Create config-driven models that includes H2O random forest as custom python and other algorithms 7. Store model performance metrics in an output dataset
Data Details	Snowflake is housing all of the input data. Currently relying on python-generated SQL queries to Snowflake tables for data prep tasks. The input data is also of dubious quality and a series of python transformations are used to control and modify the input as necessary.	Snowflake data cleaning and joining will be converted to visual recipes and run in-database whenever possible. Will implement a series of metrics and checks to ensure data quality on the input datasets.	8. Lift and shift existing Streamlit webapp into Dataiku via code studios 9. Create email reporter scheme tailored to different audiences based on scenario outcomes 10. Output a scored dataset for API access 11. Run model retraining and back testing on a to-be-determined cadence

REFERENCE LUCIDCHART DIAGRAM





Join and clean the various Snowflake tables as designed on the Lucidchart diagram 3

Create class balanced training data via under/oversampling with python and in the Lab 4

Use python to generate model training configurations and store as json in managed folder 5

Create config-driven models that includes H2O random forest as custom python and other algorithms 6

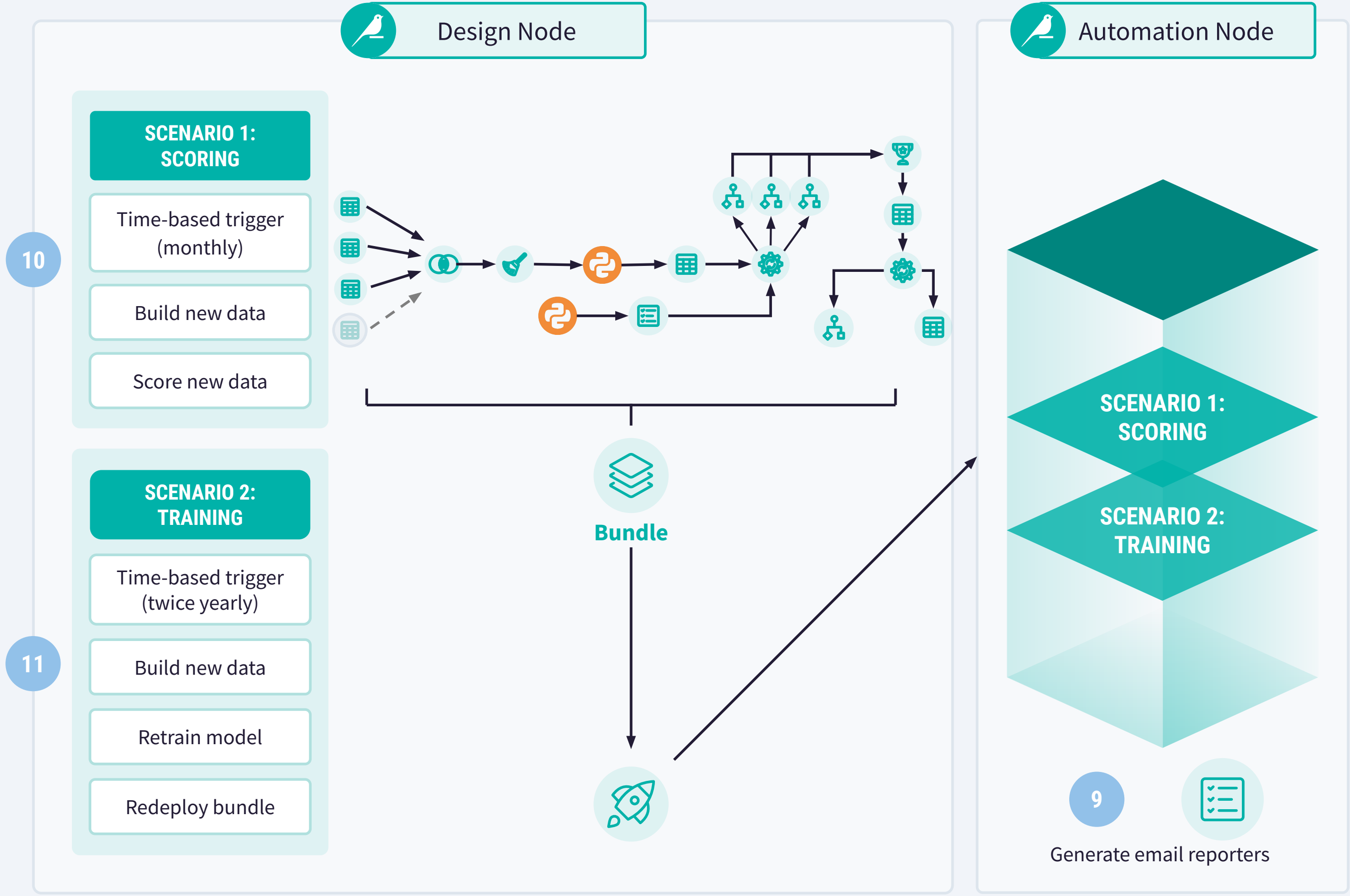
Store model performance metrics in an output dataset 7

Pre-work (actions not shown):

Confirm the functionality of Dataiku Design and Automation nodes and establish all required data connections 1

Create python code environment with all required package dependencies 2

ARCHITECTURE



Lift and shift existing Streamlit webapp into Dataiku via code studios 8

Create email reporter scheme tailored to different audiences based on scenario outcomes 9

Output a scored dataset for API access 10

Run model retraining and back testing on a to-be-determined cadence 11

TASKS & DELIVERABLES

	Phase 1 Project Setup & Discovery	Phase 2 Data Preparation	Phase 3 ML Modelling & Analysis	Phase 4 Automation & Reporting
Activities	• Confirm the functionality of Dataiku Design and Automation nodes and establish all required data connections • Create python code environment with all required package dependencies	• Join and clean the various Snowflake tables as designed on the Lucidchart diagram • Create class balanced training data via under/oversampling with python and in the Lab • Use python to generate model training configurations and store as json in managed folder	• Create config-driven models that includes H2O random forest as custom python and other algorithms • Store model performance metrics in an output dataset	• Lift and shift existing Streamlit webapp into Dataiku via code studios • Create email reporter scheme tailored to different audiences based on scenario outcomes • Output a scored dataset for API access • Run model retraining and back testing on a to-be-determined cadence
Milestones	• Dataiku project with input data pipeline	• Functional data prep pipeline • Prepared dataset for model training	• First iteration of baseline models	• Dataiku project automated via a scenario to send out a report
Assigned to	Cox Auto	Cox Auto	Cox Auto	Cox Auto
Level of effort	**0-1 weeks** (0-3 hours)*	**2-3 weeks** (6-9 hours)*	**2-3 weeks** (6-9 hours)*	**1-2 weeks** (3-6 hours)*

*Hours provided are projected **meeting hours** with your Dataiku data scientist. The customer team is expected to do work outside of coaching sessions. Additionally, the Dataiku data scientist may be required to spend additional hours outside of coaching sessions to prep or respond to follow-up questions.

PROJECT TIMELINE