US Census

Income Prediction

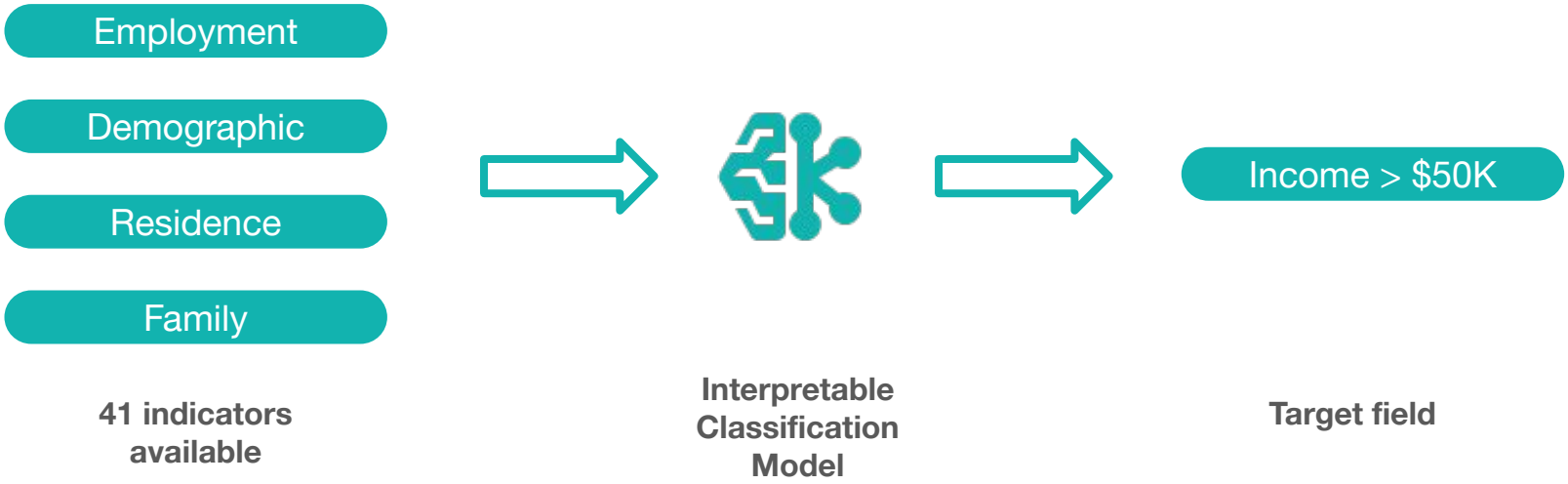
Daniel Soukup

2025.11.03

dataiku

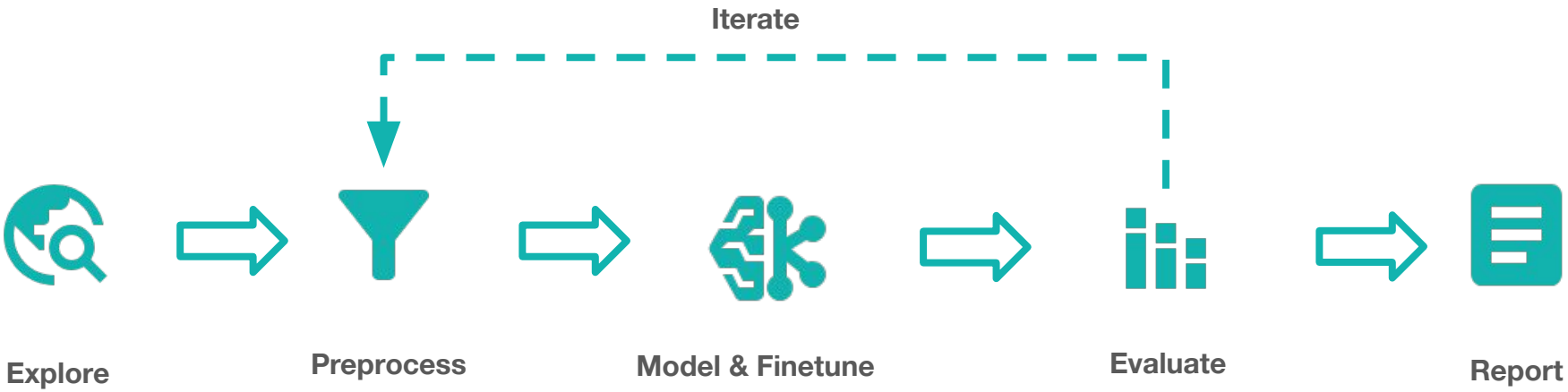
The Challenge

We are tasked with finding identifying characteristics that are associated with a person making more or less than $50,000 per year based a sample dataset from the US Census archive containing information for ~300,000 individuals.



Solution Framework

In addressing this problem, we followed an iterative process. Guided by our initial data exploration, the data was preprocessed and used to create competing classification models. Informed by subsequent evaluation, we refined the preprocessing and modeling steps to achieve higher performance.



Data Exploration

Our first step was to thoroughly review the data sets, gaining statistical insights to the distribution of each field as well as uncovering any data quality concerns. This step informs the following preprocessing and preparation of the data for model fitting.

Data Quality

Key concerns uncovered during EDA:

- **Duplicate** rows and **missing** values
- Highly **skewed distributions** with extreme outliers
- **High cardinality** categorical columns

8%

high income

The **high class imbalance** of the target will affect model tuning and evaluation.

Statistical indicators

Confirming intuition, we found that **education & occupation** fields have significant relationship with having high income.

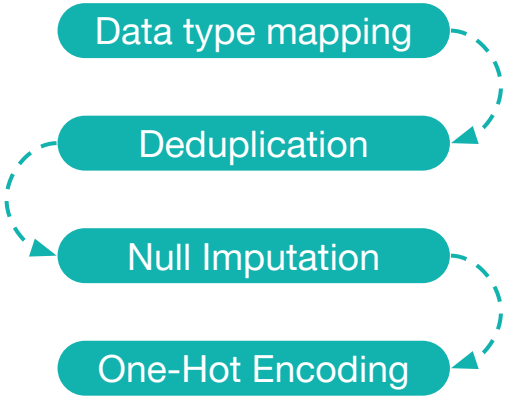
Preprocessing

This step in our solution ensures that the raw data is best prepared for the modeling step. We addressed the data quality issues and encoded the categorical features adequately for our selected tree-ensemble model.

Data Pipelines

We created a **single, modular data processing pipeline** which captures multiple custom data transformation steps. This approach ensures

- **consistency**: the same operation applied to train/test tests,
- **no data leakage,** and
- **reduces tech debt** in the long run.

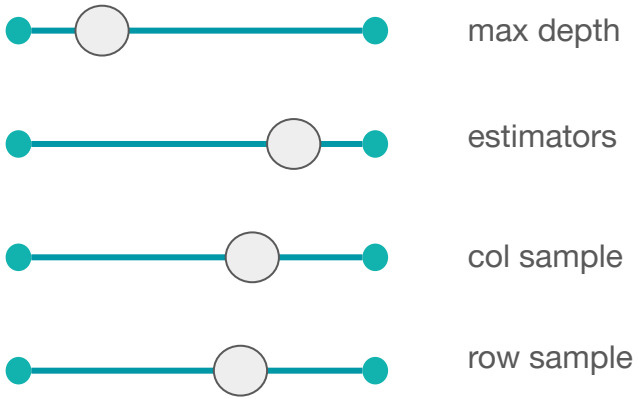


Modeling & Fine Tuning

After the preprocessing of a mix of numeric and categorical features, we optimized an XGBoost binary classification model fine-tuning over a number of hyperparameters, training 20 variations. This model type, which successively combines simple decision trees to form an ensemble, is know to handle high dimensional, complex data sets well while being robust to data skew and outliers.

Tuning Objectives

- **Cross validation** to reduce variance of loss estimates.
- Increase **boosting rounds** to create complex models but limit tree depth and lower column/row sampling rates to reduce overfitting.
- Optimize area **under the precision-recall curve** to account for class imbalance.

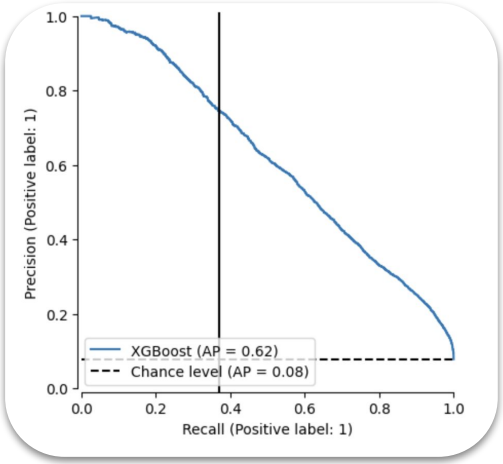


Evaluation & Interpretation

Analysing feature importance highlighted **sex**, **education** and **employment** indicators as most significant, reaffirming a well-know **bias** in this historical dataset. Our predictions correctly identified 38% of high income earners (recall) while these high income predictions being 74% of the time correct (precision).

.38 Recall

.74 Precision

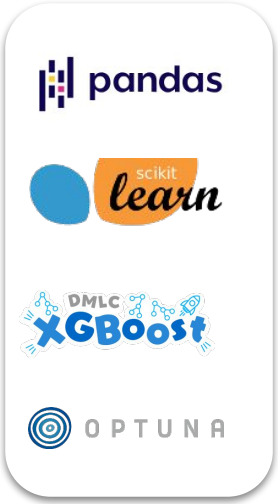


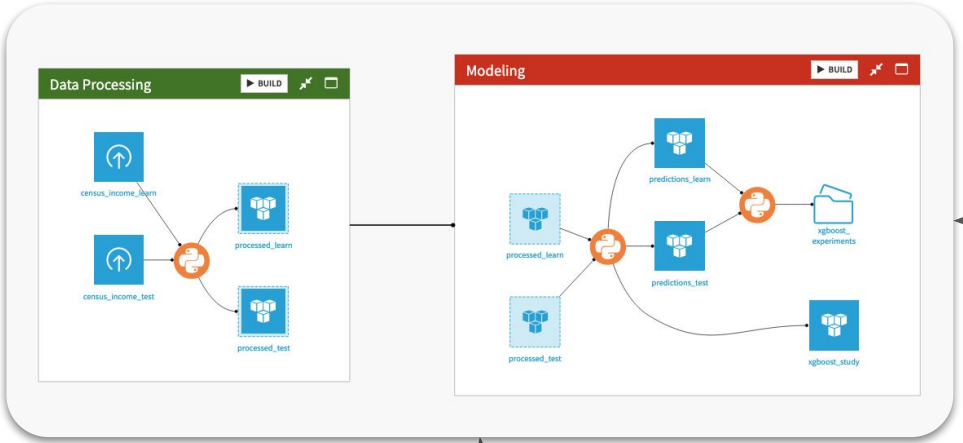
Trade-offs

The recall/precision trade-off can be tuned by adjusting the predicted probability cutoff threshold.

Implementation on DSS Cloud

The complete solution framework was implemented on Dataiku DSS Cloud using Python recipes ran on a custom project environment with 4-CPU cores and 16Gb memory, with end-to-end runtime of 6.5 minutes.





The recipe notebooks are synced with Github.

Data stored on DSS managed AWS S3 storage.

dataiku

Next Steps

Data Enrichment

There are number of steps to explore in order to improve the input data quality & quantity:

- **Balance the classes** using up/down sampling or synthetic data generation
- **Feature engineering** and alternative encoding
- **Data transformations** to adjust scale and skew

MLOps & Evaluation

More in-depth **experiment logging** and refined **analysis of the model errors** could reveal additional short-comings and inform future iterations.

Feature importances and effects can be better understood by model agnostic methods such as SHAP or LIME.

Alternative Model

A more thorough search through a **larger hyperparameter space** can further improve model performance.

Alternative model types can be tested, such as CatBoost which is build for high-dimensional categorical data.

Q & A

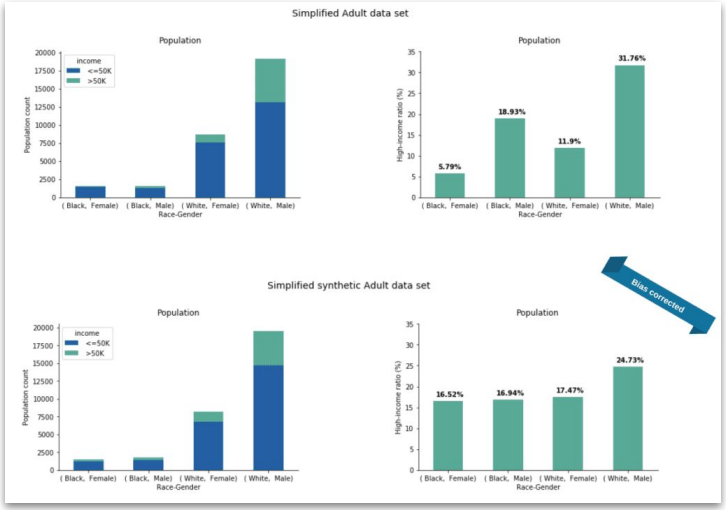
Project: Github

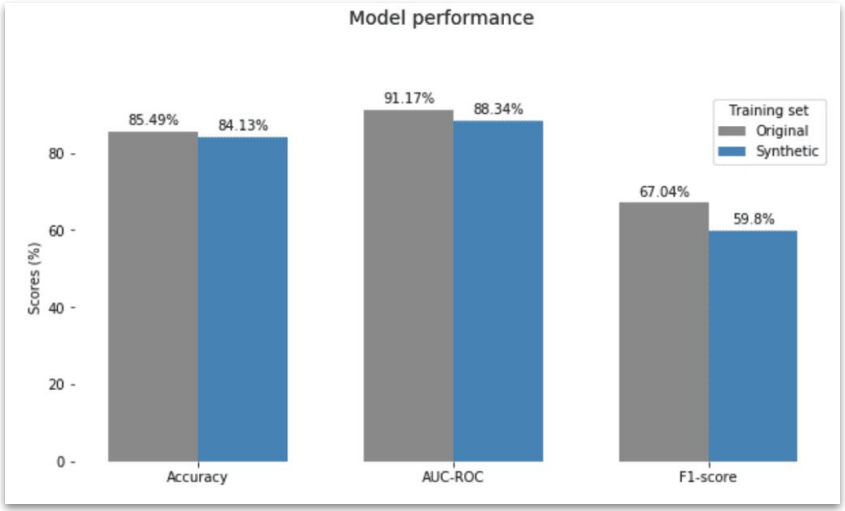
Contact: LinkedIn

dataiku

Appendix: Bias Mitigation

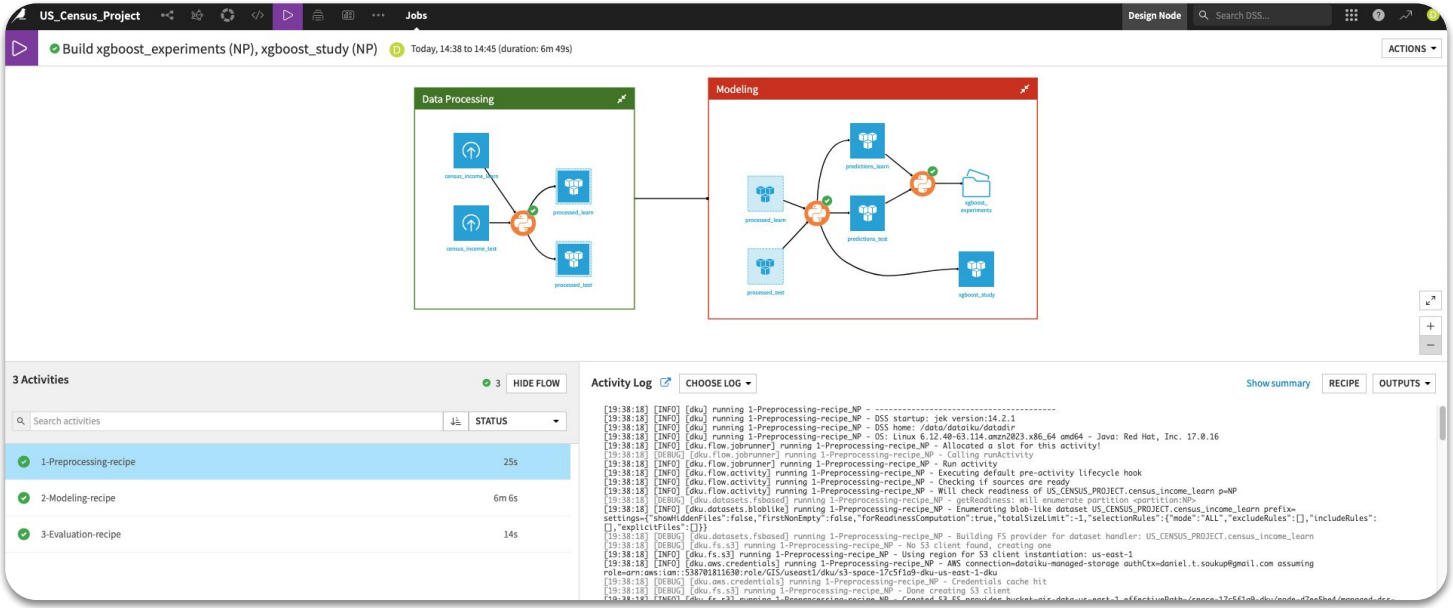
Previously, I've implemented privacy and fairness risk mitigation with bias-corrected synthetic data: this allows for privacy-preserving data sharing and also aids the fair treatment of customers (data subjects) in downstream analysis and machine learning tasks. See more here and in our paper accepted at an ICLR 2021 workshop.



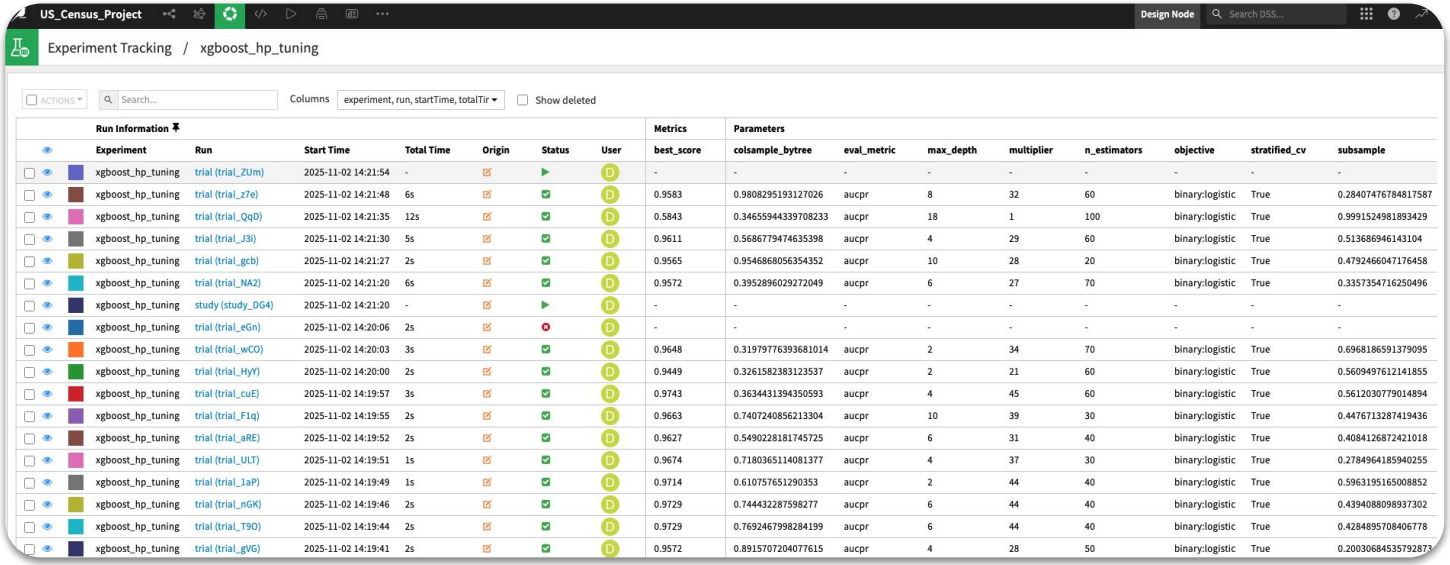


dataiku

Appendix: End-to-End Flow Build



Appendix: Experiment Tracking



	Run Information							Metrics	Parameters							
	Experiment	Run	Start Time	Total Time	Origin	Status	User	best_score	colsample_bytree	eval_metric	max_depth	multiplier	n_estimators	objective	stratified_cv	subsample
	xgboost_hp_tuning	trial (trial_ZUm)	2025-11-02 14:21:54	-		▶	D	-	-	-	-	-	-	-	-	-
	xgboost_hp_tuning	trial (trial_z7e)	2025-11-02 14:21:48	6s		✓	D	0.9583	0.9808295193127026	aucpr	8	32	60	binary:logistic	True	0.28407476784817587
	xgboost_hp_tuning	trial (trial_QqD)	2025-11-02 14:21:35	12s		✓	D	0.5843	0.34655944339708233	aucpr	18	1	100	binary:logistic	True	0.9991524981893429
	xgboost_hp_tuning	trial (trial_J3i)	2025-11-02 14:21:30	5s		✓	D	0.9611	0.5686779474635398	aucpr	4	29	60	binary:logistic	True	0.5136869461431O4
	xgboost_hp_tuning	trial (trial_gcb)	2025-11-02 14:21:27	2s		✓	D	0.9565	0.9546868056354352	aucpr	10	28	20	binary:logistic	True	0.4792466047176458
	xgboost_hp_tuning	trial (trial_NA2)	2025-11-02 14:21:20	6s		✓	D	0.9572	0.3952896029272049	aucpr	6	27	70	binary:logistic	True	0.3357354716250496
	xgboost_hp_tuning	study (study_DG4)	2025-11-02 14:21:20	-		▶	D	-	-	-	-	-	-	-	-	-
	xgboost_hp_tuning	trial (trial_eGn)	2025-11-02 14:20:06	2s		✗	D	-	-	-	-	-	-	-	-	-
	xgboost_hp_tuning	trial (trial_wCO)	2025-11-02 14:20:03	3s		✓	D	0.9648	0.31979776393681014	aucpr	2	34	70	binary:logistic	True	0.6968186591379095
	xgboost_hp_tuning	trial (trial_HyY)	2025-11-02 14:20:00	2s		✓	D	0.9449	0.3261582383123537	aucpr	2	21	60	binary:logistic	True	0.5609497612141855
	xgboost_hp_tuning	trial (trial_cuE)	2025-11-02 14:19:57	3s		✓	D	0.9743	0.3634431394350593	aucpr	4	45	60	binary:logistic	True	0.5612030779014894
	xgboost_hp_tuning	trial (trial_F1q)	2025-11-02 14:19:55	2s		✓	D	0.9663	0.7407240856213304	aucpr	10	39	30	binary:logistic	True	0.4476713287419436
	xgboost_hp_tuning	trial (trial_aRE)	2025-11-02 14:19:52	2s		✓	D	0.9627	0.5490228181745725	aucpr	6	31	40	binary:logistic	True	0.4084126872421018
	xgboost_hp_tuning	trial (trial_ULT)	2025-11-02 14:19:51	1s		✓	D	0.9674	0.7180365114081377	aucpr	4	37	30	binary:logistic	True	0.2784964185940255
	xgboost_hp_tuning	trial (trial_1aP)	2025-11-02 14:19:49	1s		✓	D	0.9714	0.610757651290353	aucpr	2	44	40	binary:logistic	True	0.5963195165008852
	xgboost_hp_tuning	trial (trial_nGK)	2025-11-02 14:19:46	2s		✓	D	0.9729	0.744432287598277	aucpr	6	44	40	binary:logistic	True	0.4394088098937302
	xgboost_hp_tuning	trial (trial_T9O)	2025-11-02 14:19:44	2s		✓	D	0.9729	0.7692467998284199	aucpr	6	44	40	binary:logistic	True	0.4284895708406778
	xgboost_hp_tuning	trial (trial_gVG)	2025-11-02 14:19:41	2s		✓	D	0.9572	0.8915707204077615	aucpr	4	28	50	binary:logistic	True	0.20030684535792873

Appendix: HP Tuning Analysis