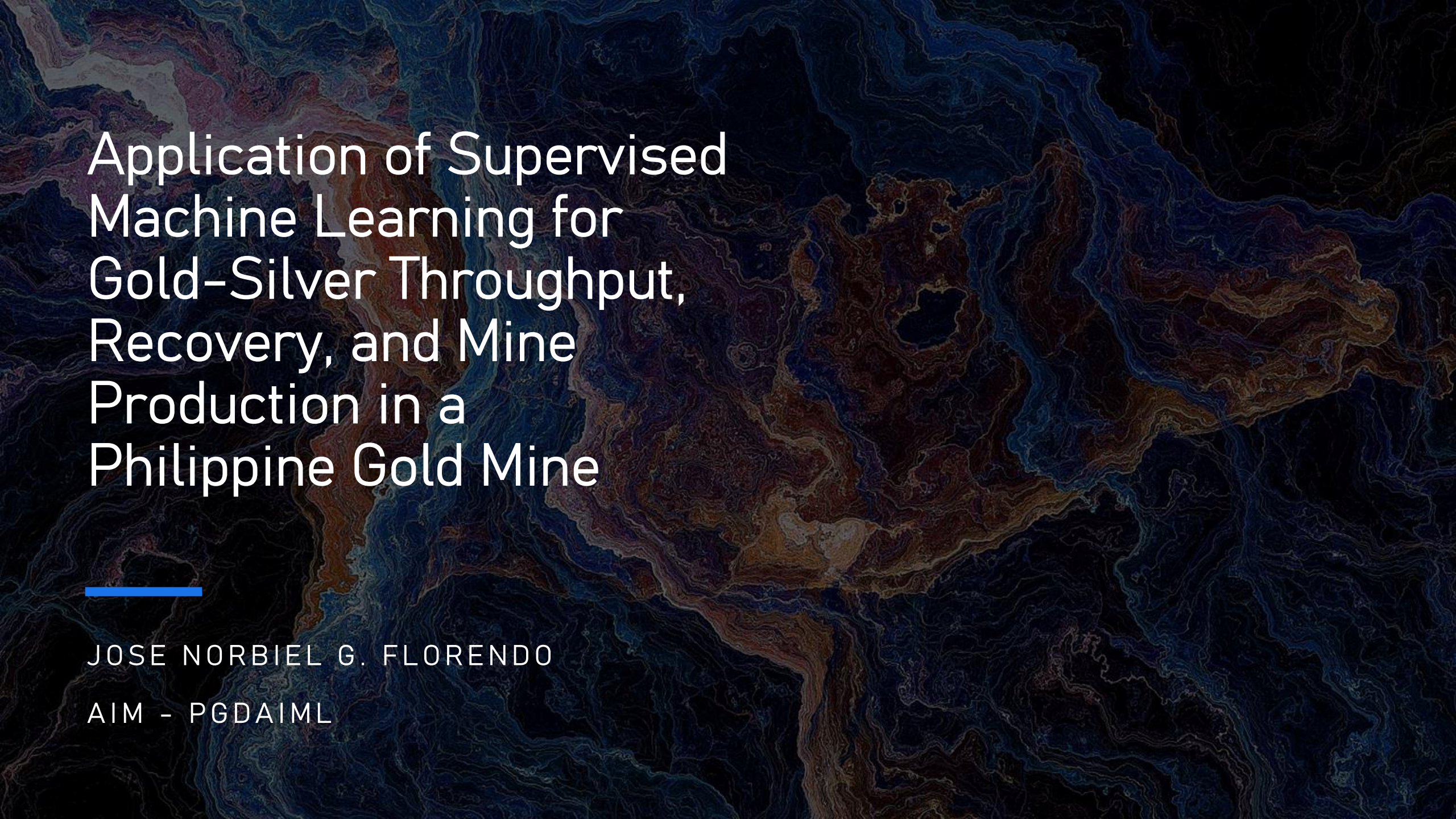


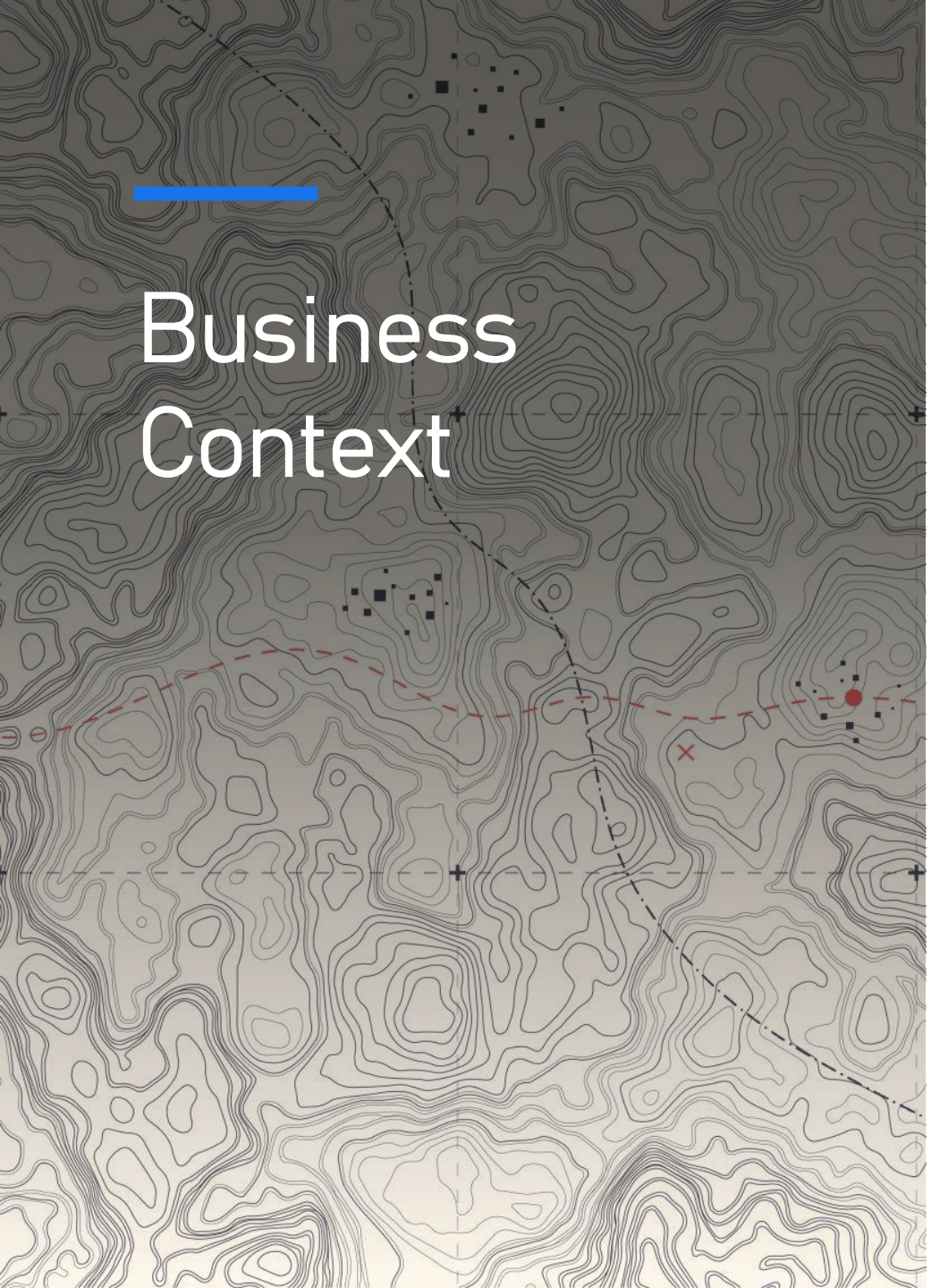
Application of Supervised Machine Learning for Gold-Silver Throughput, Recovery, and Mine Production in a Philippine Gold Mine

JOSE NORBIEL G. FLORENDO

AIM - PGDAIML

Outline

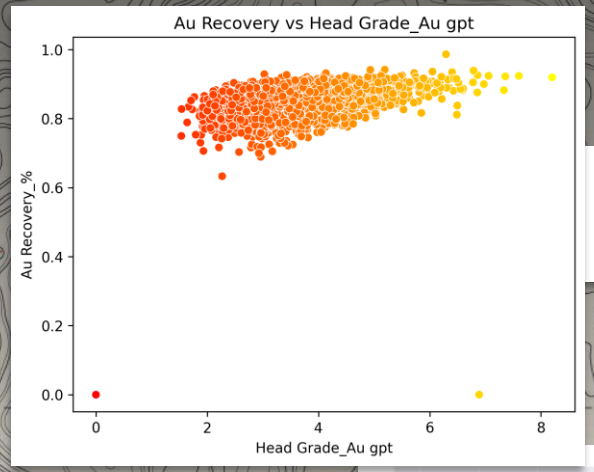
- Business Context and Problem Statement
- Methodology
- Exploratory Data Analysis
- Model Creation
- Results and Discussion
  - Performance of Baseline Models
  - Performance of Tuned Models
  - Comparison of Tuned and Baseline Models
  - Model Selection
- Conclusion and Recommendation



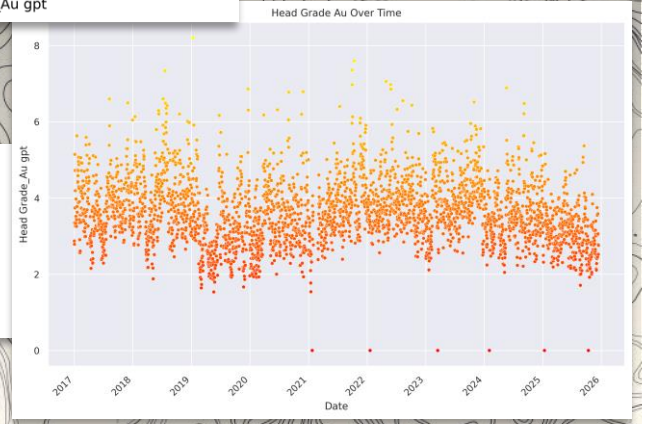
Business Context

- A mining company's financial profitability and viability depends almost entirely on its level of output (productivity) and its output efficiency.
- Productivity and Output Efficiency in a Gold-Silver mine is measured respectively using **Gold and Silver Output** and **Recovery** (for Gold and Silver)
  - Recovery (%) is the total metal output successfully extracted from the total metal content of the ore input
    - $\frac{\text{90g gold extracted from 1 ton of Ore}}{\text{100g gold concentration in 1 ton of Ore}}$ = 90% Gold Recovery
  - Mine Tonnage is the total amount of ore (in tons) produced by a mine for a specific period.
- For a mining company to be profitable and efficient it must be able to maintain a high enough level of productivity and output efficiency given a wide range of inputs.

Problem Statement



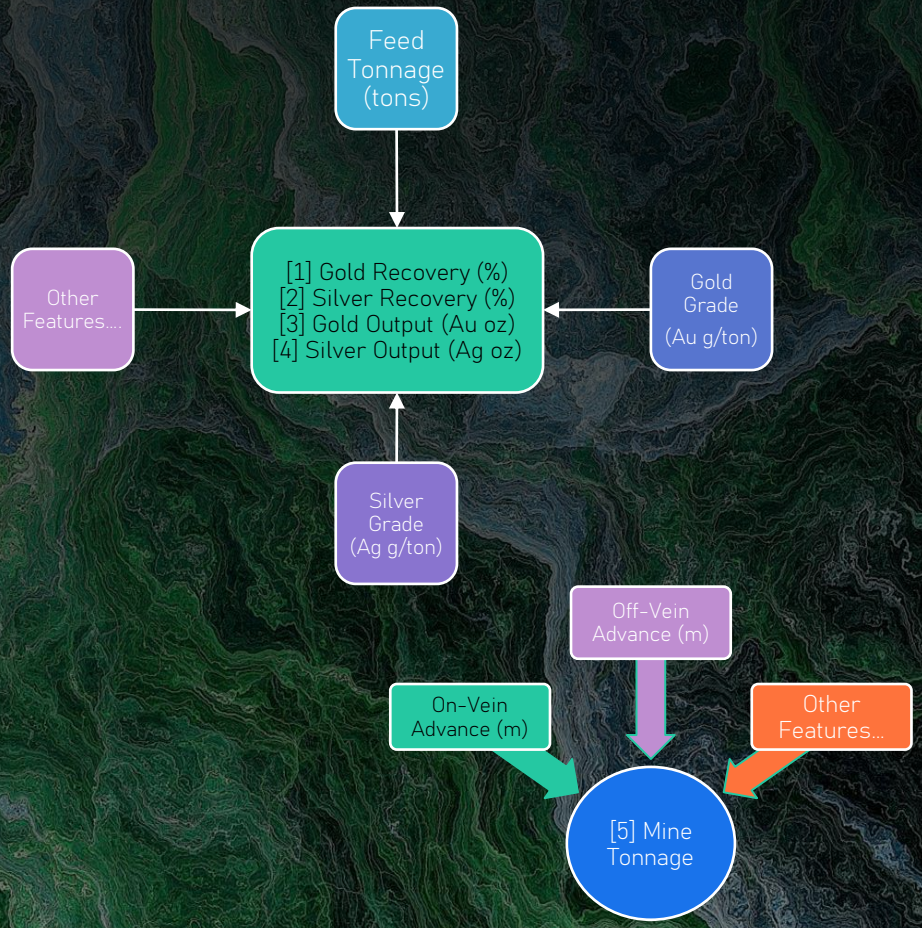
Gold Recovery is not linear



Gold Grade is not constant over time

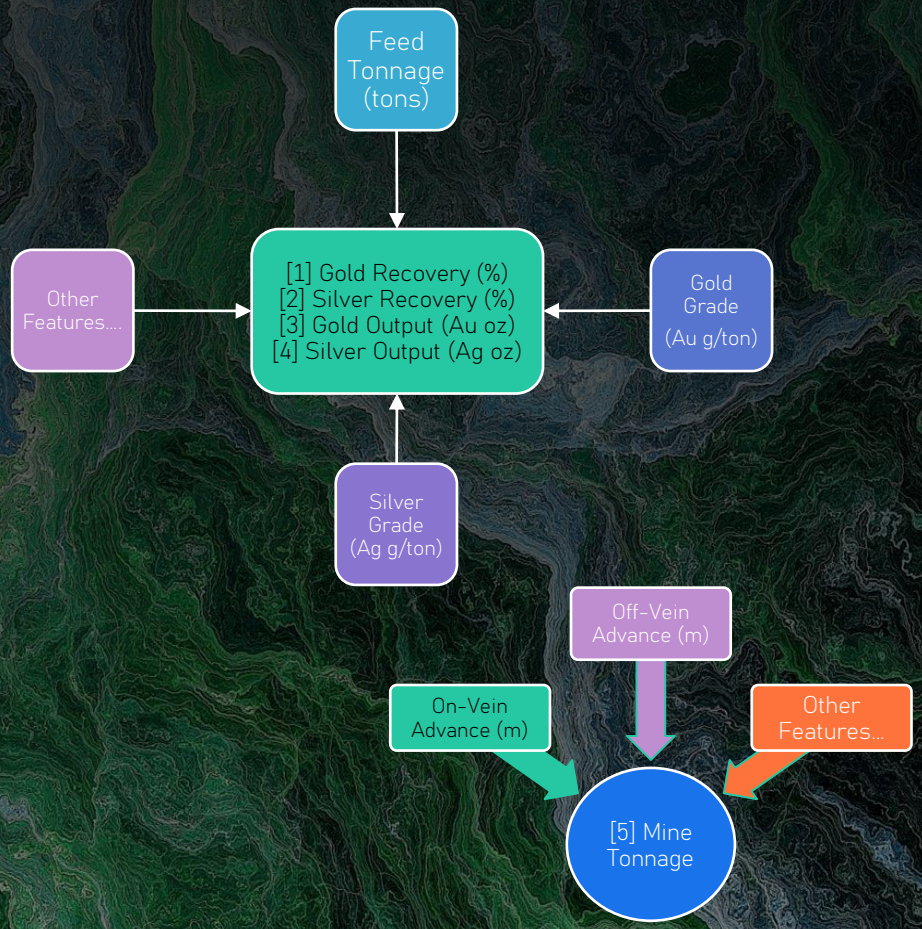
- **Business problem:**
  - Recovery and Output can only be calculated **after the metal** (gold and silver) **has already been extracted** from the ore.
  - Recovery prediction is **back-calculated through Linear Regression**, which may not have sufficient predictive power for non-linear behaviors and to handle varying ore grades.
- **Business opportunity:**
  - If the mining company can **anticipate** and **adapt** swiftly by **accurately predicting key mining outputs**: [1] gold and [2] silver recovery, [3] the total ounces of gold and [4] silver produced, and the [5] using minimal inputs, the mining company can **change operational parameters immediately** to increase ore recovery, and this ultimately **maximizes value** of its raw product.

Methodology



- **Data Science Task:**
  - the data science problem is comprised of three main prediction/regression tasks:
    - **Gold and Silver Throughput Prediction:** Predict 'Gold Ounces Produced' and 'Silver Ounces Produced'.
    - **Gold and Silver Recovery Prediction:** Predict 'Gold Recovery', 'Gold and 'Silver Recovery'.
    - **Mine Tonnage Prediction:** Predict 'Mine Tonnage_tons'
- **Prediction/Regression Task**:
  - Create a prediction model using several methods such as ensemble, artificial neural network and regression techniques which encompass the following **seven (7) methods** in this study:
    - **Linear Regression**, **XGBoost** Regression, **LightGBM** Regression, **CatBoost** Regression, **Decision Tree** Regression, **Support Vector Machine** Regression, **Multi-layer Perceptron** Regression (ANN)
- **Success Metrics**:
  - All the 7 regression models will be evaluated using the following performance metrics to assess model performance:
    - **R-squared** (R^2), **Mean Absolute Error** (MAE), **Mean Squared Error** (MSE), **Root Mean Squared Error** (RMSE)

Methodology

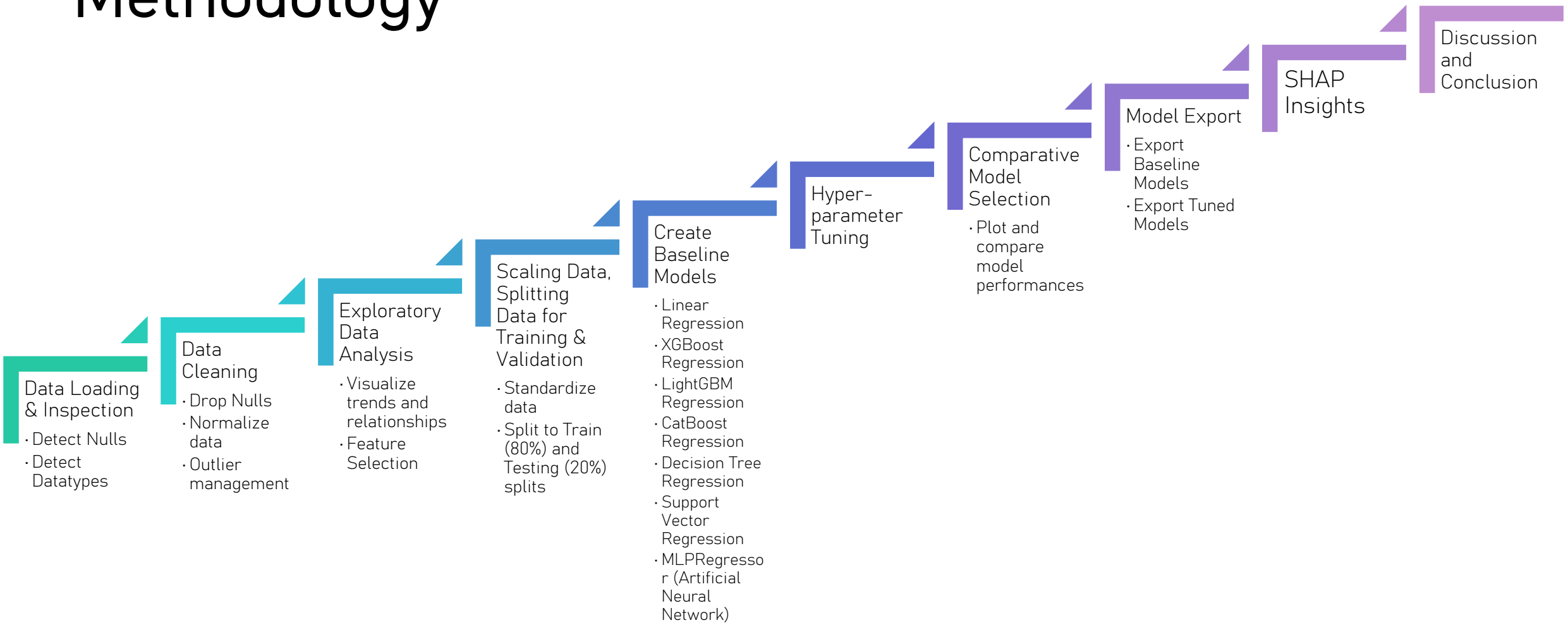


- **Data Science Task:**
  - the data science problem is comprised of three main prediction/regression tasks:
    - **Gold and Silver Throughput Prediction:**
      - Target Variable: 'Gold Ounces Produced' and 'Silver Ounces Produced'.
      - Predictor Variables: 'Gold Grade', 'Feed Tonnage', 'Silver Grade'
    - **Gold and Silver Recovery Prediction:**
      - Target Variable: 'Gold Recovery', 'Gold and 'Silver Recovery'
      - Predictor Variables: 'Gold Grade', 'Feed Tonnage', 'Silver Grade'
    - **Mine Tonnage Prediction:**
      - Target Variable: 'Mine Tonnage_tons'
      - Predictor Variables: 'On-vein Advance', 'Off-vein Advance', 'Total_Advance' (feature-engineered variable)

Other features may exist to give our models higher predictive accuracy but were not yet included in this study since:

1. Some features can only be recorded after the entire mining/milling process has already completed, making our prediction less relevant
2. Some features are too cumbersome to collect for the mine to make a good enough and time-sensitive prediction

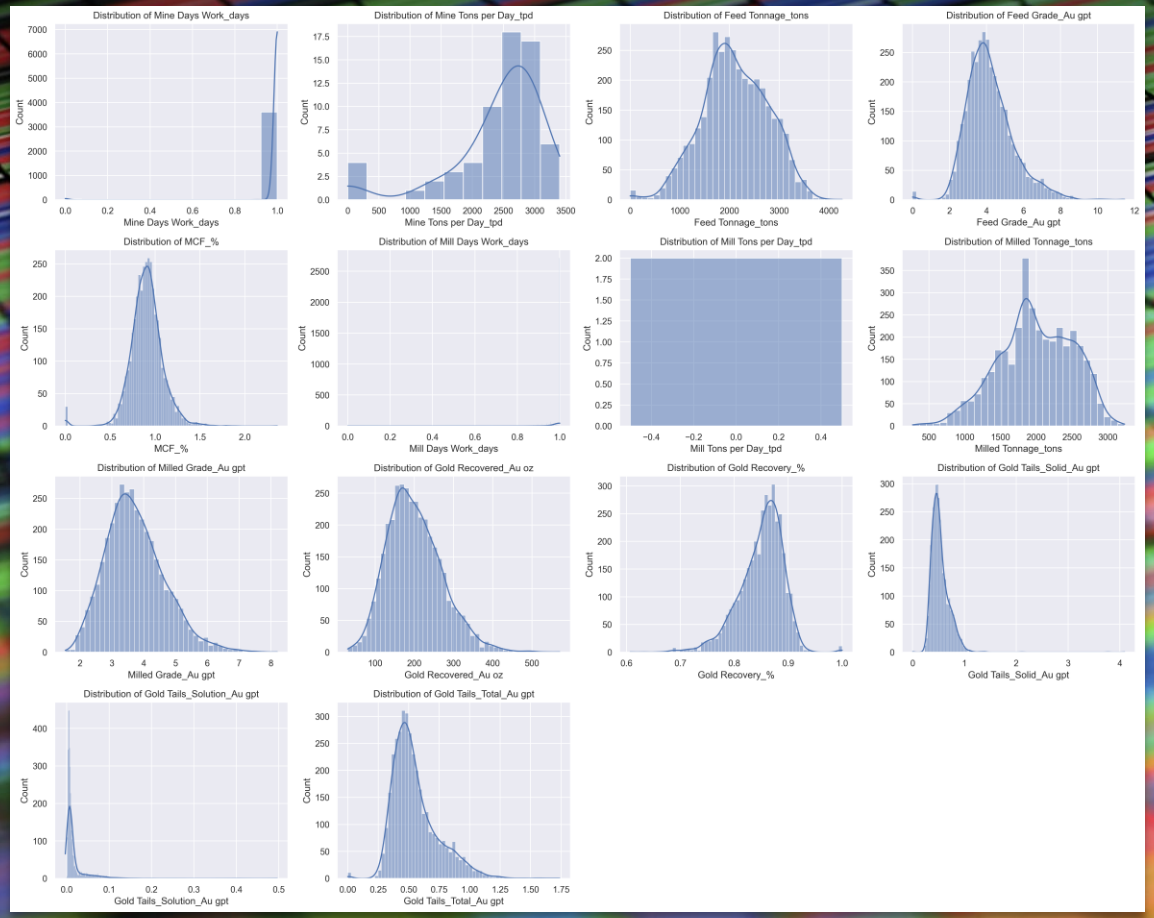
Methodology



Exploratory Data Analysis

DF_MCF, DF_MILL_REPORT,
DF_MINE_PRODUCTION, DF_MINE_ADVANCE

Data Overview

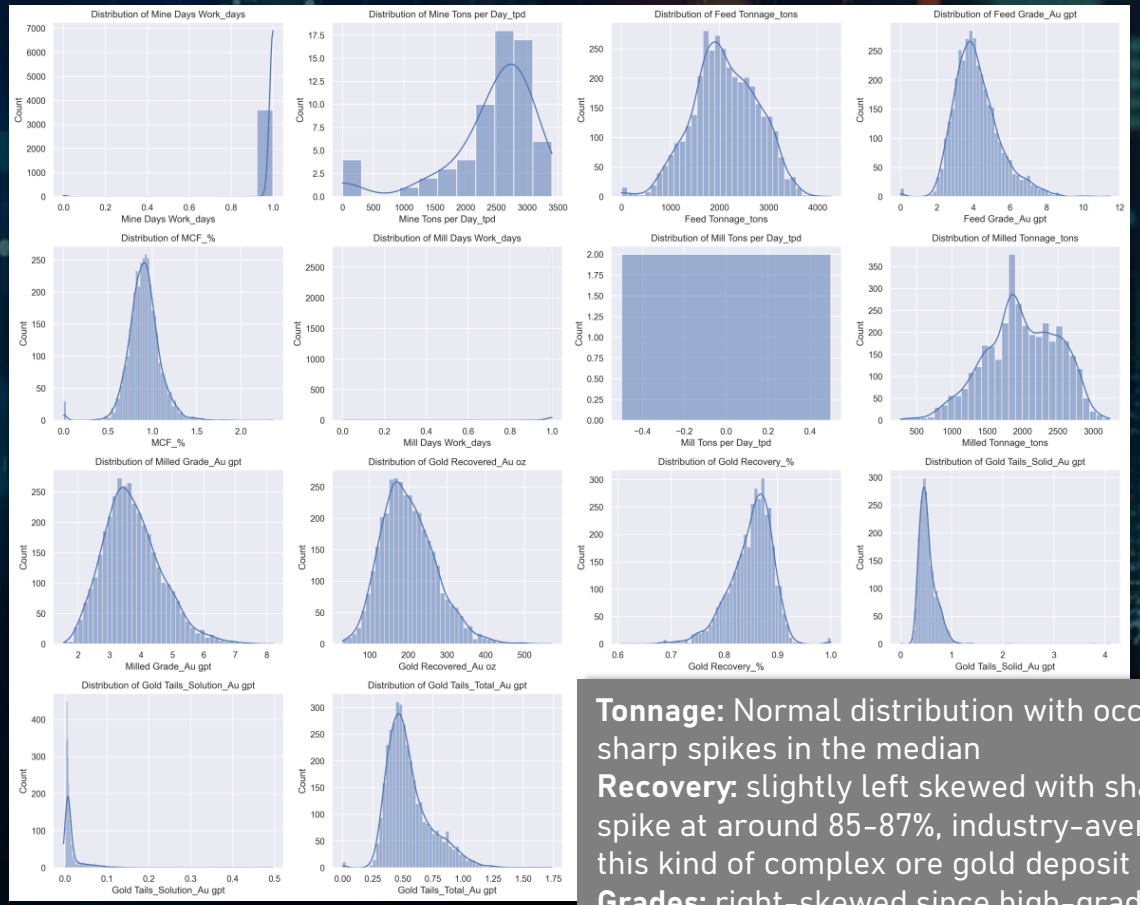


- **Dataset 1/4: df_mcf**
  - MCF, Mill_Reports, Mine_Production, Mine_Advance

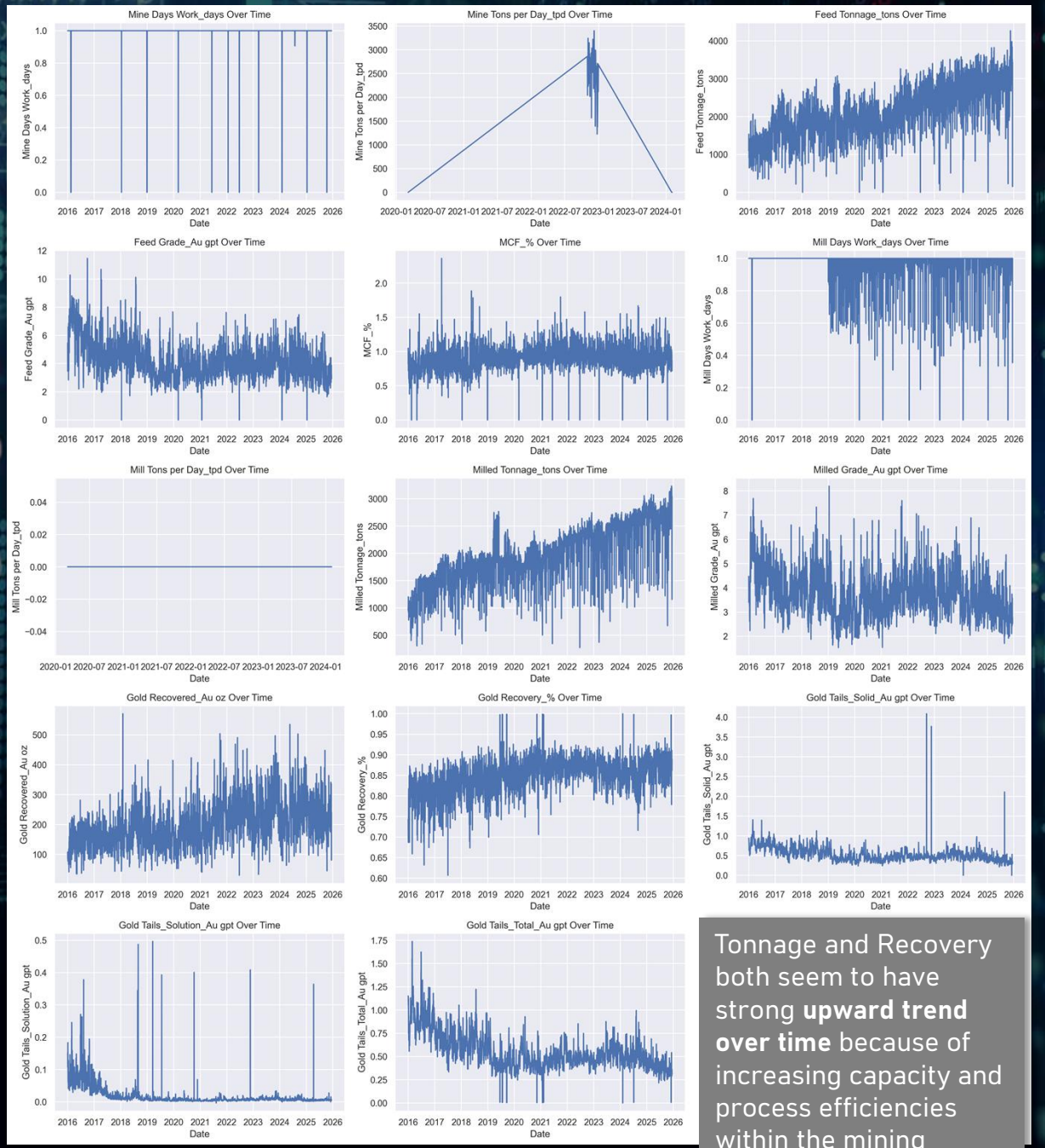
Column Name	Non-Null Count	Data Type	Unit	Min Value	Max Value	Average Value
Date	3637	datetime64[ns]		01/01/2016	15/12/2025	23/12/2020
Mine Days Work_days	3637	float64	days	0.000	1.000	0.993
Mine Tons per Day_tpd	65	float64	tpd	0.000	3402.000	2413.788
Feed Tonnage_tons	3621	float64	tons	0.000	4267.563	2108.104
Feed Grade_Au gpt	3621	float64	Au gpt	0.000	11.463	4.191
MCF_%	3633	float64	%	0.000	2.362	0.905
Mill Days Work_days	3633	float64	days	0.000	1.000	0.962
Mill Tons per Day_tpd	2	float64	tpd	0.000	0.000	0.000
Milled Tonnage_tons	3612	float64	tons	272.228	3233.961	1985.267
Milled Grade_Au gpt	3612	float64	Au gpt	1.535	8.201	3.736
Gold Recovered_Au oz	3612	float64	Au oz	30.047	570.830	201.217
Gold Recovery_%	3612	float64	%	0.607	1.000	0.850
Gold Tails_Solid_Au gpt	3612	float64	Au gpt	0.000	4.091	0.532
Gold Tails_Solution_Au gpt	3612	float64	Au gpt	-0.003	0.497	0.018
Gold Tails_Total_Au gpt	3612	float64	Au gpt	0.000	1.739	0.552

MCF Dataset was originally intended to encompass both Mine Production and Mill Production datasets as one combined dataset but has some relative inconsistencies in terms of ore output tallies as well as date-tagging. More importantly, this dataset does not have dataset for Silver. For the capstone, the dataset was only used to visualize long term time-series trends (if any) in the dataset.

Data Overview: df_mcf

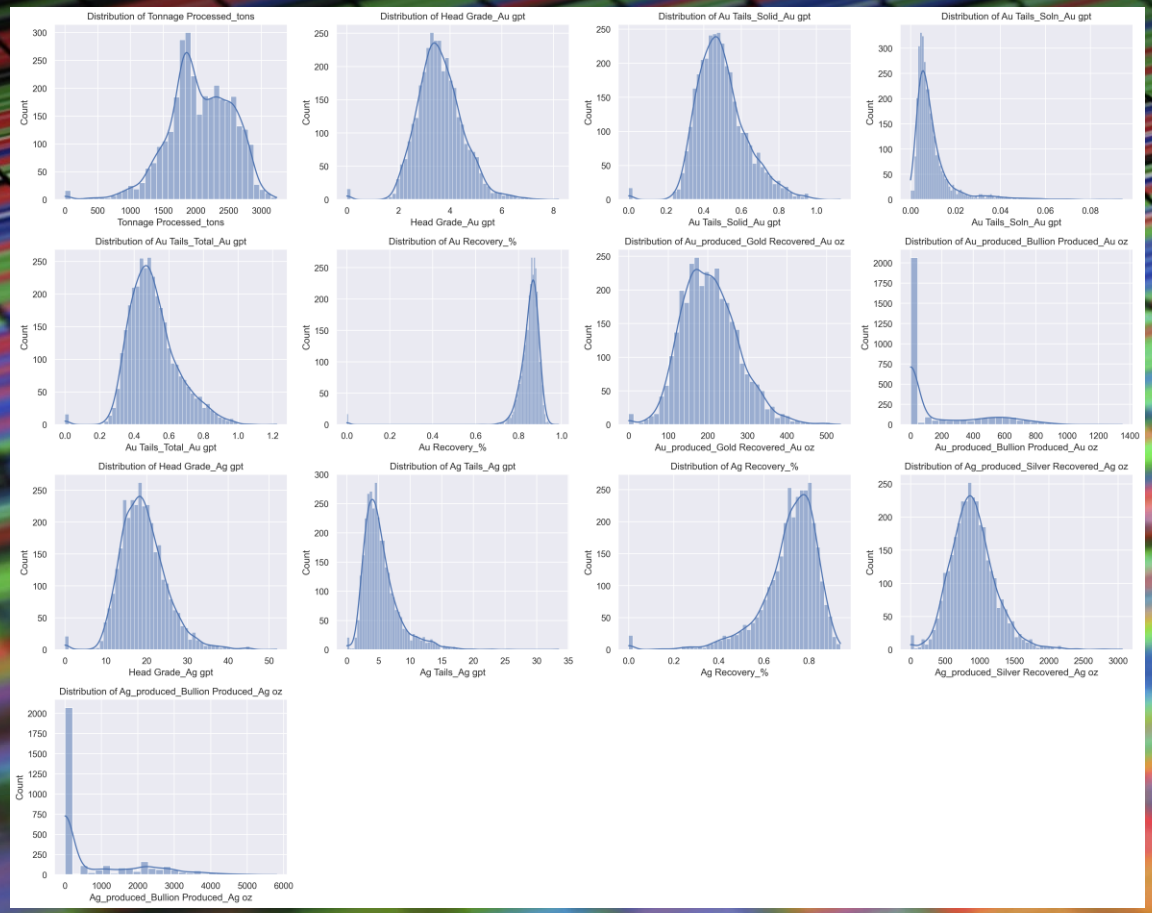


Tonnage: Normal distribution with occasional sharp spikes in the median
Recovery: slightly left skewed with sharp spike at around 85-87%, industry-average for this kind of complex ore gold deposit
Grades: right-skewed since high-grade ore is much rarer



Tonnage and Recovery both seem to have strong **upward trend over time** because of increasing capacity and process efficiencies within the mining company.

Data Overview

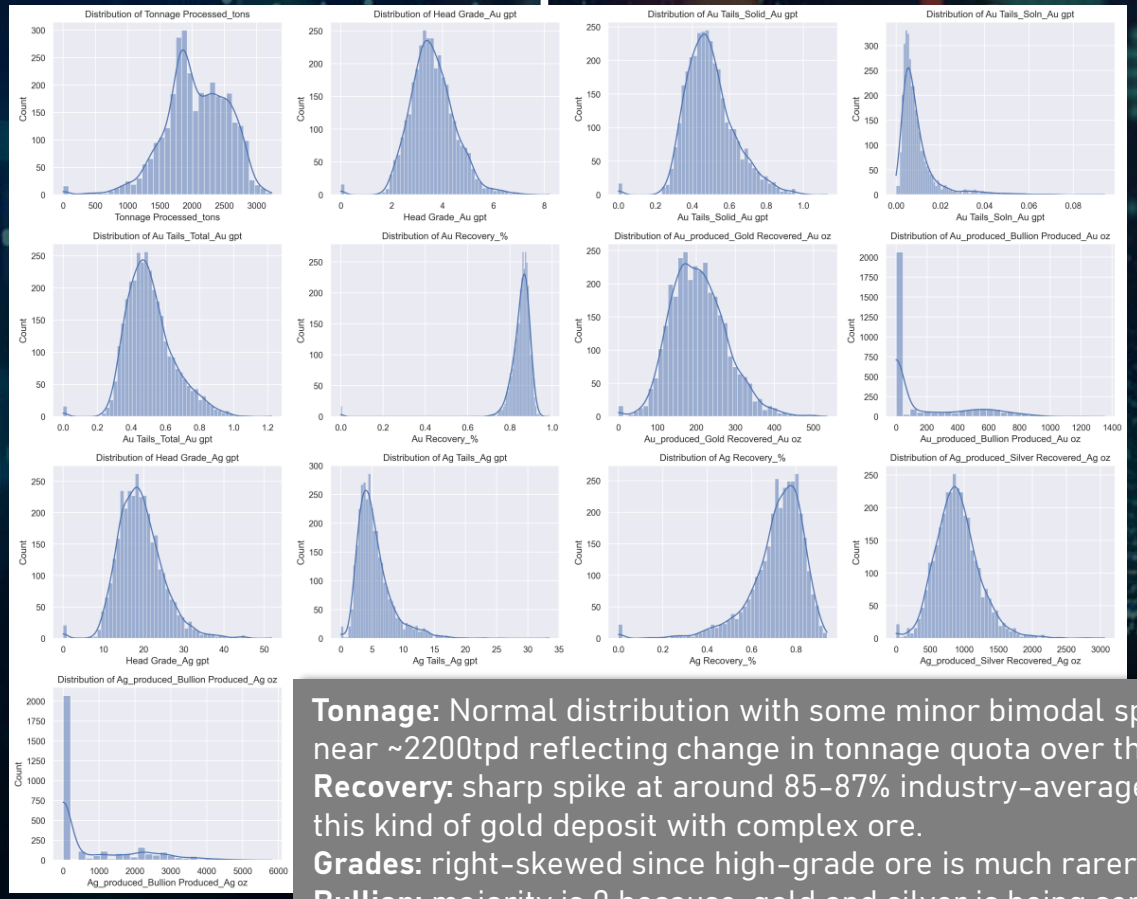


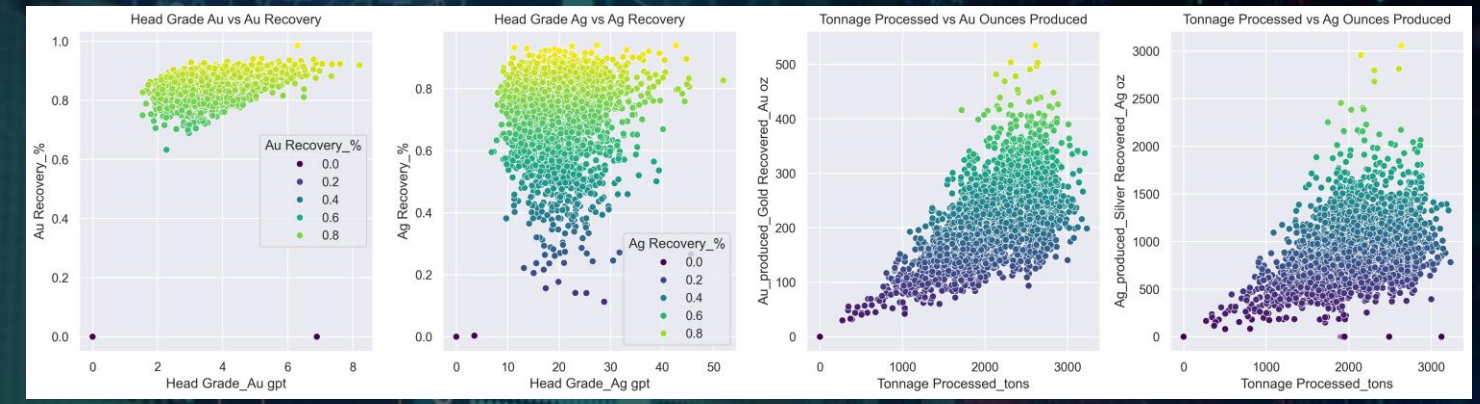
- **Dataset 2/4: df_mill_reports**
  - MCF, **Mill_Reports**, Mine_Production, Mine_Advance

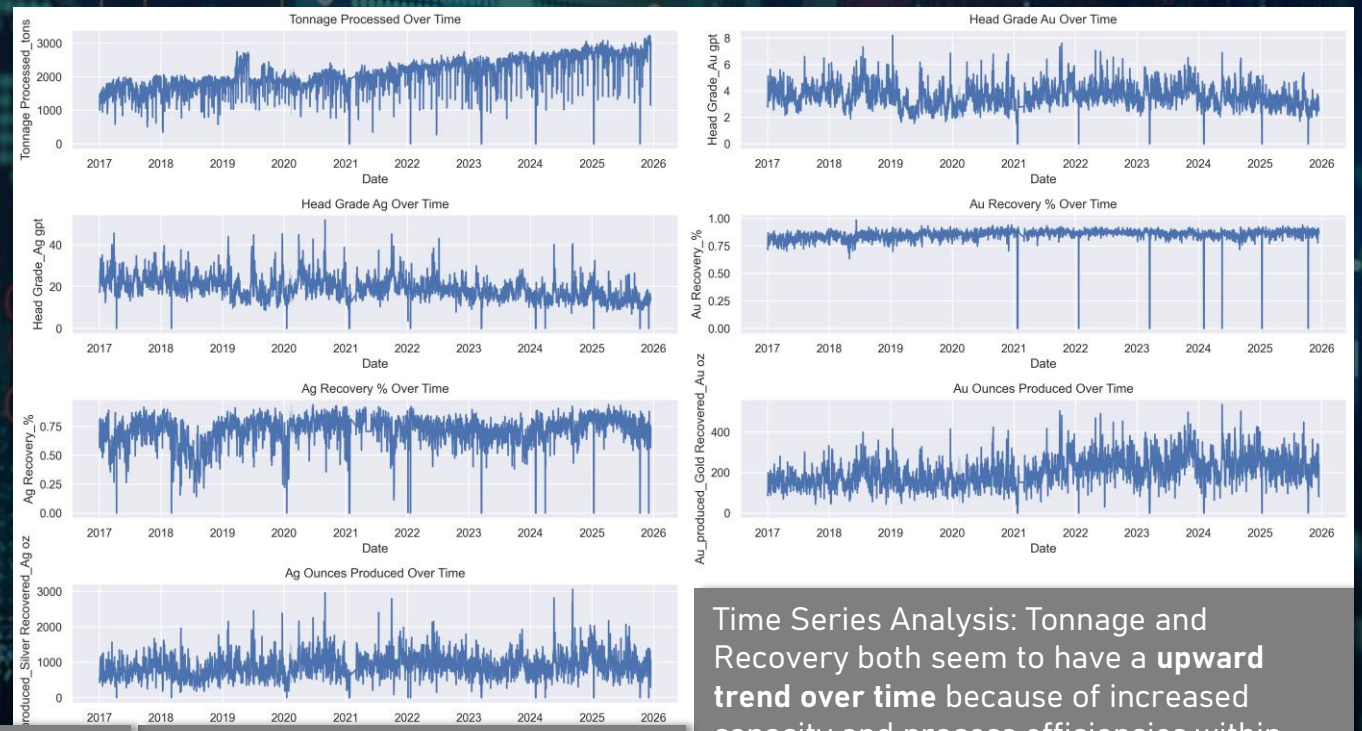
Column Name	Non-Null Count	Data Type	Unit	Min Value	Max Value	Ave Value
Date	3263	datetime64[ns]		01/01/2017	15/12/2025	22/06/2021
Tonnage Processed_tons	3263	float64	tons	0	3233.961	2056.116
Head Grade_Au gpt	3263	float64	Au gpt	0	8.201	3.614
Au Tails_Solid_Au gpt	3263	float64	Au gpt	0	1.128	0.496
Au Tails_Soln_Au gpt	3263	float64	Au gpt	0	0.094	0.010
Au Tails_Total_Au gpt	3263	float64	Au gpt	0	1.224	0.510
Au Recovery_%	3263	float64	%	0	0.987	0.851
Au_produced_Gold Recovered_Au oz	3263	float64	Au oz	0	535.035	205.609
Au_produced_Bullion Produced_Au oz	3260	float64	Au oz	0	1352.196	171.869
Head Grade_Ag gpt	3263	float64	Ag gpt	0	51.856	19.207
Ag Tails_Ag gpt	3263	float64	Ag gpt	0	33.464	5.384
Ag Recovery_%	3263	float64	%	0	0.939	0.714
Ag_produced_Silver Recovered_Ag oz	3263	float64	Ag oz	0	3060.283	901.278
Ag_produced_Bullion Produced_Ag oz	3260	float64	Ag oz	0	5812.123	769.942

Mill Reports dataset contains the complete dataset coming from the Mill Division and will become the source dataset for training the Au-Ag Recovery and Au-Ag Output.

Data Overview: df_mill_reports





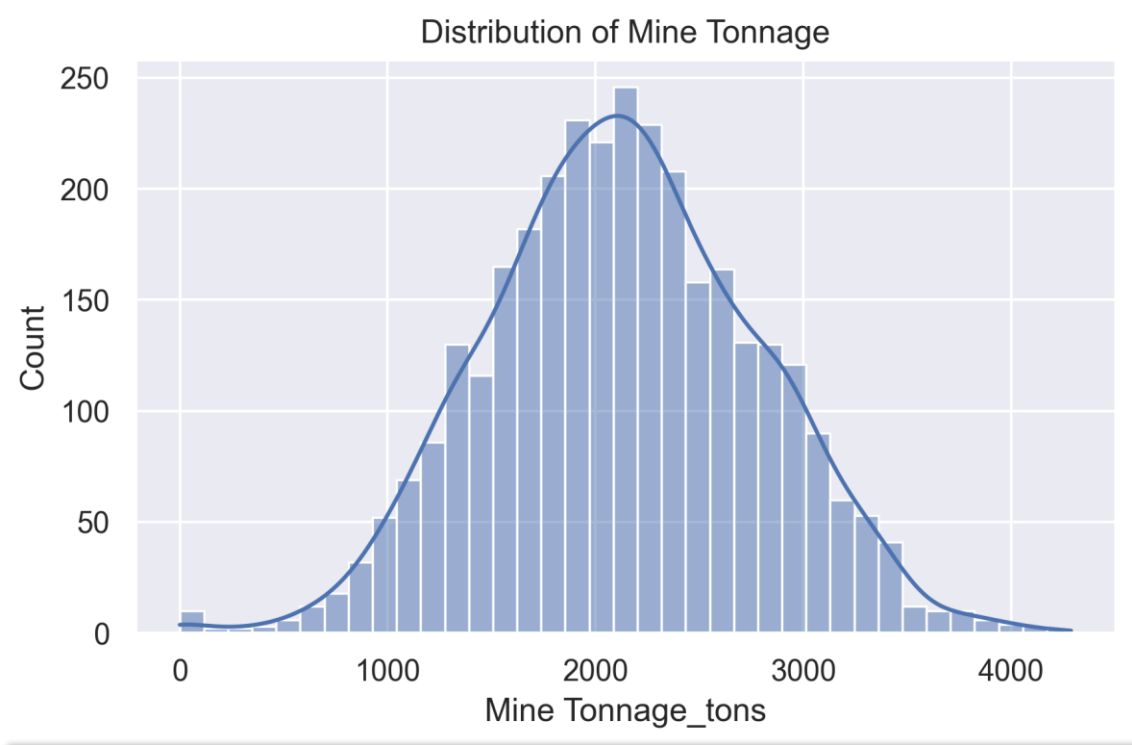


Tonnage: Normal distribution with some minor bimodal spike near ~2200tpd reflecting change in tonnage quota over the years
Recovery: sharp spike at around 85-87% industry-average for this kind of gold deposit with complex ore.
Grades: right-skewed since high-grade ore is much rarer
Bullion: majority is 0 because, gold and silver is being sent in lots and normally produced once a week

Scatterplot Analysis:
Head grade for Au and Ag are not Linear Related with Recovery
Au and Ag ounces increase as tonnage increases but has much higher dispersion at higher tonnages

Time Series Analysis: Tonnage and Recovery both seem to have a **upward trend over time** because of increased capacity and process efficiencies within the mining company
Head-grade for Au and Ag are stable but has some long-term downward trend reflecting lower ore grades as time passes

Data Overview

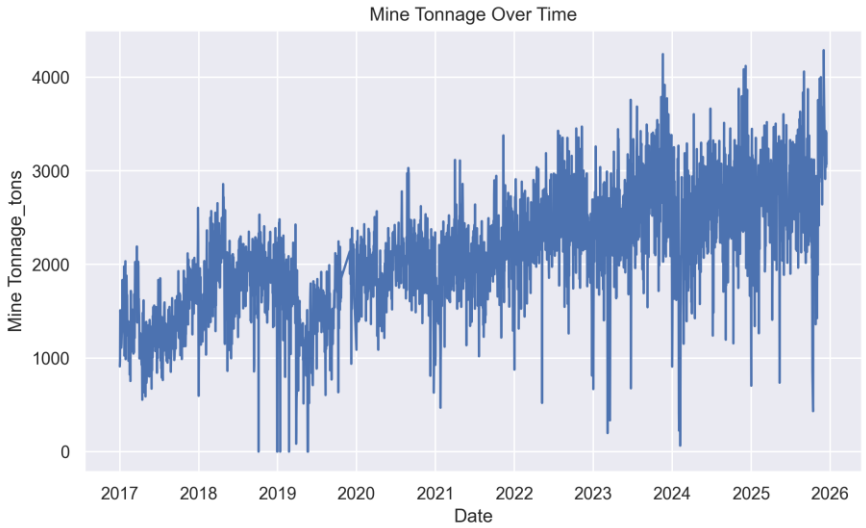


Mine Tonnage has a strong normal distribution hovering at around 2000tpd

- **Dataset 3/4: df_mine_production**
  - MCF, Mill_Reports, **Mine_Production**, Mine_Advance

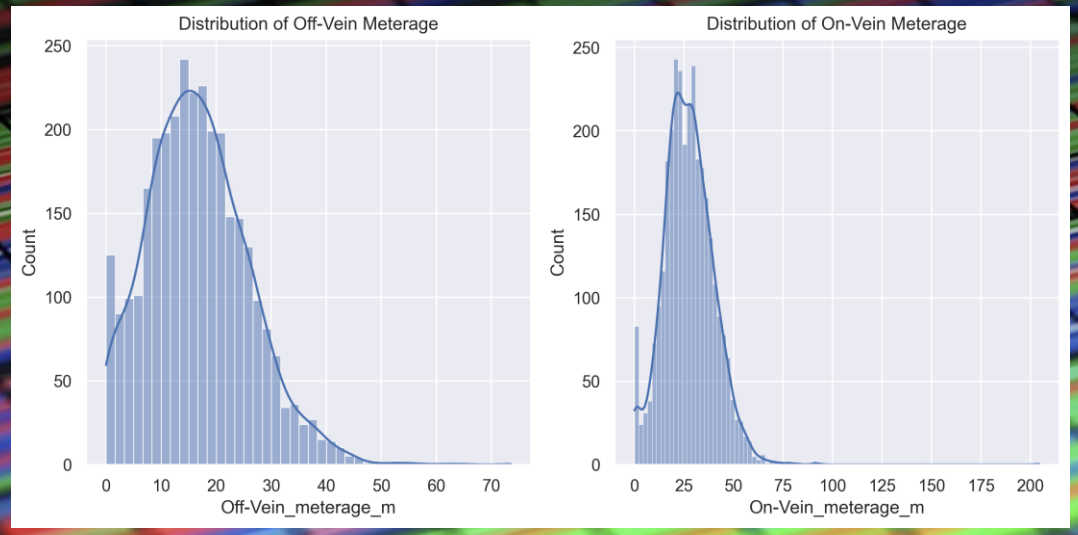
Column Name	Non-Null Count	Data Type	Unit	Min Value	Max Value	Average Value
Date	3225	datetime64[ns]		01/01/2017	15/12/2025	02/07/2021
Mine Tonnage_tons	3221	float64	tons	0.0000	4288.0000	2128.9354
Mine Grade_Au gpt	3221	float64	Au gpt	0.0000	11.0600	4.1335

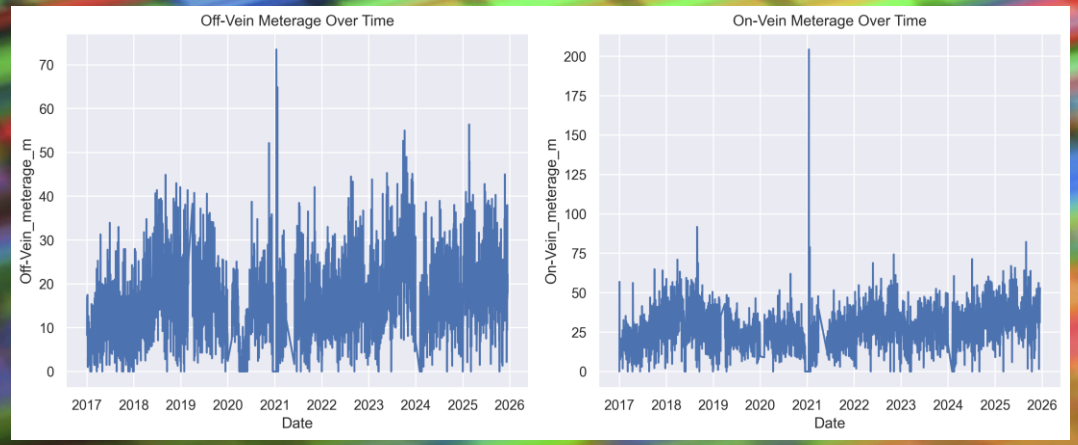
Mine Production Dataset is the official daily tally of Mine Production Data used as the measure of total output from the Mine Division. This dataset will become a source dataset for training the Mine Tonnage.



Time Series Analysis:
Tonnage has an **upward trend over time** because of increased capacity and process efficiencies within the mining company

Data Overview





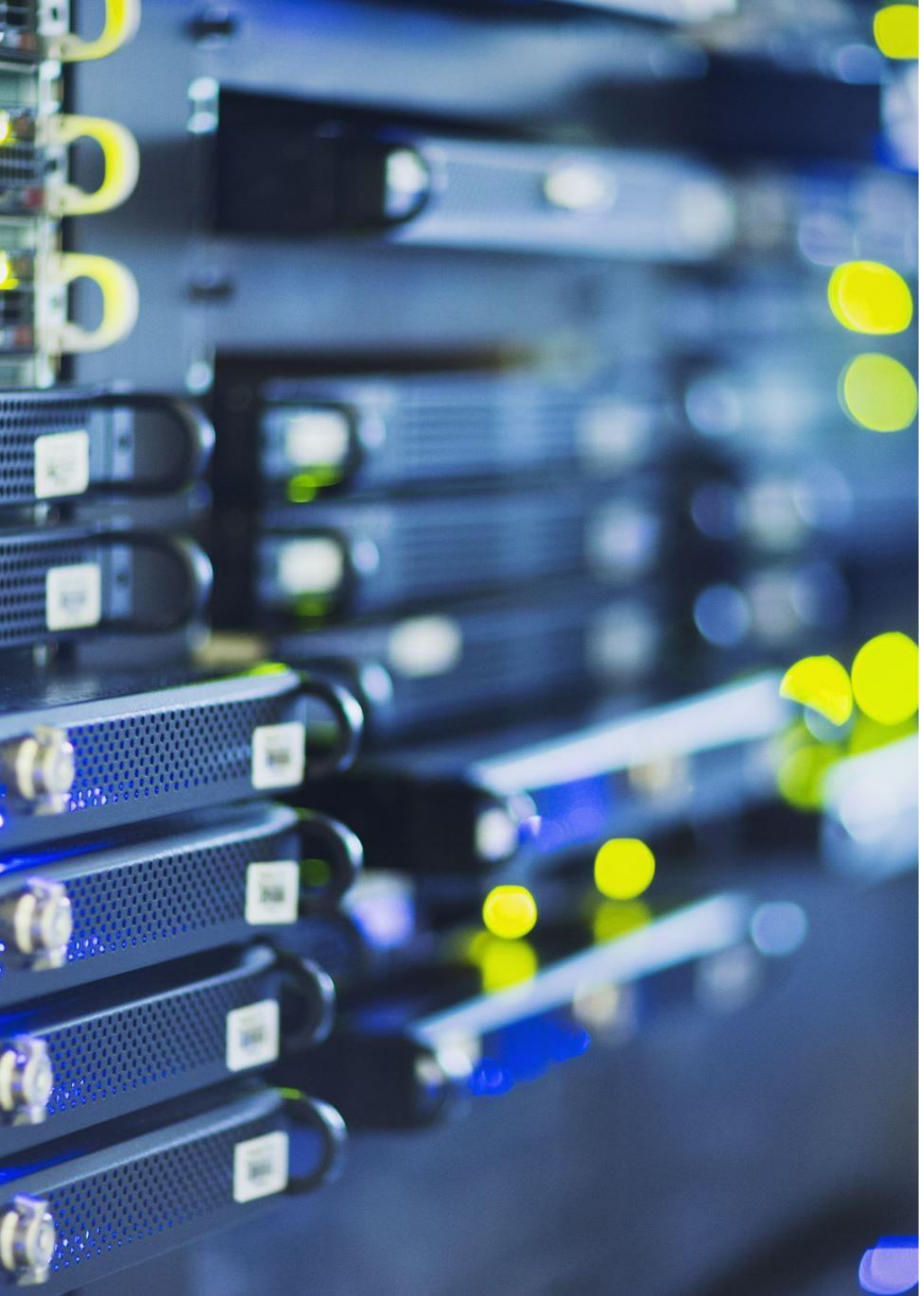
- **Dataset 4/4: df_mine_advance**
  - MCF, Mill_Reports, Mine_Production, **Mine_Advance**

Column Name	Non-Null Count	Data Type	Unit	Min Value	Max Value	Average Value
Date	3115	datetime64[ns]		01/01/2017	15/12/2025	29/06/2021
Off-Vein_meterage_m	3115	float64	m	0	73.5000	16.6925
On-Vein_meterage_m	3115	float64	m	0	204.3300	27.3853

Mine Advance Dataset is the official daily tally of Mine Development Data used as the measure of total development accomplishments from the Mine Division. This dataset will become a source dataset for training the Mine Tonnage.

On-vein meterage and Off-vein meterage seem to have a right-skewed normal distribution

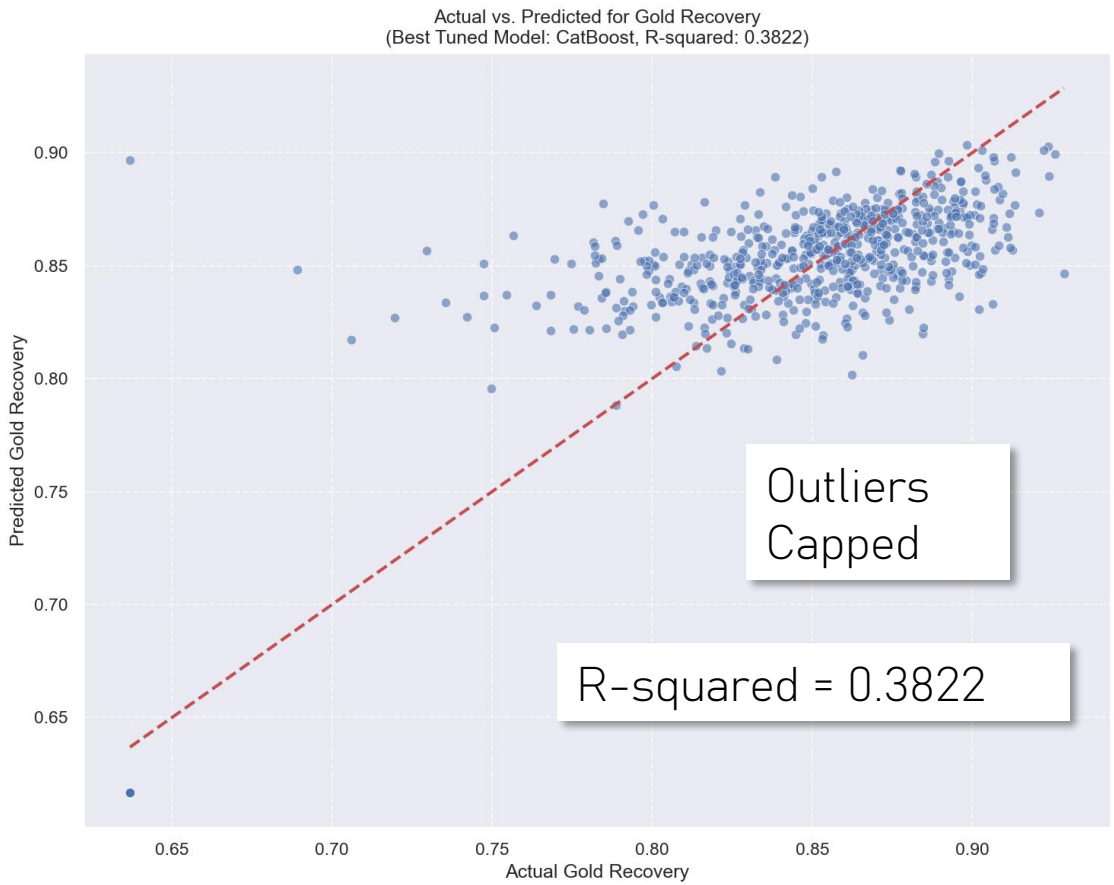
Time Series Analysis:
Meterage has a weak **upward trend over time** because of increased capacity and process efficiencies within the mining company

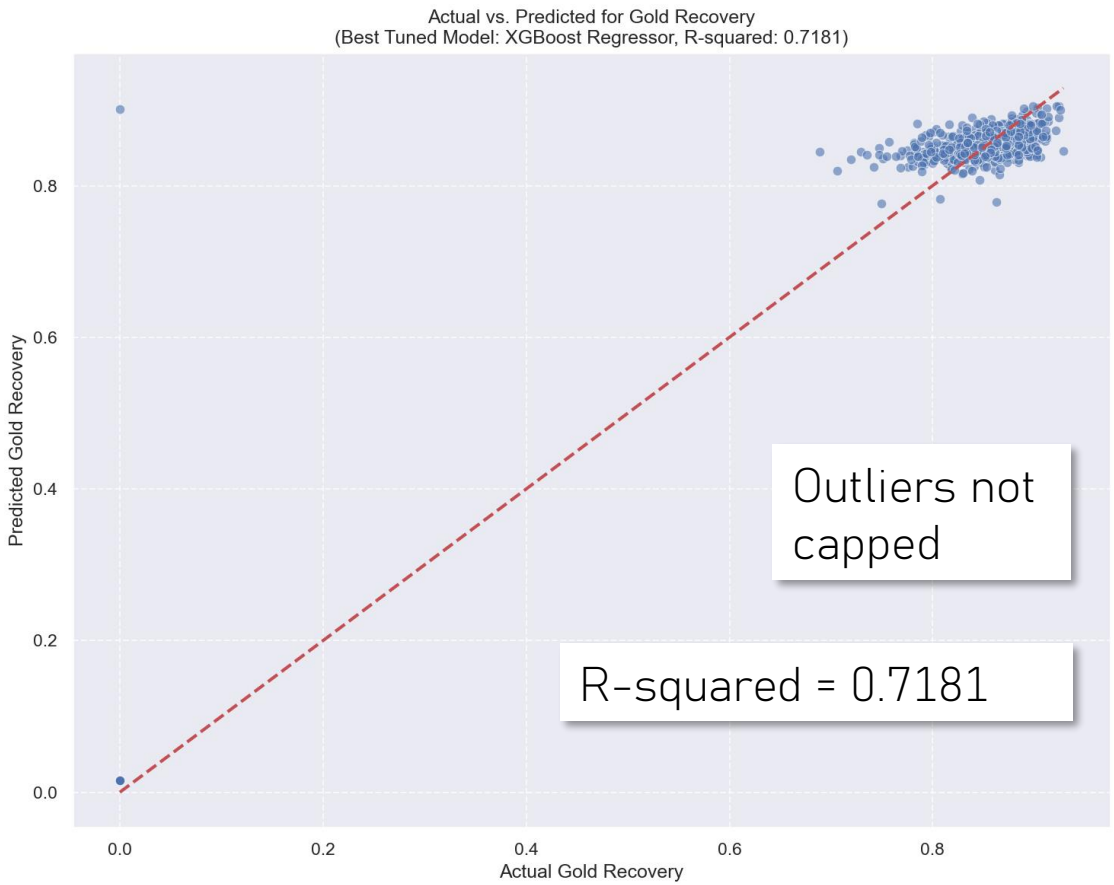


Data Cleaning

- The datasets were stored in centralized relational databases with strict schemas therefore they no long required any major data cleaning steps such as normalization of data entries.
- Handling of Null Values:
  - Null Values were dropped and no longer imputed since doing so will change the statistical characteristics of the dataset
- Handling of Outliers and Zero values
  - Outliers were not dropped or imputed
    - Outlier detection methods like Interquartile Range (IQR) or Z-score **will cut out necessary high-grade or high-tonnage datapoints and reduce the overall Feature space of the predictor variables**
  - Zero Values were also not dropped or imputed
    - Zero values are actual valid measurements
      - e.g. 0 tonnage = No operations; 0% recovery during 0 tonnage = No gold recovered since there was 0 input ore

Why didn't we cap or remove the 'Outliers'?





Capping the 'Outliers using Z-score method (std. dev >3) led to a 0.3359 decrease in R-squared for one of our tuned models

Scaling and Splitting of Data for Training/Testing

- The data was scaled using **StandardScaler** from sklearn.preprocessing library
- The data was split using **train_test_split** from sklearn.model_selection library
- For iterability, we placed the scaling and splitting **within a Pipeline** along with model selection through structured dictionaries (which make it much more human-readable)

```
# 5. We create and initialize an empty dictionary to store the training and testing sets
gold_silver_data_splits = {}

# 6. For each target variable in the dictionary:
for target_name, target_col in target_variables.items():
    # a. Create the feature set X
    X = df_gold_silver[predictor_cols]

    # b. Create the target set y
    y = df_gold_silver[target_col]

    # c. Split X and y into training and testing sets, we create an 80-20 split for Train and Test set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # d. Store these sets within the gold_silver_data_splits dictionary
    gold_silver_data_splits[target_name] = {
        'X_train': X_train,
        'X_test': X_test,
        'y_train': y_train,
        'y_test': y_test
    }
```

```
# create dictionary that will store the model performance
gold_ounces_model_performance = {}

# Get the data splits for 'Gold Ounces Produced'
X_train_go = gold_silver_data_splits['Gold Ounces Produced']['X_train']
X_test_go = gold_silver_data_splits['Gold Ounces Produced']['X_test']
y_train_go = gold_silver_data_splits['Gold Ounces Produced']['y_train']
y_test_go = gold_silver_data_splits['Gold Ounces Produced']['y_test']

# Recycle the models dictionary defined previously
# models = {...} # (Assuming it's still in memory)

for name, model in models.items():
    print(f"\nTraining and evaluating {name} for Gold Ounces Produced...")

    # Create a pipeline with StandardScaler and the model
    pipeline = Pipeline([
        ('scaler', StandardScaler()),
        ('regressor', model)
    ])

    # Fit the pipeline
    pipeline.fit(X_train_go, y_train_go)
```

Baseline Model Creation

- Models and Baseline Parameters are **stored in a Dictionary** so that it would be much easier to iterate through all the models.

```python
# create dictionary that will store the model, model library and parameters
# we use initial parameters that will eventually be tuned during hyperparameter tuning later
models = {
    'Linear Regression': LinearRegression(),
    'XGBoost Regressor': XGBRegressor(random_state=42, n_estimators=100, learning_rate=0.1, max_depth=3),
    'LightGBM': LGBMRegressor(random_state=42, n_estimators=100, learning_rate=0.1, max_depth=5),
    'CatBoost': CatBoostRegressor(random_state=42, verbose=0, iterations=100, learning_rate=0.1, depth=6),
    'Decision Tree Regressor': DecisionTreeRegressor(random_state=42),
    'Support Vector Machine': SVR(kernel='rbf'), # RBF kernel is a good general choice
    'Artificial Neural Network': MLPRegressor(random_state=42, max_iter=500, hidden_layer_sizes=(100, 50), solver='adam', activation='relu', early_stopping=True, n_iter_no_change=50)
}

# to iterate through all model, we create for loop that will iterate through the dictionary we just created, this
for name, model in models.items():
    print(f"\nTraining and evaluating {name}...")

    # Create a pipeline with StandardScaler and the model
    pipeline = Pipeline([
        ('scaler', StandardScaler()),
        ('regressor', model)
    ])

    # Fit the pipeline
    pipeline.fit(X_train_gr, y_train_gr)

    # Make predictions
    y_pred = pipeline.predict(X_test_gr)

    # Calculate metrics
    r2 = r2_score(y_test_gr, y_pred)
    mae = mean_absolute_error(y_test_gr, y_pred)
    mse = mean_squared_error(y_test_gr, y_pred)
    rmse = np.sqrt(mse)

    # Store metrics
    gold_recovery_model_performance[name] = {
        'R-squared': r2,
        'MAE': mae,
        'MSE': mse,
        'RMSE': rmse
    }

    print(f"{name} - R-squared: {r2:.4f}, MAE: {mae:.4f}, MSE: {mse:.4f}, RMSE: {rmse:.4f}")
```

Model Parameter Dictionary

Baseline Model Creation

- The For-loop will iterate through the item pairs of the dictionary which includes the scaling
- The model performance is stored in another dictionary so that it would be easier to recall it for tabulation later.

```python
# create dictionary that will store the model, model library and parameters
# we use initial parameters that will eventually be tuned during hyperparameter tuning later
models = {
    'Linear Regression': LinearRegression(),
    'XGBoost Regressor': XGBRegressor(random_state=42, n_estimators=100, learning_rate=0.1, max_depth=3),
    'LightGBM': LGBMRegressor(random_state=42, n_estimators=100, learning_rate=0.1, max_depth=5),
    'CatBoost': CatBoostRegressor(random_state=42, verbose=0, iterations=100, learning_rate=0.1, depth=6),
    'Decision Tree Regressor': DecisionTreeRegressor(random_state=42),
    'Support Vector Machine': SVR(kernel='rbf'), # RBF kernel is a good general choice
    'Artificial Neural Network': MLPRegressor(random_state=42, max_iter=500, hidden_layer_sizes=(100, 50), solver='adam', activation='relu', early_stopping=True, n_iter_no_change=50)
}

# to iterate through all model, we create for loop that will iterate through the dictionary we just created, this will make it easier to do for the other prediction
for name, model in models.items():
    print(f"\nTraining and evaluating {name}...")

    # Create a pipeline with StandardScaler and the model
    pipeline = Pipeline([
        ('scaler', StandardScaler()),
        ('regressor', model)
    ])

    # Fit the pipeline
    pipeline.fit(X_train_gr, y_train_gr)

    # Make predictions
    y_pred = pipeline.predict(X_test_gr)

    # Calculate metrics
    r2 = r2_score(y_test_gr, y_pred)
    mae = mean_absolute_error(y_test_gr, y_pred)
    mse = mean_squared_error(y_test_gr, y_pred)
    rmse = np.sqrt(mse)

    # Store metrics
    gold_recovery_model_performance[name] = {
        'R-squared': r2,
        'MAE': mae,
        'MSE': mse,
        'RMSE': rmse
    }

    print(f"{name} - R-squared: {r2:.4f}, MAE: {mae:.4f}, MSE: {mse:.4f}, RMSE: {rmse:.4f}")
```

For-loop iterator with pipeline

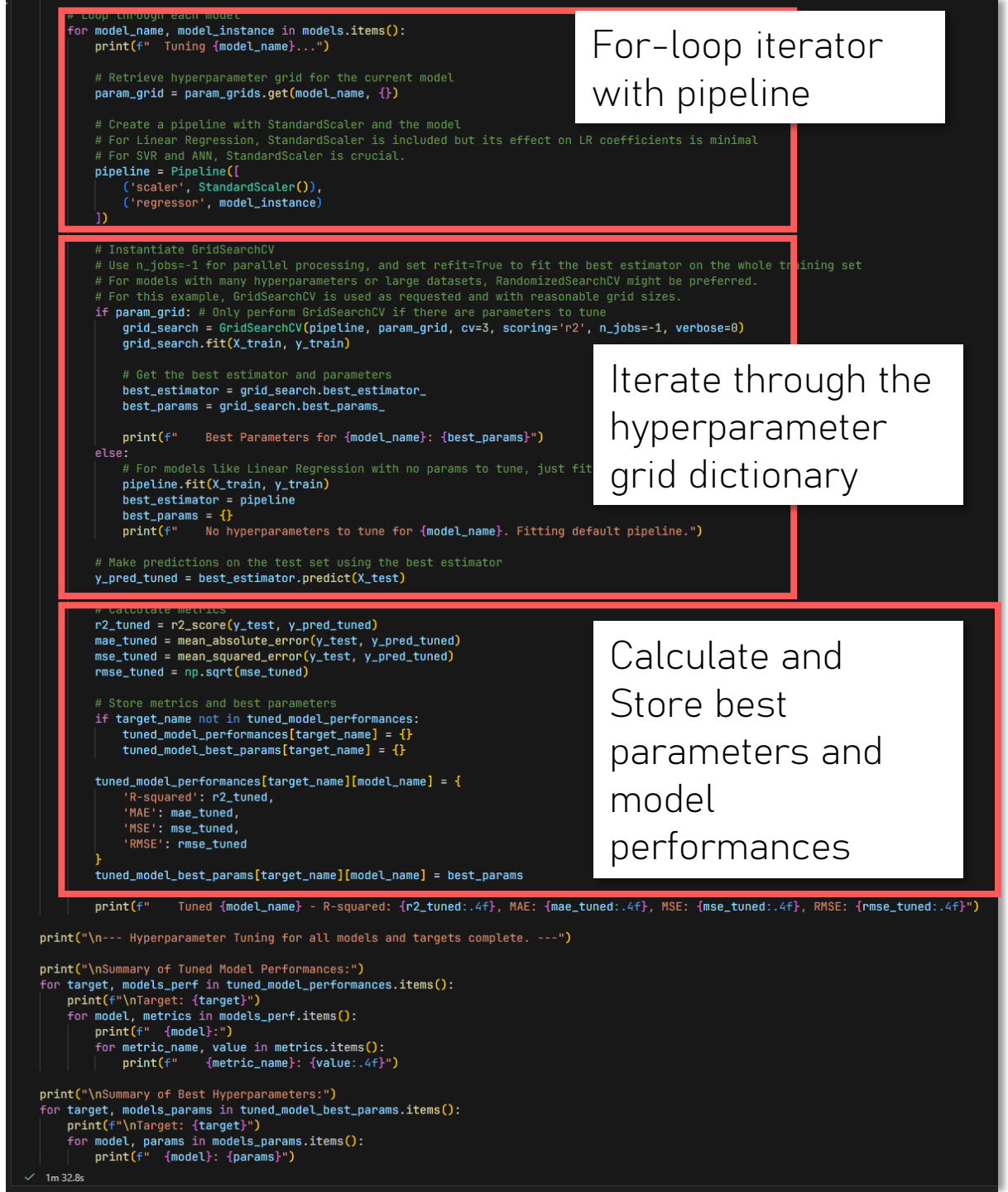
Storing model performance in a dictionary

Hyperparameter Tuning

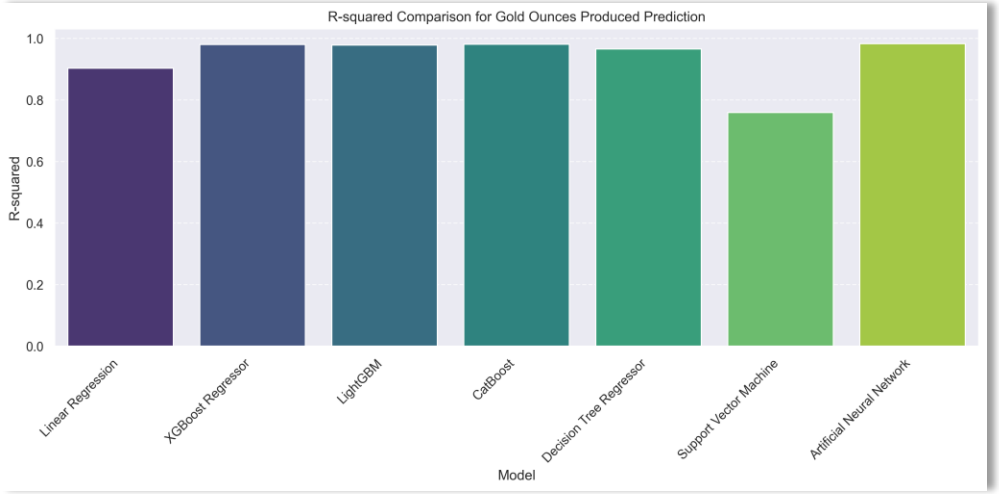
- A **hyperparameter grid** was created and stored inside a dictionary so that it will be much easier to tweak the ranges and for it to be easier to iterate over, using a For loop.

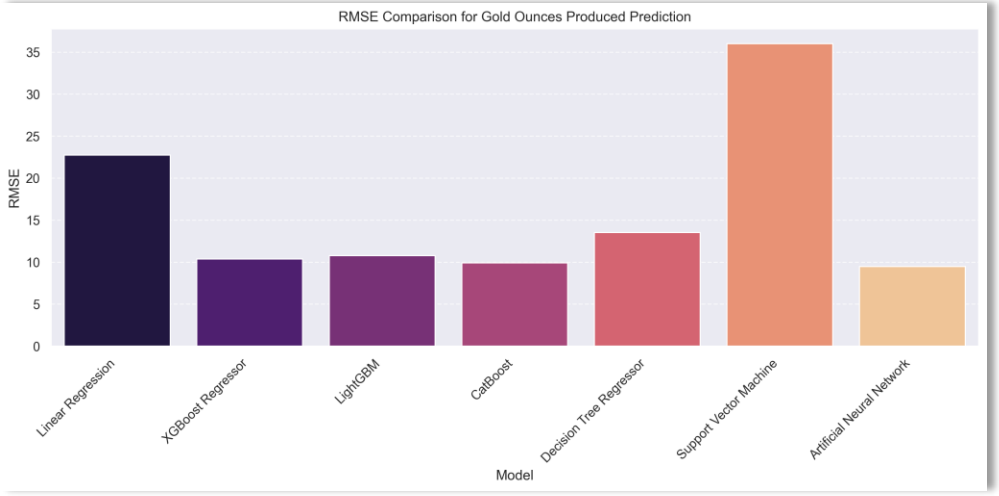
Hyperparameter Tuning

- The For loop also includes StandardScaler similar to the baseline model and will iterate through the items within the model dictionary and hyperparameter grid.
- Afterwards, the results of the tuning results and best parameters are stored in a dictionary so that it can be referred to later for plotting of tables and charts.

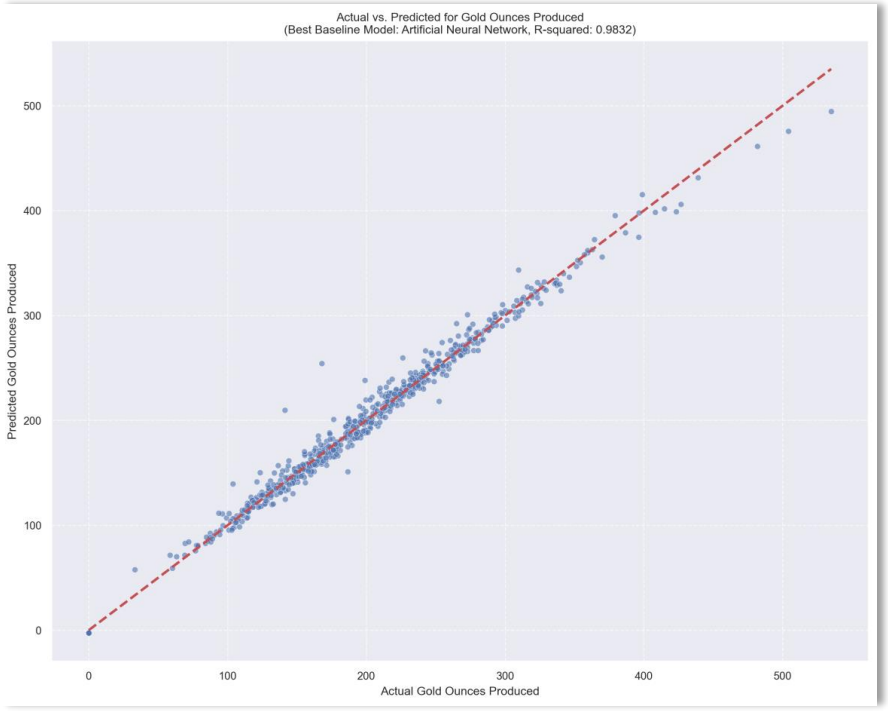


Baseline Model Results: Gold Ounces





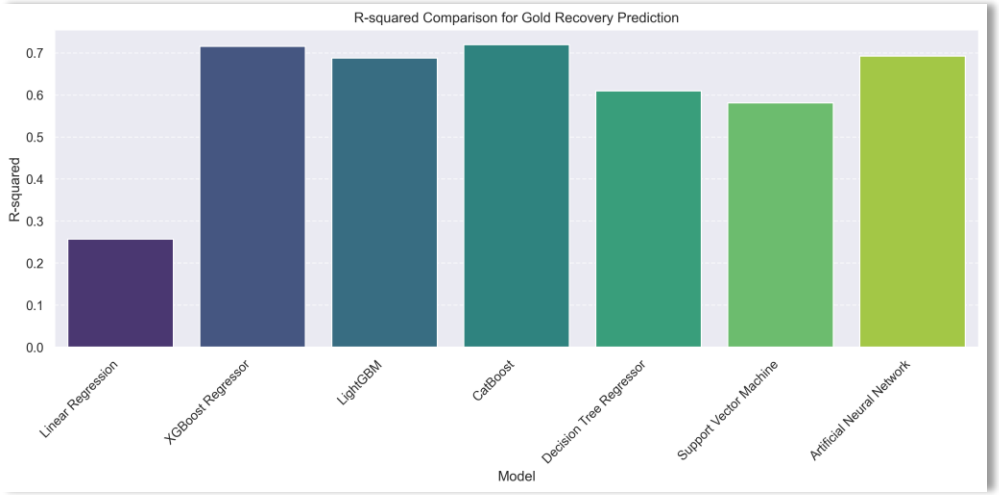
Target Variable	Model	R-squared	MAE	MSE	RMSE
Gold Ounces Produced	***Artificial Neural Network***	**0.983**	**6.297**	**90.243**	**9.5**
Gold Ounces Produced	*CatBoost*	0.982	6.483	98.684	9.934
Gold Ounces Produced	*XGBoost Regressor*	0.98	6.827	108.048	10.395
Gold Ounces Produced	*LightGBM*	0.978	6.886	116.381	10.788
Gold Ounces Produced	*Decision Tree Regressor*	0.966	9.594	182.872	13.523
Gold Ounces Produced	*Linear Regression*	0.904	12.75	517.838	22.756
Gold Ounces Produced	*Support Vector Machine*	0.759	15.035	1297.479	36.021

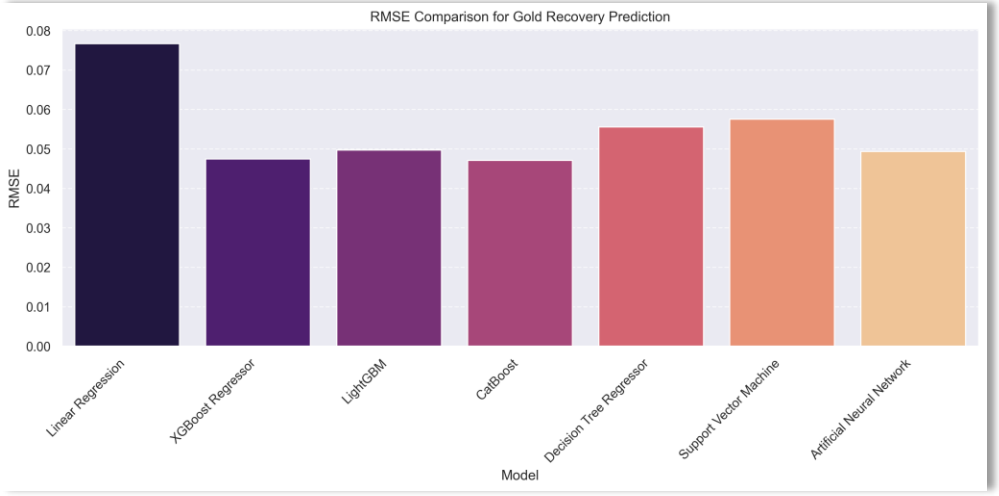


Best Performer:
Artificial Neural Network
(R-squared: 0.983; RMSE: 9.5)

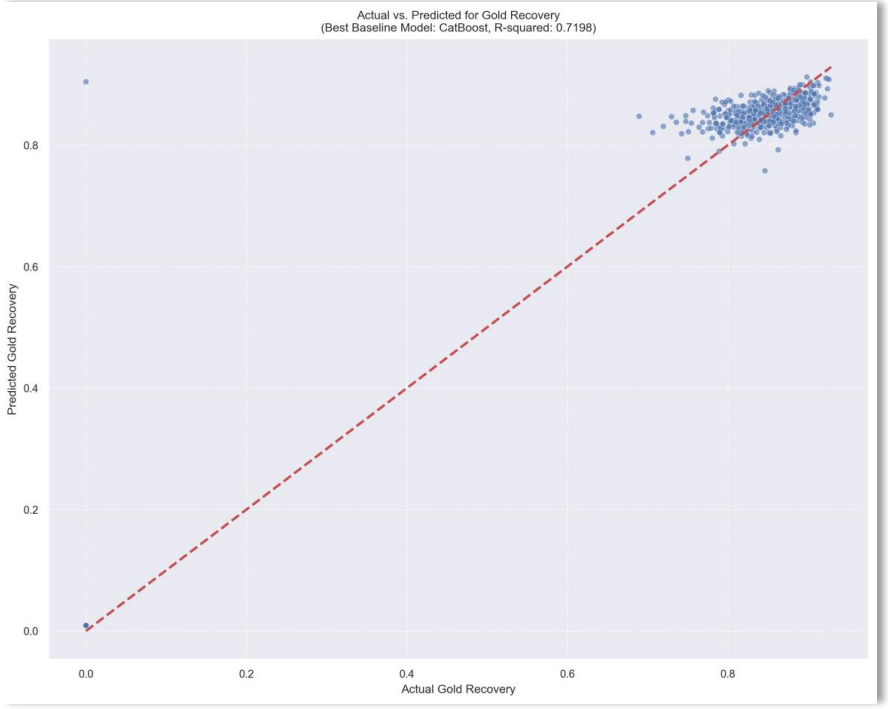
Worst Performer:
Support Vector Regression
(R-squared: 0.759; RMSE: 36.021)

Baseline Model Results: Gold Recovery





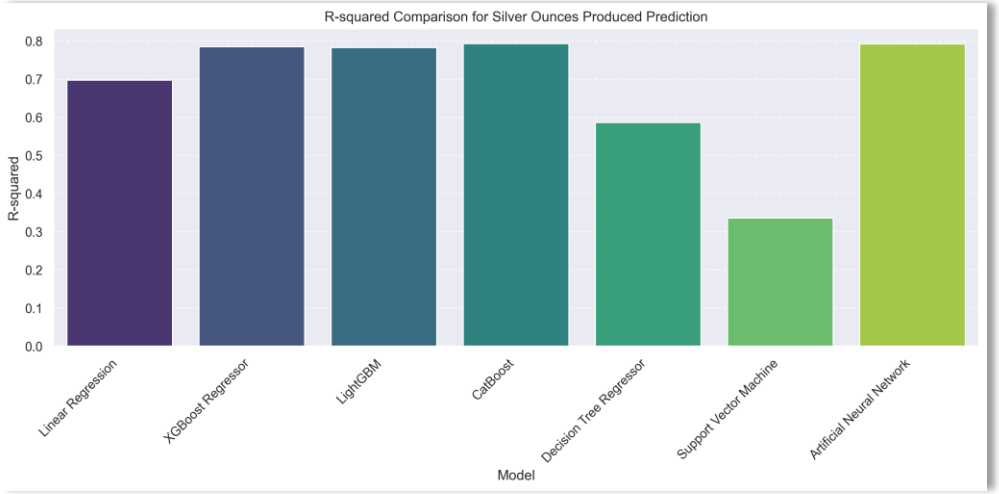
Target Variable	Model	R-squared	MAE	MSE	RMSE
Gold Recovery	***CatBoost***	**0.72**	**0.024**	**0.002**	**0.047**
Gold Recovery	*XGBoost Regressor*	0.716	0.025	0.002	0.047
Gold Recovery	*Artificial Neural Network*	0.692	0.026	0.002	0.049
Gold Recovery	*LightGBM*	0.688	0.027	0.002	0.05
Gold Recovery	*Decision Tree Regressor*	0.61	0.035	0.003	0.056
Gold Recovery	*Support Vector Machine*	0.581	0.042	0.003	0.058
Gold Recovery	*Linear Regression*	0.257	0.035	0.006	0.077

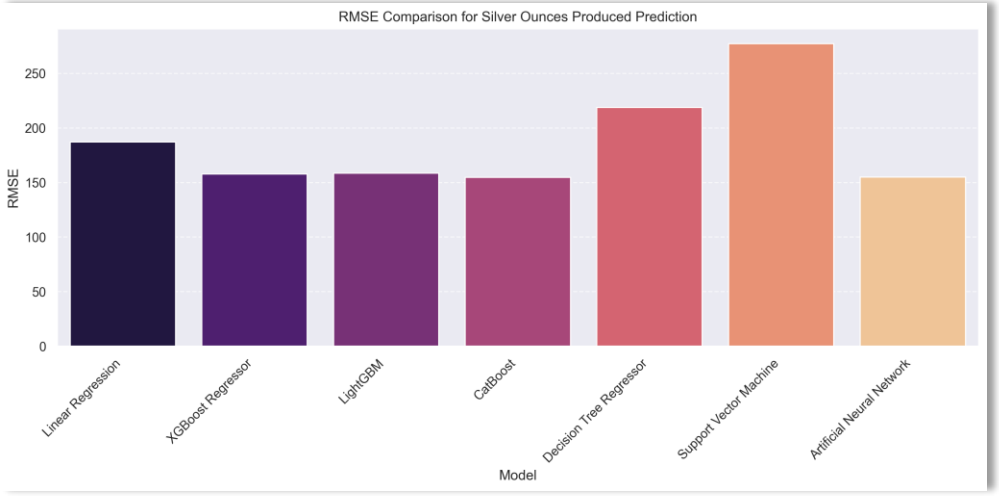


Best Performer:
CatBoost
(R-squared: 0.72; RMSE: 0.047)

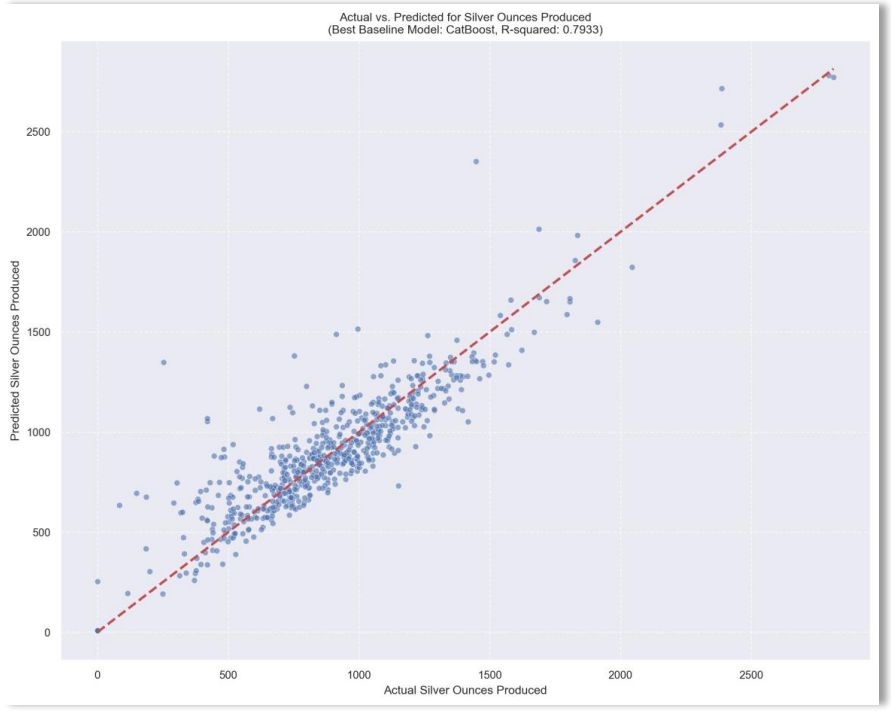
Worst Performer:
Linear Regression
(R-squared: 0.257; RMSE: 0.047)

Baseline Model Results: Silver Ounces





Target Variable	Model	R-squared	MAE	MSE	RMSE
Silver Ounces Produced	***CatBoost***	**0.793**	**108.078**	**23940.979**	**154.729**
Silver Ounces Produced	*Artificial Neural Network*	0.793	108.903	24021.369	154.988
Silver Ounces Produced	*XGBoost Regressor*	0.785	110.09	24896.291	157.786
Silver Ounces Produced	*LightGBM*	0.783	111.181	25161.088	158.622
Silver Ounces Produced	*Linear Regression*	0.698	126.973	35001.914	187.088
Silver Ounces Produced	*Decision Tree Regressor*	0.587	154.653	47853.135	218.754
Silver Ounces Produced	*Support Vector Machine*	0.336	182.81	76904.071	277.316



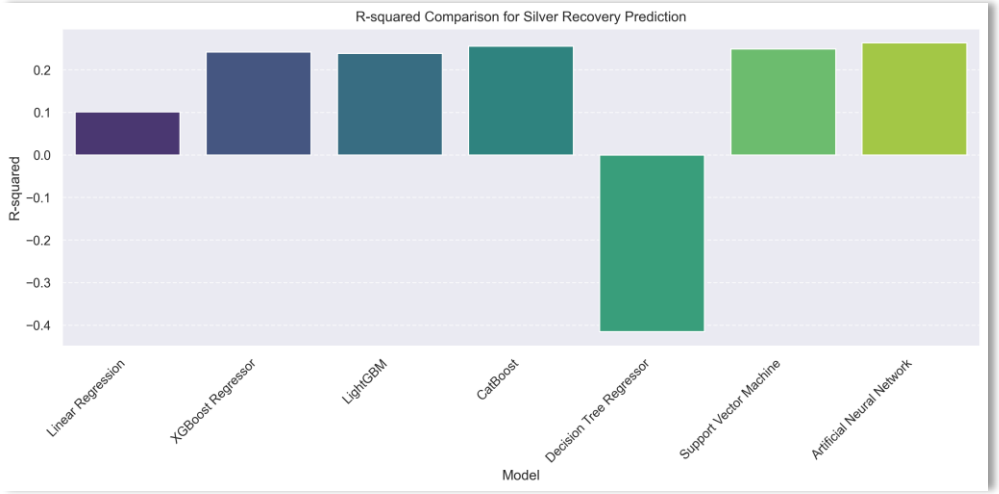
Best Performer:

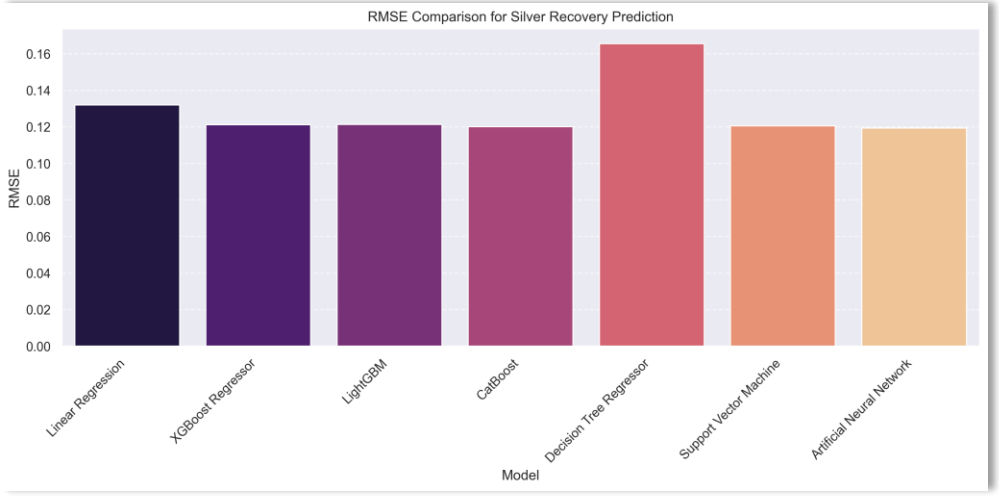
CatBoost
(R-squared: 0.793; RMSE: 154.729)

Worst Performer:

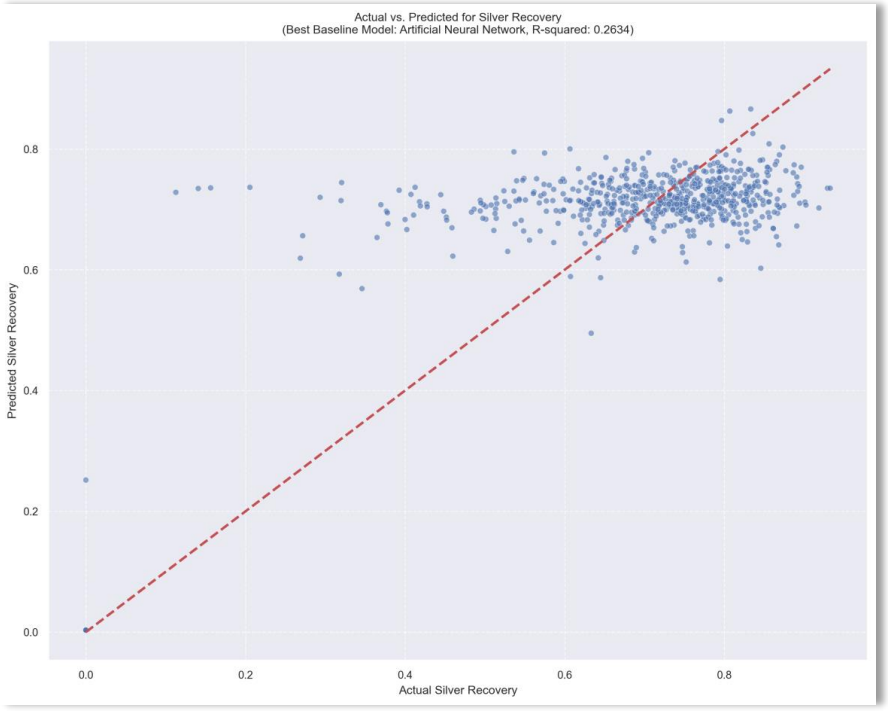
Support Vector Regression
(R-squared: 0.336; RMSE: 277.316)

Baseline Model Results: Silver Recovery





Target Variable	Model	R-squared	MAE	MSE	RMSE
Silver Recovery	***Artificial Neural Network***	**0.263**	**0.088**	**0.014**	**0.119**
Silver Recovery	*CatBoost*	0.256	0.086	0.014	0.12
Silver Recovery	*Support Vector Machine*	0.249	0.088	0.015	0.121
Silver Recovery	*XGBoost Regressor*	0.242	0.087	0.015	0.121
Silver Recovery	*LightGBM*	0.239	0.088	0.015	0.121
Silver Recovery	*Linear Regression*	0.101	0.095	0.017	0.132
Silver Recovery	*Decision Tree Regressor*	-0.416	0.123	0.027	0.166



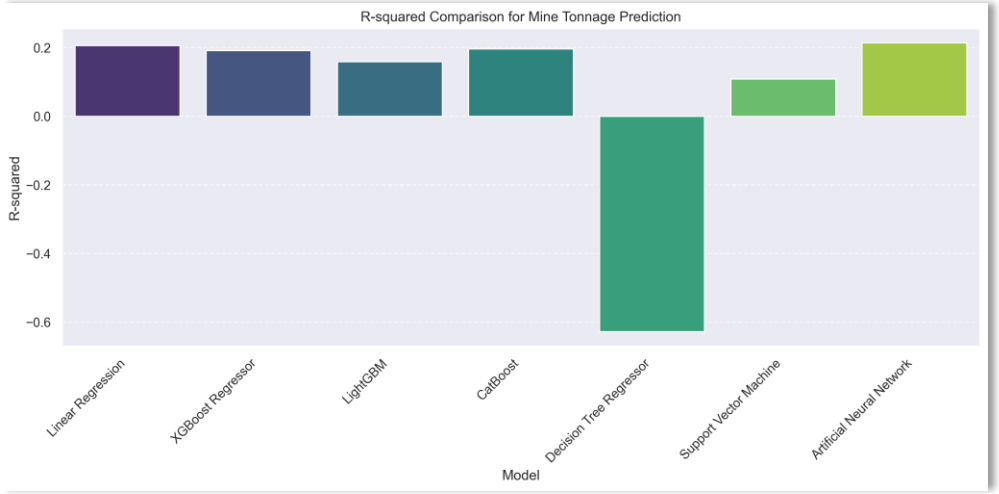
Best Performer:

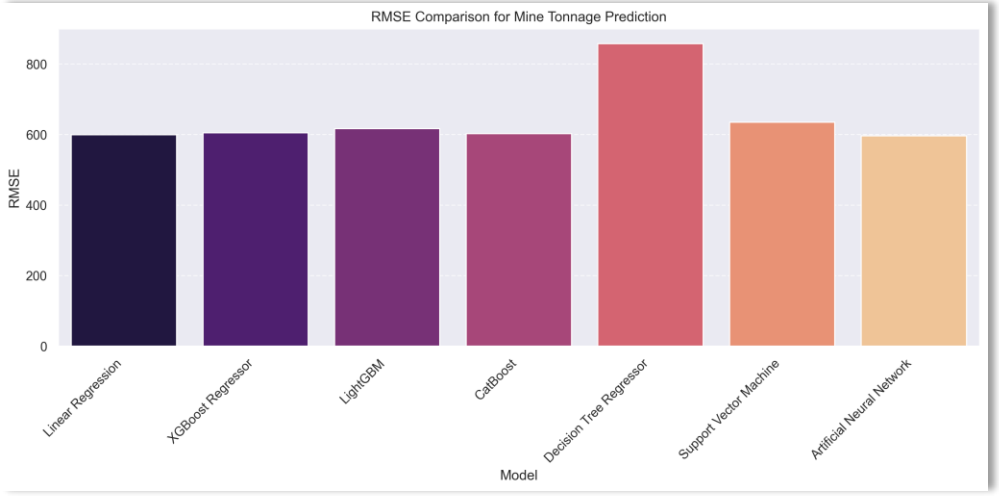
Artificial Neural Network
(R-squared: 0.263; RMSE: 0.119)

Worst Performer:

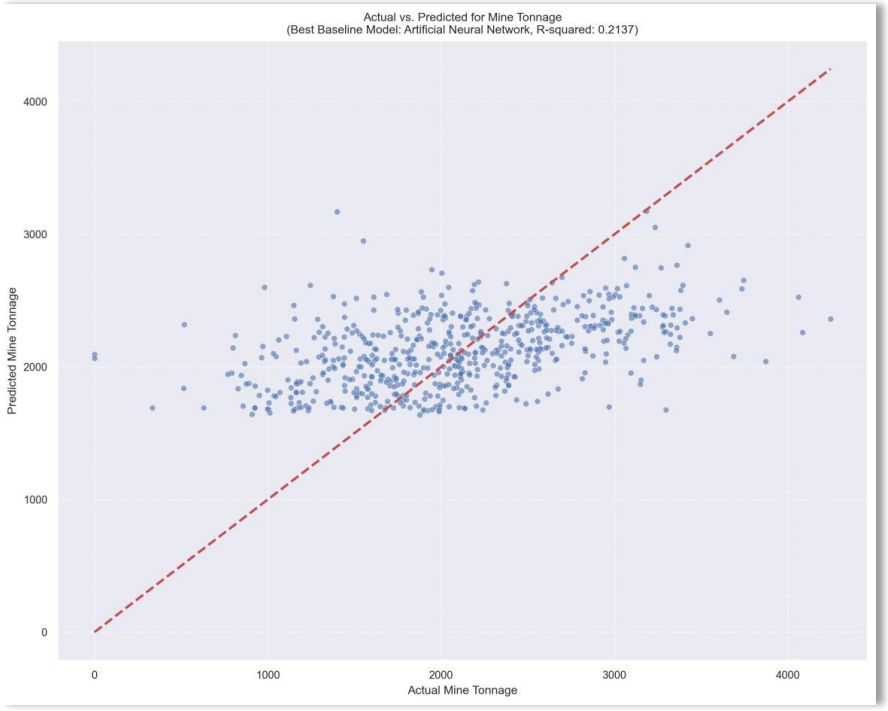
Decision Tree Regressor
(R-squared: -0.416; RMSE: 0.166)

Baseline Model Results: Mine Tonnage





Target Variable	Model	R-squared	MAE	MSE	RMSE
Mine Tonnage	***Artificial Neural Network***	**0.214**	**471.098**	**355453.886**	**596.2**
Mine Tonnage	*Linear Regression*	0.206	474.077	358928.383	599.106
Mine Tonnage	*CatBoost*	0.196	476.798	363232.126	602.687
Mine Tonnage	*XGBoost Regressor*	0.191	478.296	365673.128	604.709
Mine Tonnage	*LightGBM*	0.159	486.136	380103.423	616.525
Mine Tonnage	*Support Vector Machine*	0.108	503.697	403181.883	634.966
Mine Tonnage	*Decision Tree Regressor*	-0.628	690.725	735900.724	857.847



Best Performer:
Artificial Neural Network
(R-squared: 0.214; RMSE: 596.2)

Worst Performer:
Decision Tree Regressor
(R-squared: -628; RMSE: 857.847)

Baseline Model Results: Summary

Target Variable	Model	Rank	R-squared	MAE	MSE	RMSE
Gold Ounces Produced	*Artificial Neural Network*	Best Performer	0.983	6.297	90.243	9.5
Gold Ounces Produced	*Support Vector Machine*	Worst Performer	0.759	15.035	1297.479	36.021
Gold Recovery	*CatBoost*	Best Performer	0.72	0.024	0.002	0.047
Gold Recovery	*Linear Regression*	Worst Performer	0.257	0.035	0.006	0.077
Silver Ounces Produced	*CatBoost*	Best Performer	0.793	108.078	23940.979	154.729
Silver Ounces Produced	*Support Vector Machine*	Worst Performer	0.336	182.81	76904.071	277.316
Silver Recovery	*Artificial Neural Network*	Best Performer	0.263	0.088	0.014	0.119
Silver Recovery	*Decision Tree Regressor*	Worst Performer	-0.416	0.123	0.027	0.166
Mine Tonnage	*Artificial Neural Network*	Best Performer	0.214	471.098	355453.886	596.2
Mine Tonnage	*Decision Tree Regressor*	Worst Performer	-0.628	690.725	735900.724	857.847

Best Performers:

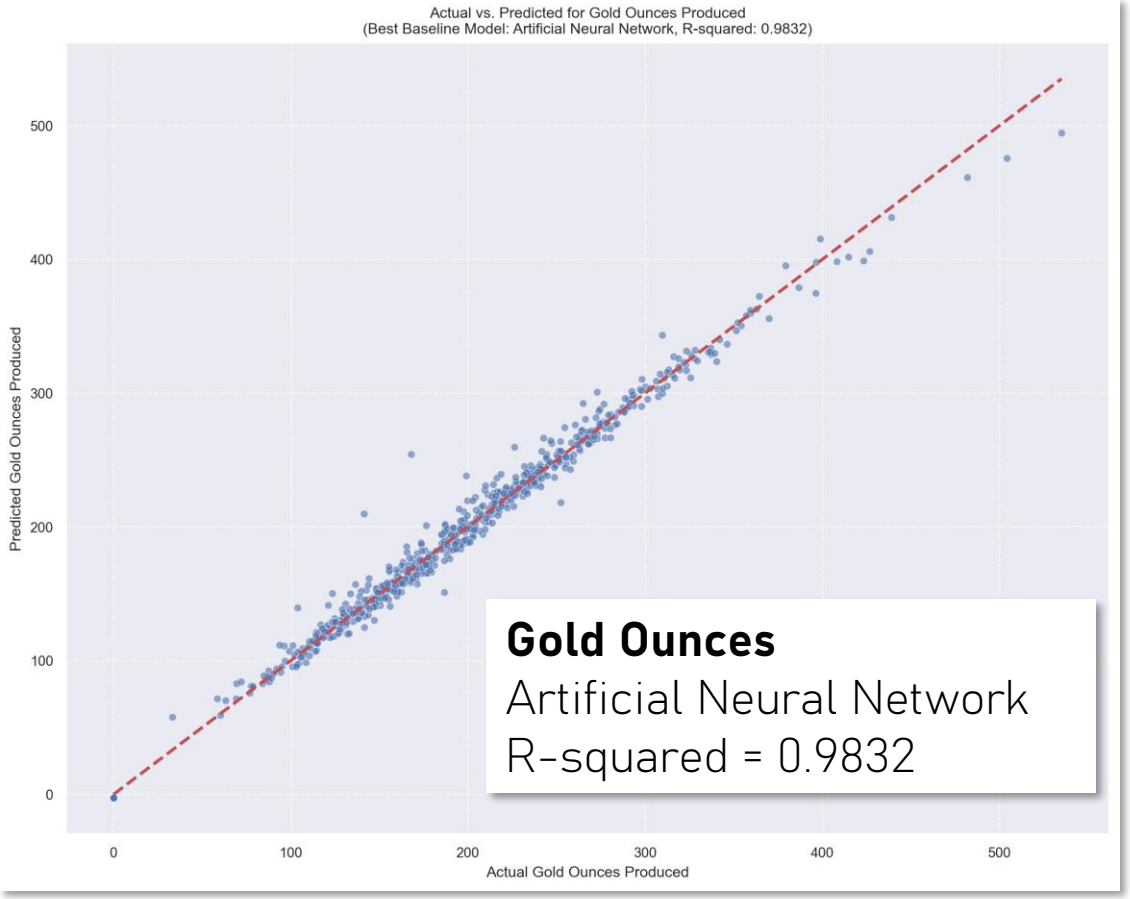
The best performing baseline models are mostly Artificial Neural Networks (3/5) followed by CatBoost (2/5)

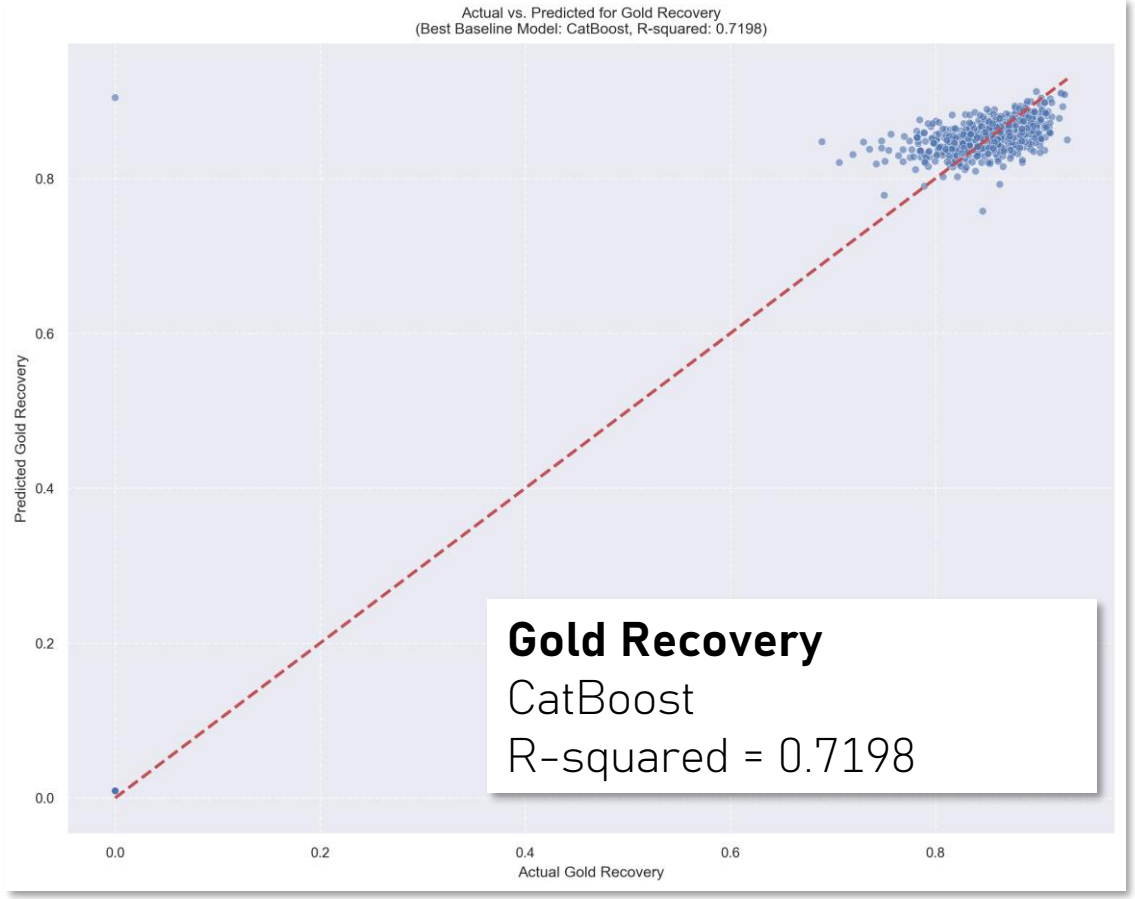
The good performance of the ANN may be because it performs well even to non-linear relationships.

Worst Performers:

The worst performing networks comprised of Support Vector Machine, Linear Regression, and Decision Tree Regressor with the Decision Tree Regressor obtaining R-squared values that are worse than the rest due to its negative values.

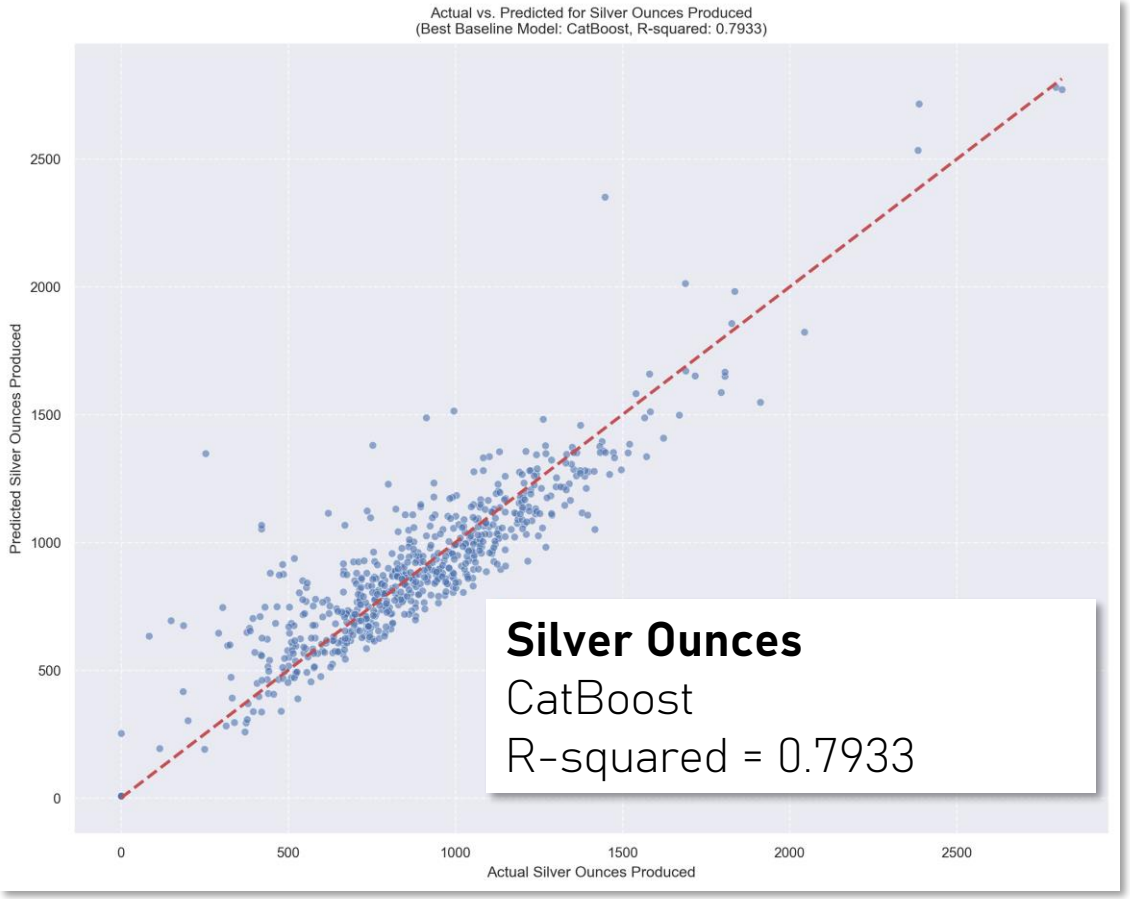
Baseline Model Results: Gold Output and Recovery

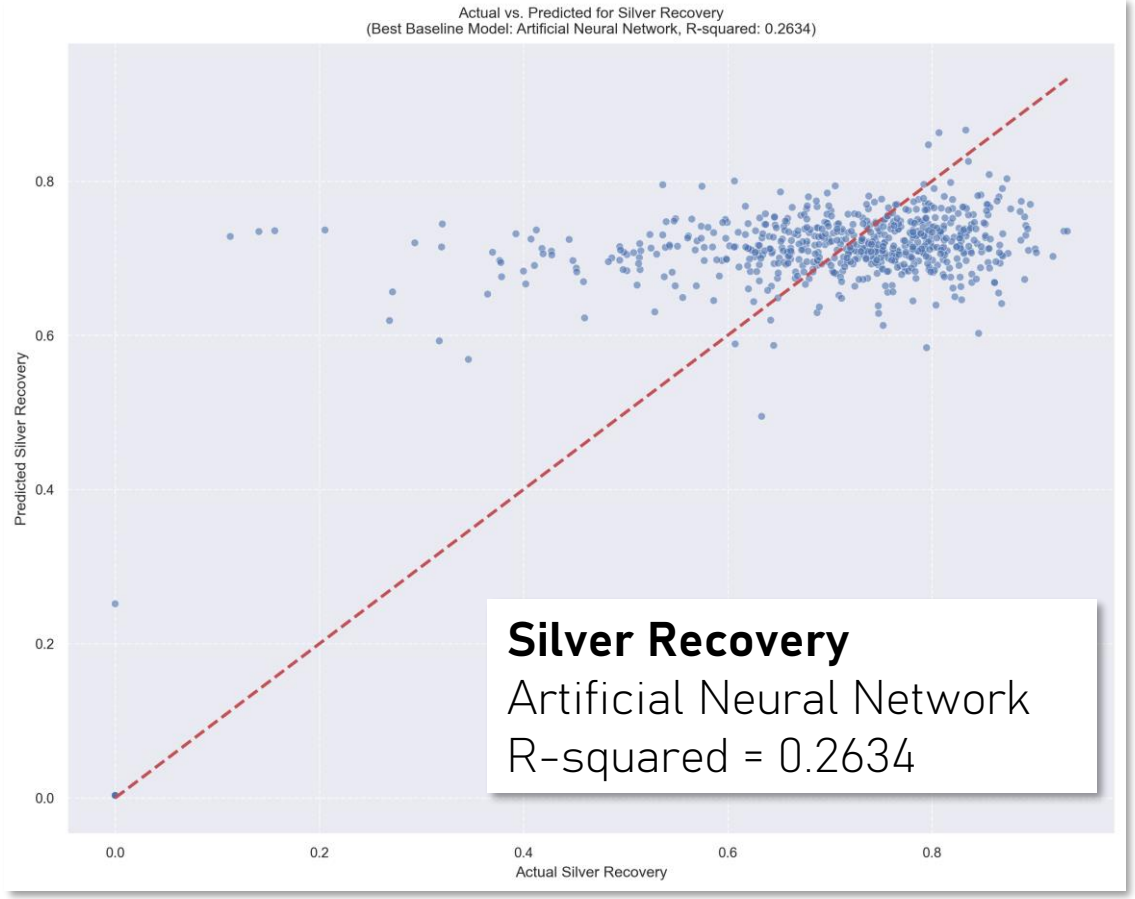




Gold Ounces seems to perform well even prior to hyperparameter tuning, this may entail that the performance after parameter tuning may only improve very slightly. **Gold Recovery** also has performed surprisingly well.

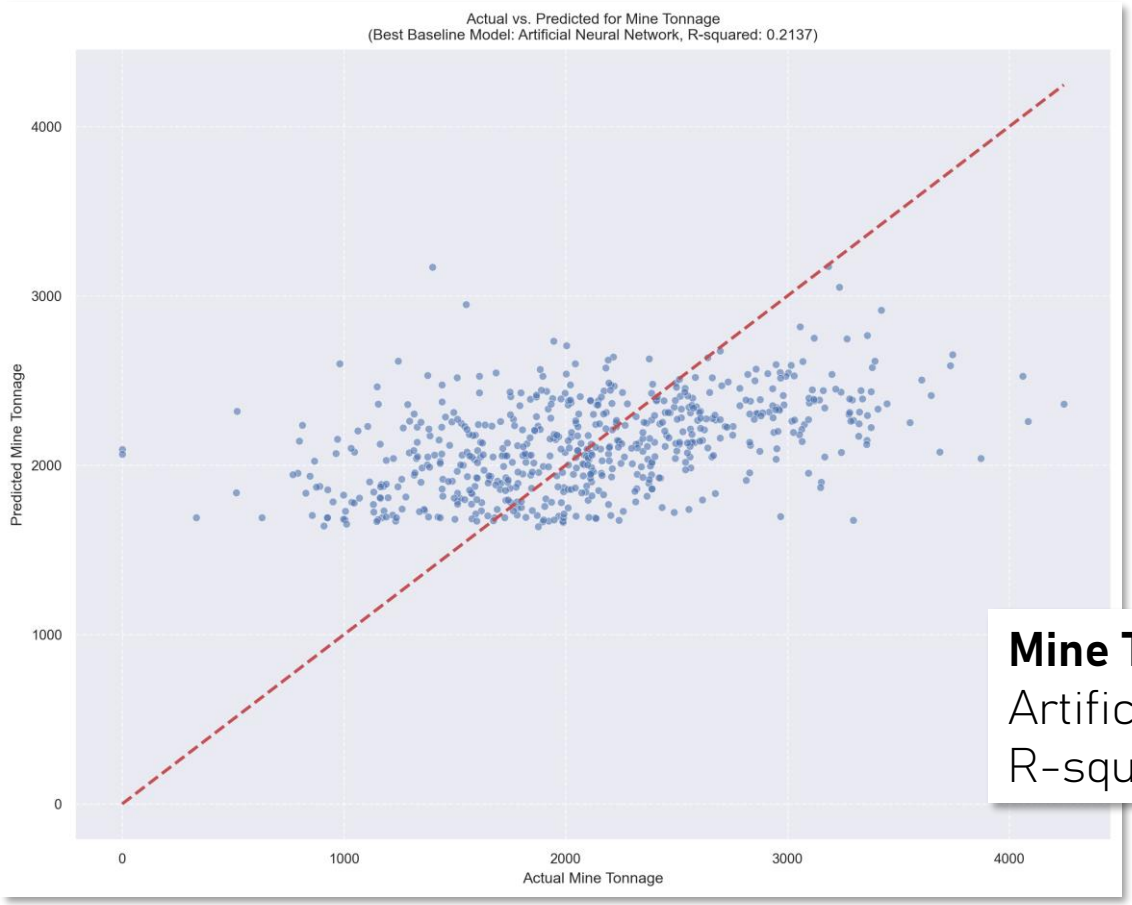
Baseline Model Results: Silver Output and Recovery





Silver Ounces seems to perform also relatively well prior to hyperparameter tuning. **Silver Recovery** has a much lower performance compared to Gold.

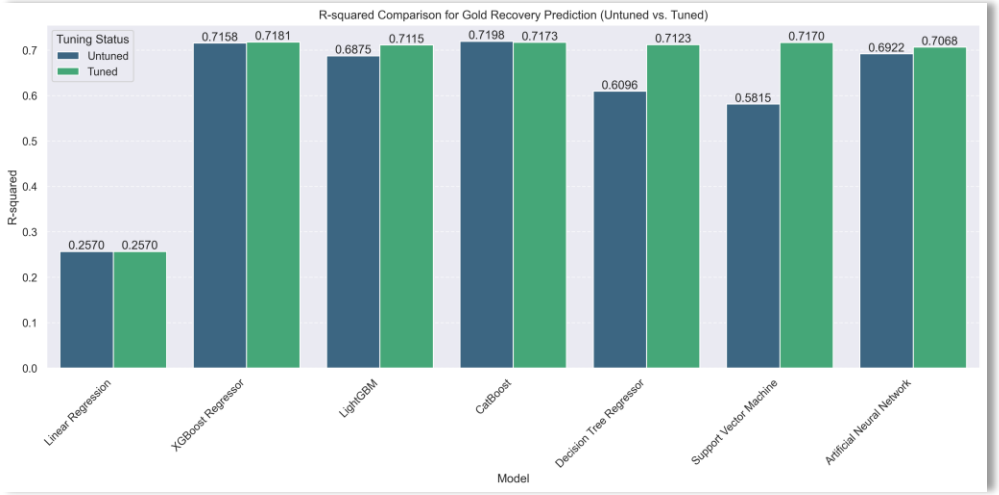
Baseline Model Results: Mine Tonnage

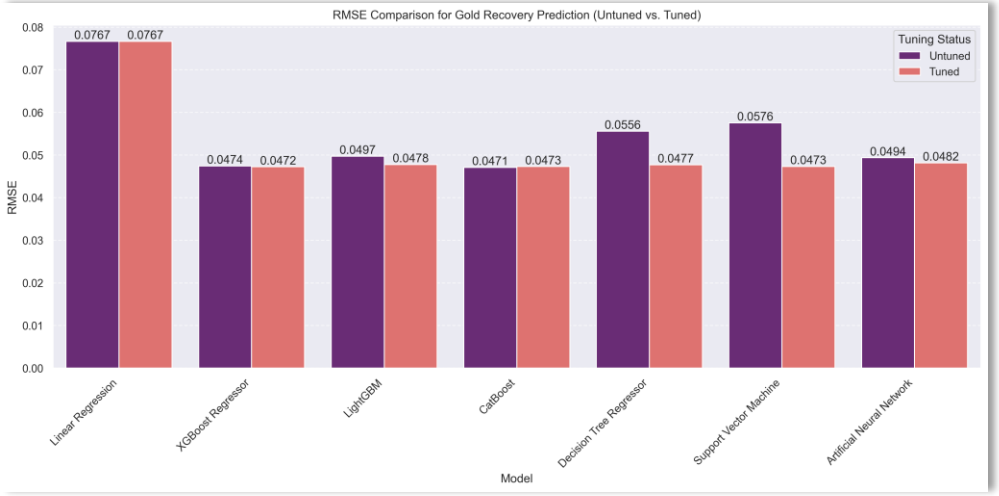


Mine Tonnage
Artificial Neural Network
R-squared = 0.2137

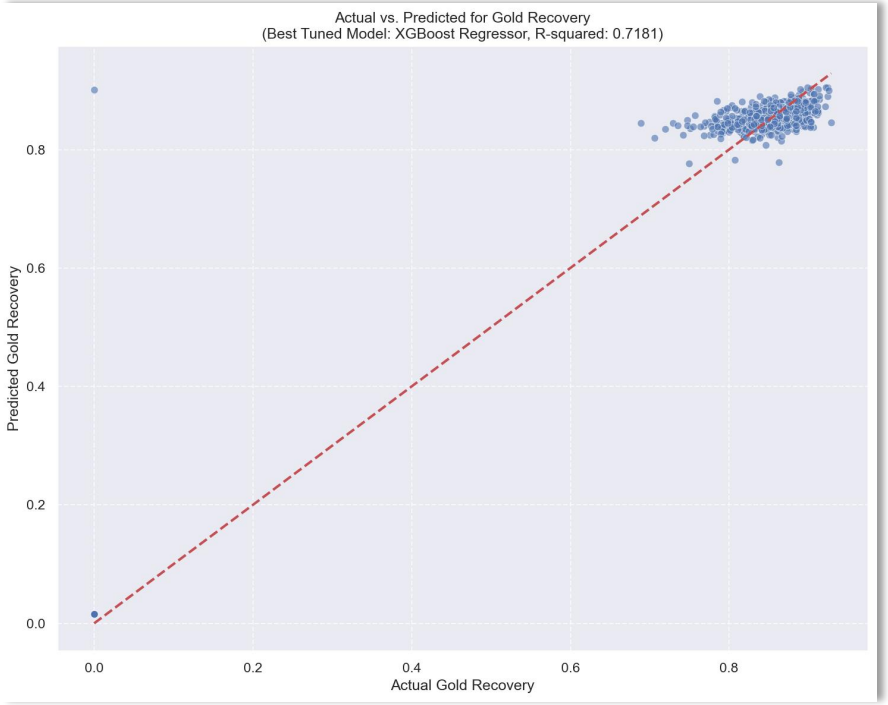
Mine Tonnage, similar to Silver recovery has a much lower performance compared to Gold Recovery and Throughput, this may improve after hyperparameter tuning.

Hyperparameter Tuned Model Results: Gold Recovery





Target	Model	R-squared	MAE	MSE	RMSE	Tuning Status
Gold Recovery	**CatBoost**	**0.7198**	**0.0245**	**0.0022**	**0.0471**	**Untuned**
Gold Recovery	XGBoost Regressor	0.7181	0.0250	0.0022	0.0472	Tuned
Gold Recovery	CatBoost	0.7173	0.0248	0.0022	0.0473	Tuned
Gold Recovery	Support Vector Machine	0.7170	0.0251	0.0022	0.0473	Tuned
Gold Recovery	XGBoost Regressor	0.7158	0.0249	0.0022	0.0474	Untuned
Gold Recovery	Decision Tree Regressor	0.7123	0.0256	0.0023	0.0477	Tuned



Best Tuned Performer:

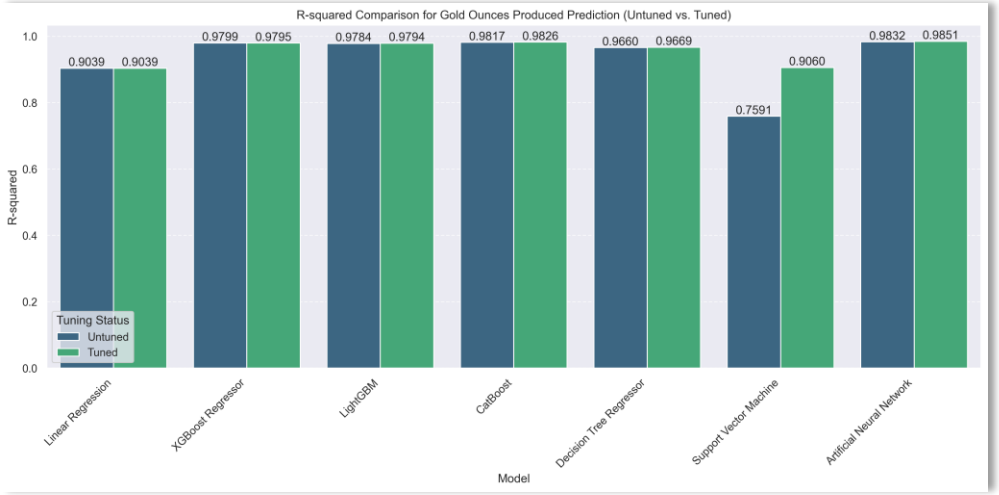
XGBoost Regressor
(R-squared: 0.7181; RMSE: 0.0472)

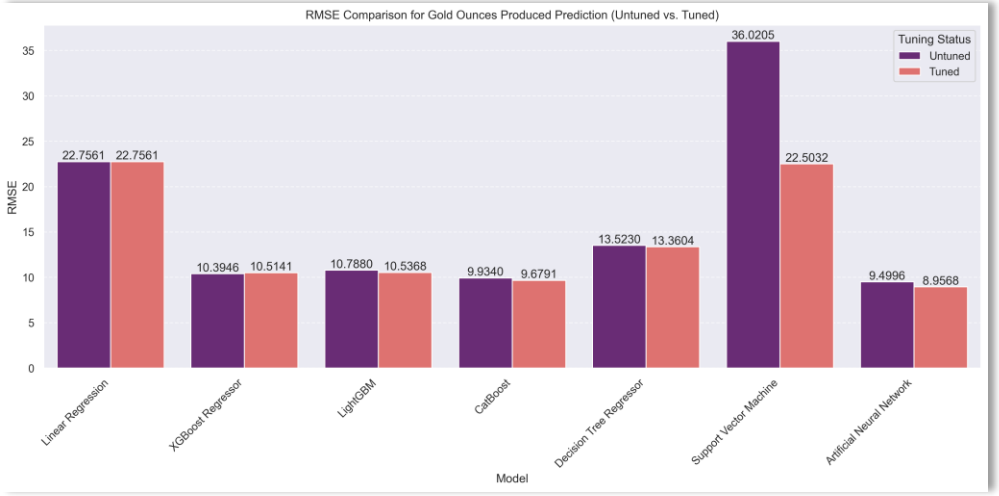
Best Overall Performer:

CatBoost (BASELINE)
(R-squared: 0.7198; RMSE: 0.0471)

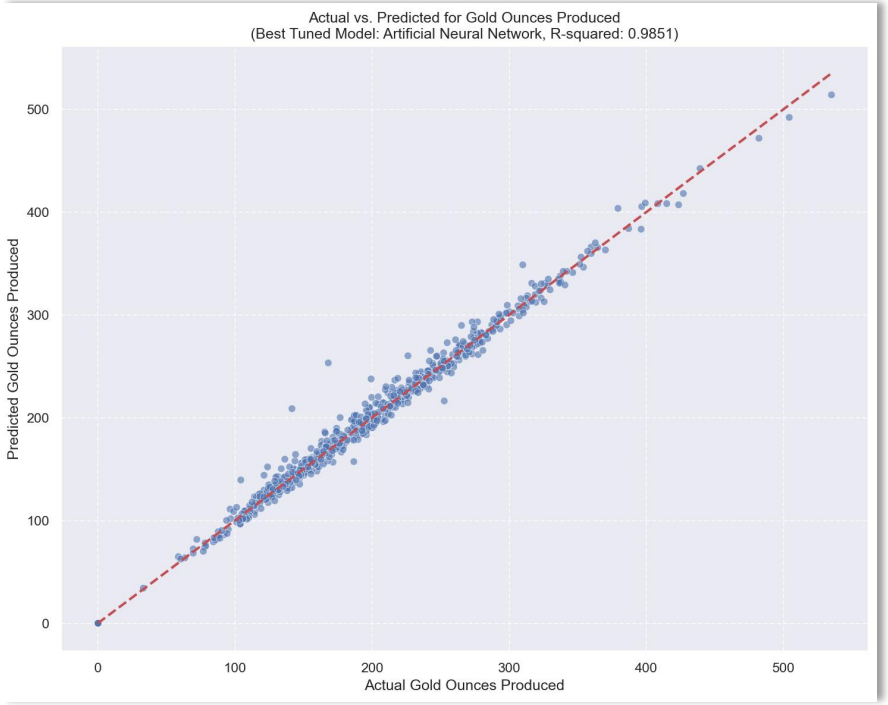
Peculiarly, the Tuned Model **actually performed slightly worse** than the baseline model, which means that the hyperparameter tuning yielded only minor changes with the accuracy of our model, however the Decision Tree and Support Vector Regression both had significant improvements after tuning. We will still use XGBoost (tuned) because it has already gone through cross-validation.

Hyperparameter Tuned Model Results: Gold Ounces





Target	Model	R-squared	MAE	MSE	RMSE	Tuning Status
Gold Ounces Produced	**Artificial Neural Network**	**0.9851**	**5.8910**	**80.2247**	**8.9568**	**Tuned**
Gold Ounces Produced	Artificial Neural Network	0.9832	6.2966	90.2433	9.4996	Untuned
Gold Ounces Produced	CatBoost	0.9826	6.3086	93.6846	9.6791	Tuned
Gold Ounces Produced	CatBoost	0.9817	6.4835	98.6841	9.9340	Untuned
Gold Ounces Produced	XGBoost Regressor	0.9799	6.8273	108.0476	10.3946	Untuned
Gold Ounces Produced	XGBoost Regressor	0.9795	6.8458	110.5458	10.5141	Tuned

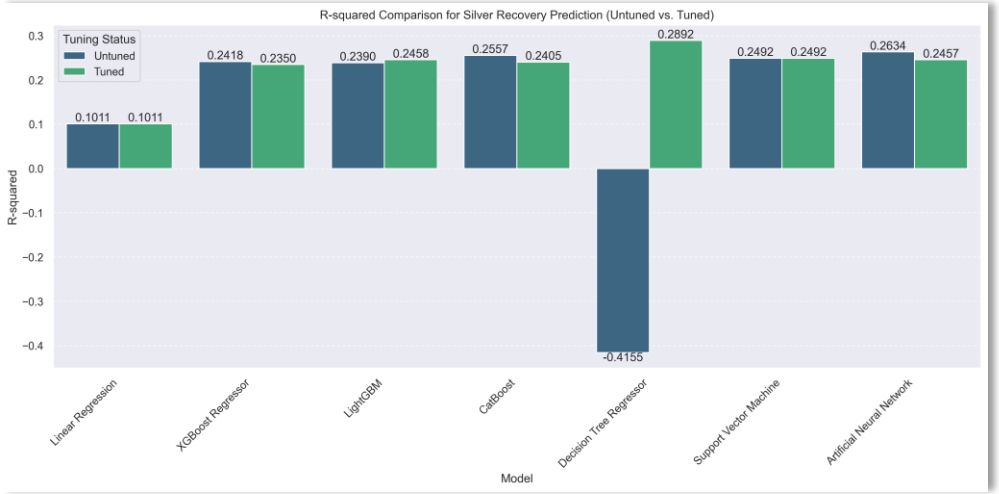


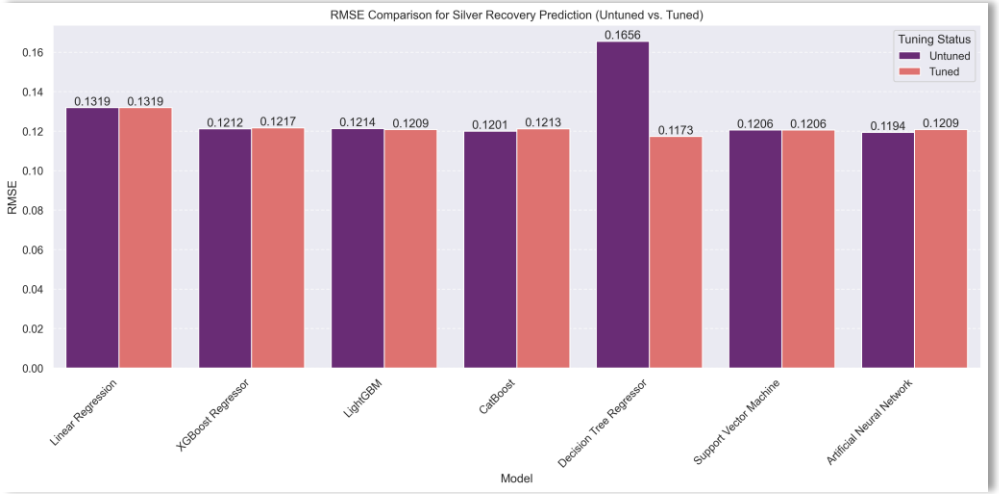
Best Overall Performer:

Artificial Neural Network (TUNED)
(R-squared: 0.9851; RMSE: 8.9568)

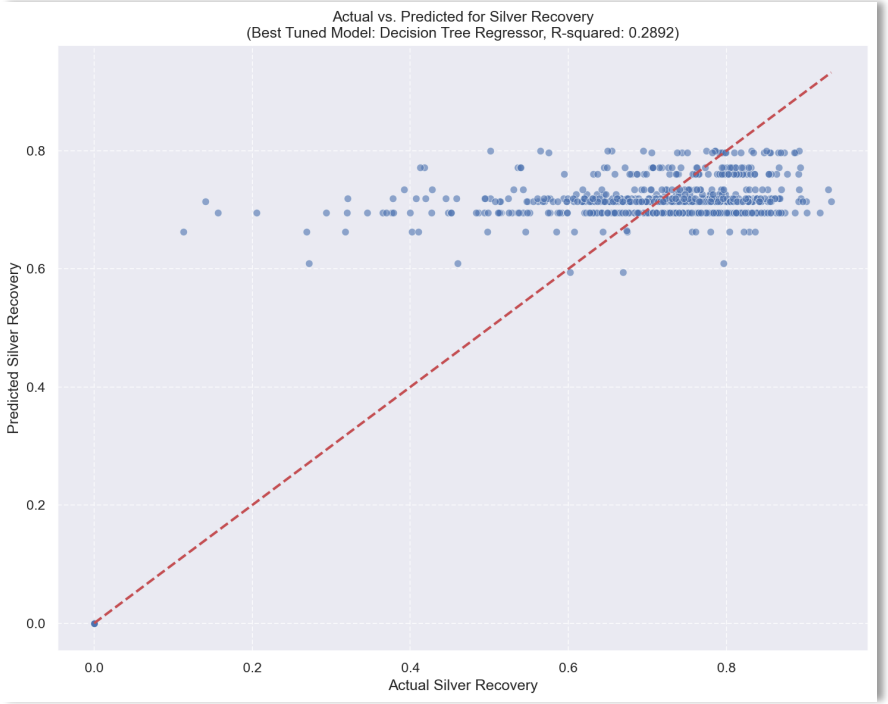
The tuned Artificial Neural Network Model had a **slightly improved performance** than the baseline model, which means that the hyperparameter tuning yielded only minor changes with the accuracy of our model, however the Support Vector Regression had significant improvements after tuning.

Hyperparameter Tuned Model Results: Silver Recovery





Target	Model	R-squared	MAE	MSE	RMSE	Tuning Status
Silver Recovery	**Decision Tree Regressor**	**0.2892**	**0.0850**	**0.0138**	**0.1173**	**Tuned**
Silver Recovery	Artificial Neural Network	0.2634	0.0877	0.0143	0.1194	Untuned
Silver Recovery	CatBoost	0.2557	0.0856	0.0144	0.1201	Untuned
Silver Recovery	Support Vector Machine	0.2492	0.0877	0.0145	0.1206	Tuned
Silver Recovery	Support Vector Machine	0.2492	0.0877	0.0145	0.1206	Untuned
Silver Recovery	LightGBM	0.2458	0.0874	0.0146	0.1209	Tuned

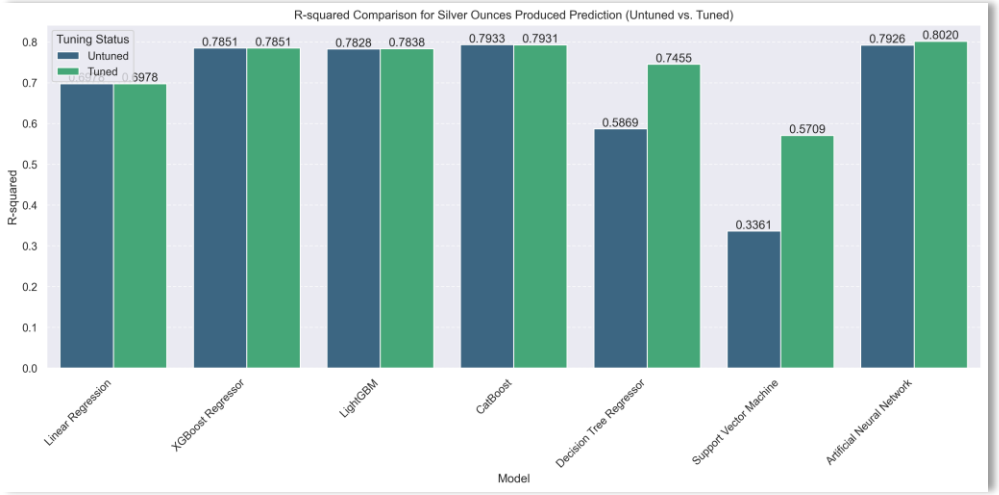


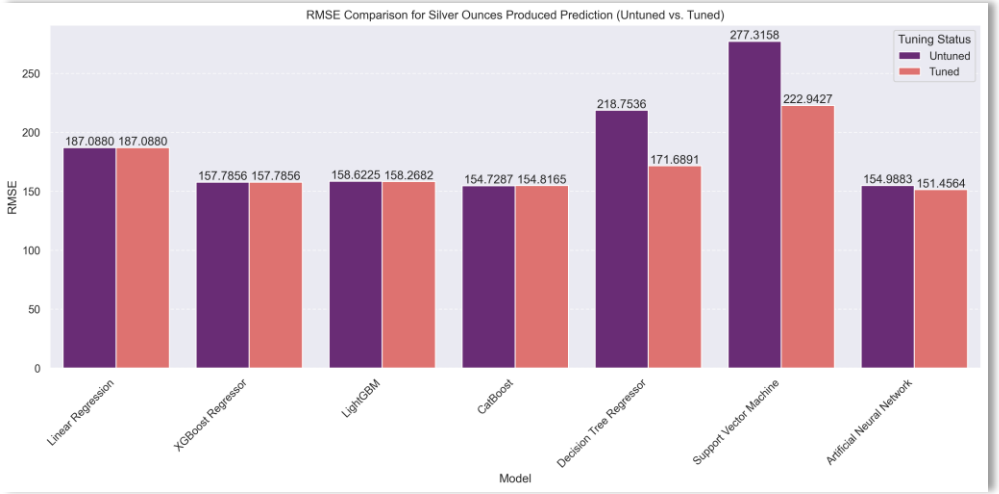
Best Tuned Performer:

Decision Tree Regressor
(R-squared: 0.2892; RMSE: 0.1173)

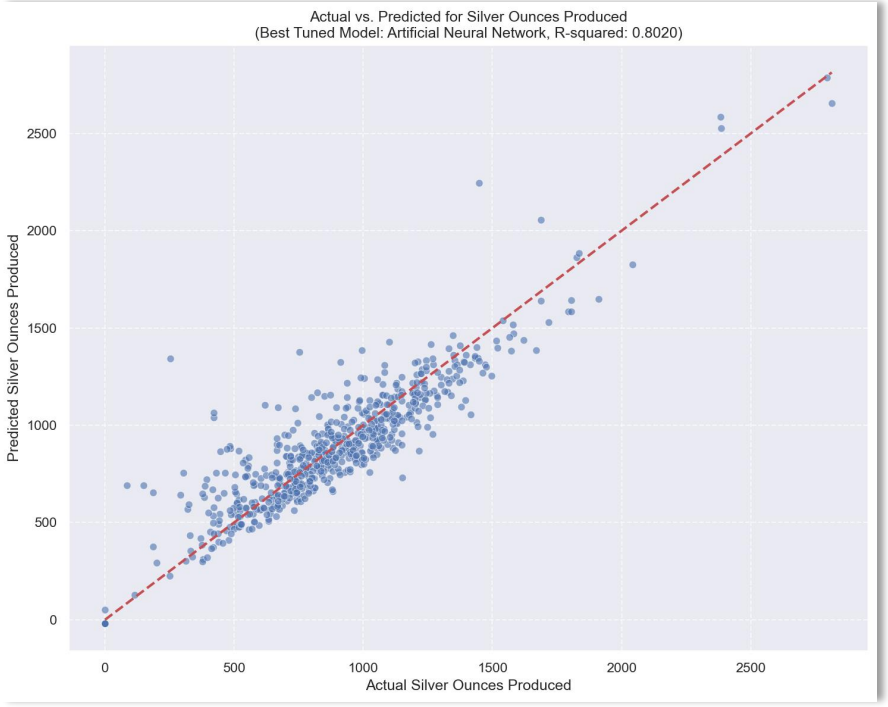
The Decision Tree Regressor **performed much better** prior to its baseline version, and coincidentally, the Decision Tree Regressor is also the most improved model after hyperparameter tuning. All the models has performed slightly worse or did not improve after the tuning, this maybe because the cross-validation step of GridSearchCV.

Hyperparameter Tuned Model Results: Silver Ounces





Target	Model	R-squared	MAE	MSE	RMSE	Tuning Status
Silver Ounces Produced	**Artificial Neural Network**	**0.8020**	**107.2076**	**22939.0438**	**151.4564**	**Tuned**
Silver Ounces Produced	CatBoost	0.7933	108.0777	23940.9791	154.7287	Untuned
Silver Ounces Produced	CatBoost	0.7931	108.0324	23968.1402	154.8165	Tuned
Silver Ounces Produced	Artificial Neural Network	0.7926	108.9027	24021.3693	154.9883	Untuned
Silver Ounces Produced	XGBoost Regressor	0.7851	110.0901	24896.2907	157.7856	Tuned
Silver Ounces Produced	XGBoost Regressor	0.7851	110.0901	24896.2907	157.7856	Untuned

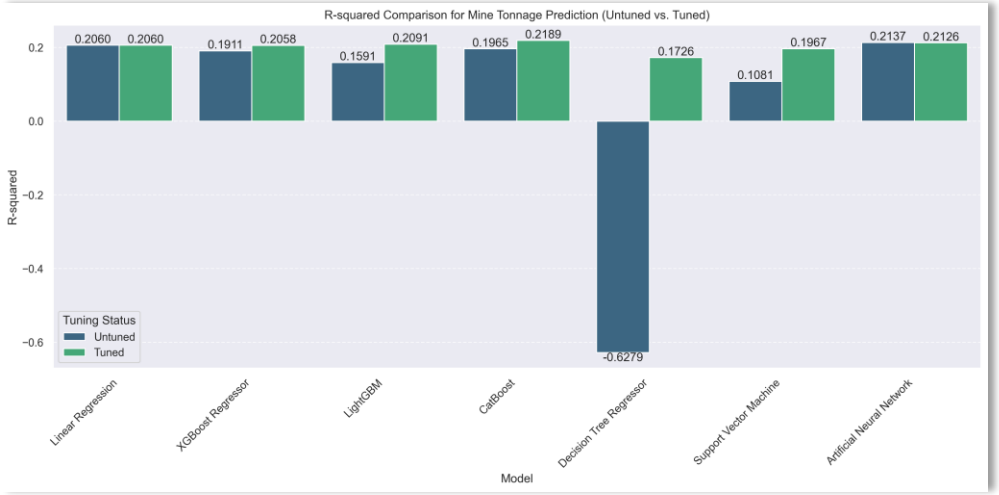


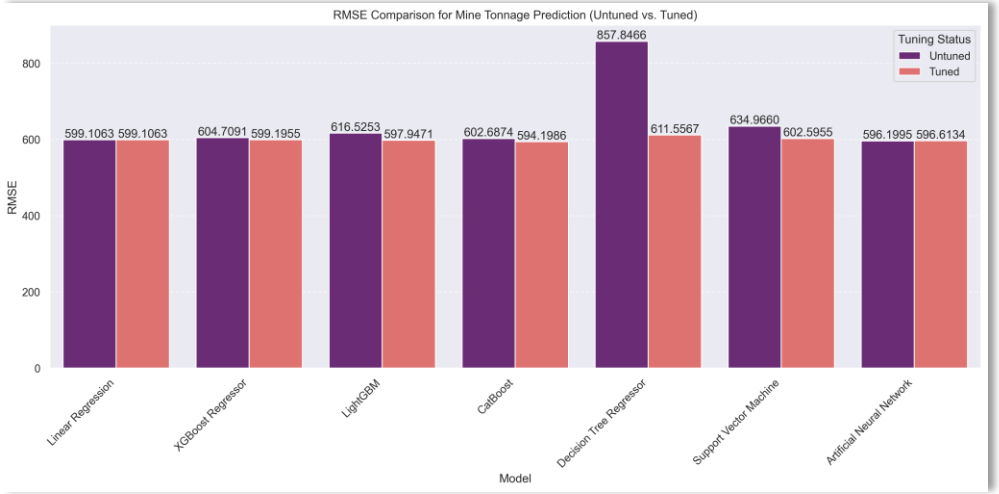
Best Tuned Performer:

Artificial Neural Network
(R-squared: 0.8020; RMSE: 151.4564)

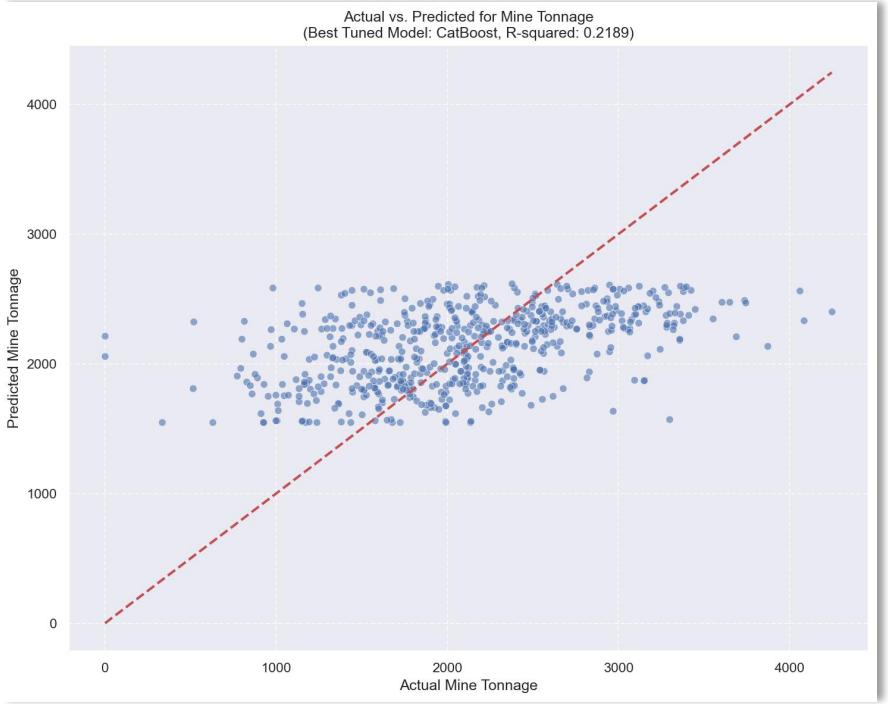
The Artificial Neural Network **performed slightly better** whilst the other models have also slightly improved or maintained their R-squared values. The RMSE values for all the models slightly decreased or remained the same. Hyperparameter Tuning was most effective in improving model performance for the Decision Tree and SVR models.

Hyperparameter Tuned Model Results: Mine Tonnage





Target	Model	R-squared	MAE	MSE	RMSE	Tuning Status
Mine Tonnage	**CatBoost**	**0.2189**	**469.5555**	**353071.9989**	**594.1986**	**Tuned**
Mine Tonnage	Artificial Neural Network	0.2137	471.0984	355453.8864	596.1995	Untuned
Mine Tonnage	Artificial Neural Network	0.2126	470.9128	355947.5545	596.6134	Tuned
Mine Tonnage	LightGBM	0.2091	474.7449	357540.7370	597.9471	Tuned
Mine Tonnage	Linear Regression	0.2060	474.0768	358928.3828	599.1063	Tuned
Mine Tonnage	Linear Regression	0.2060	474.0768	358928.3828	599.1063	Untuned



Best Tuned Performer:

CatBoost
(R-squared: 0.2189; RMSE: 594.1986)

The CatBoost tuned model **performed slightly better** than our best baseline ANN model. Some of the models have performed slightly worse or did not improve after the tuning, this maybe because the cross-validation step of GridSearchCV. Meanwhile, the Decision Tree Regressor and SVR model improved significantly after hyperparameter tuning.

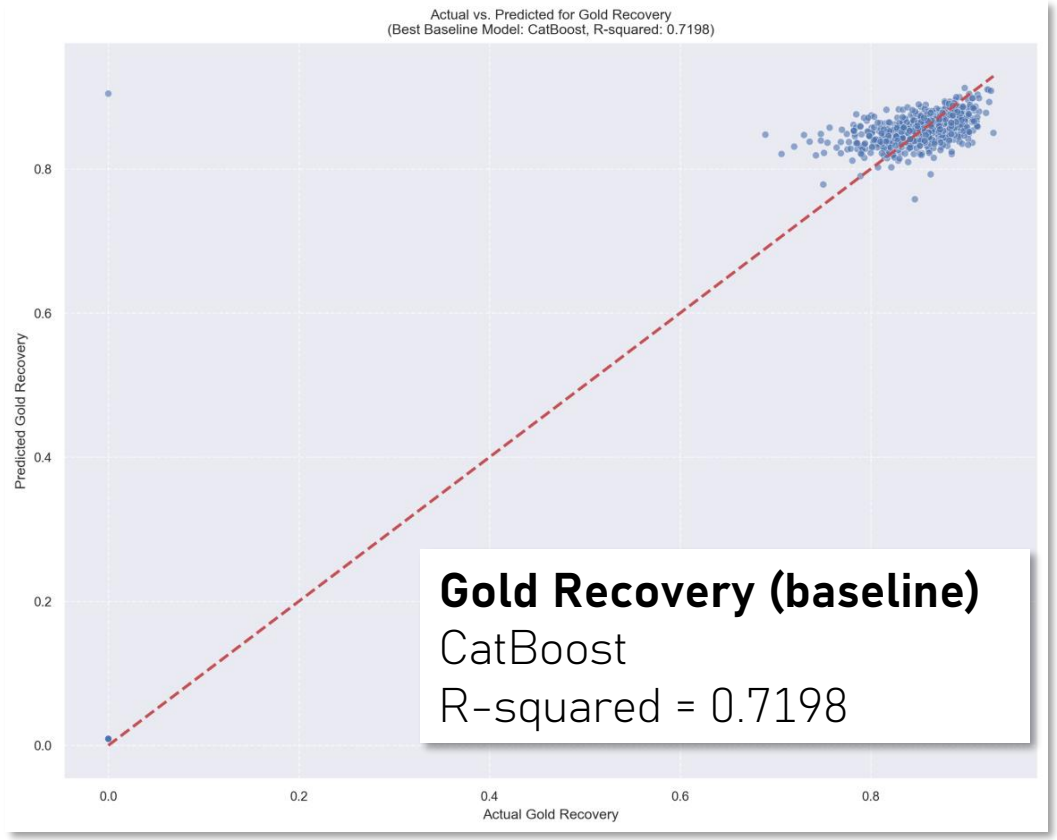
Tuned vs Baseline Complete Model Comparisons: Gold Ounces and Gold Recovery

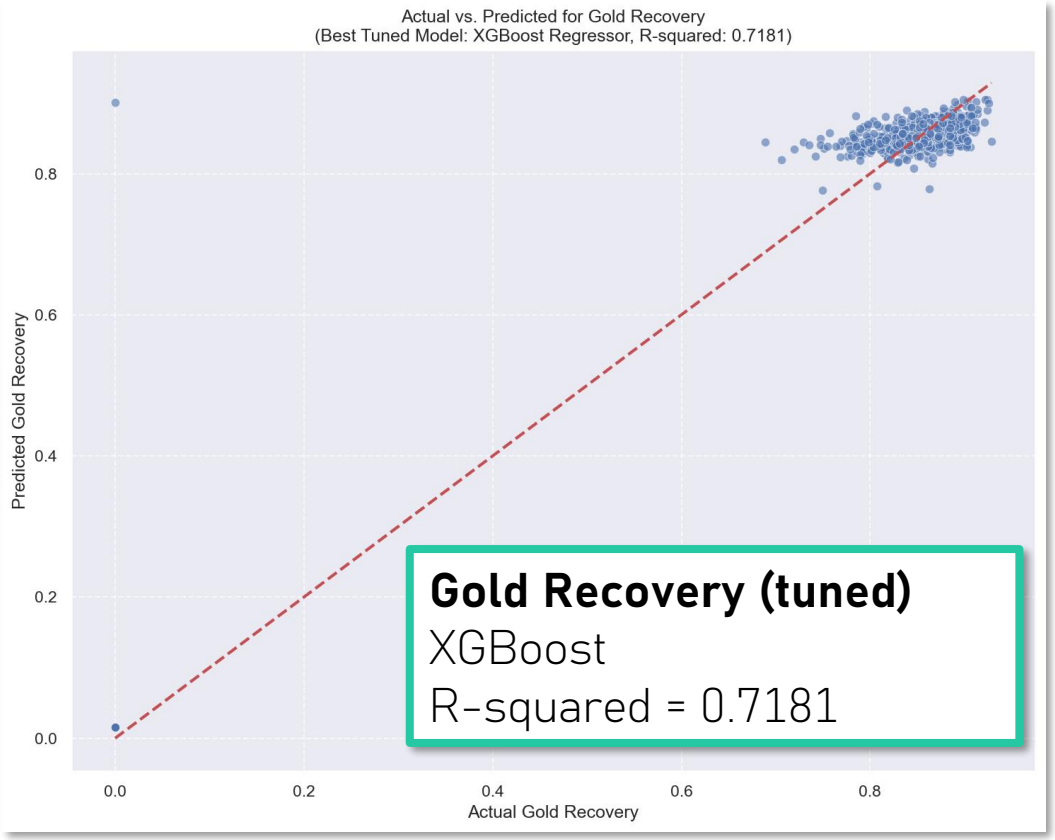
Gold Ounces: minor improvement between best Baseline and Tuned models, which may entail that the model is already optimal prior to hyperparameter tuning.
Gold Recovery: only minor improvement was observed compared to the previous best Baseline and Tuned models

Target	Model	R-squared	MAE	MSE	RMSE	Tuning Status
Gold Ounces Produced	**Artificial Neural Network**	**0.9851**	**5.8910**	**80.2247**	**8.9568**	**Tuned**
Gold Ounces Produced	Artificial Neural Network	0.9832	6.2966	90.2433	9.4996	Untuned
Gold Ounces Produced	CatBoost	0.9826	6.3086	93.6846	9.6791	Tuned
Gold Ounces Produced	CatBoost	0.9817	6.4835	98.6841	9.9340	Untuned
Gold Ounces Produced	XGBoost Regressor	0.9799	6.8273	108.0476	10.3946	Untuned
Gold Ounces Produced	XGBoost Regressor	0.9795	6.8458	110.5458	10.5141	Tuned
Gold Ounces Produced	LightGBM	0.9794	6.7502	111.0239	10.5368	Tuned
Gold Ounces Produced	LightGBM	0.9784	6.8858	116.3813	10.7880	Untuned
Gold Ounces Produced	Decision Tree Regressor	0.9669	8.9890	178.5002	13.3604	Tuned
Gold Ounces Produced	Decision Tree Regressor	0.9660	9.5935	182.8720	13.5230	Untuned
Gold Ounces Produced	Support Vector Machine	0.9060	8.9034	506.3919	22.5032	Tuned
Gold Ounces Produced	Linear Regression	0.9039	12.7502	517.8380	22.7561	Tuned
Gold Ounces Produced	Linear Regression	0.9039	12.7502	517.8380	22.7561	Untuned
Gold Ounces Produced	Support Vector Machine	0.7591	15.0351	1297.4788	36.0205	Untuned
Gold Recovery	**CatBoost**	**0.7198**	**0.0245**	**0.0022**	**0.0471**	**Untuned**
Gold Recovery	**XGBoost Regressor**	**0.7181**	**0.0250**	**0.0022**	**0.0472**	**Tuned**
Gold Recovery	CatBoost	0.7173	0.0248	0.0022	0.0473	Tuned
Gold Recovery	Support Vector Machine	0.7170	0.0251	0.0022	0.0473	Tuned
Gold Recovery	XGBoost Regressor	0.7158	0.0249	0.0022	0.0474	Untuned
Gold Recovery	Decision Tree Regressor	0.7123	0.0256	0.0023	0.0477	Tuned
Gold Recovery	LightGBM	0.7115	0.0254	0.0023	0.0478	Tuned
Gold Recovery	Artificial Neural Network	0.7068	0.0252	0.0023	0.0482	Tuned
Gold Recovery	Artificial Neural Network	0.6922	0.0258	0.0024	0.0494	Untuned
Gold Recovery	LightGBM	0.6875	0.0268	0.0025	0.0497	Untuned
Gold Recovery	Decision Tree Regressor	0.6096	0.0347	0.0031	0.0556	Untuned
Gold Recovery	Support Vector Machine	0.5815	0.0425	0.0033	0.0576	Untuned
Gold Recovery	Linear Regression	0.2570	0.0346	0.0059	0.0767	Tuned
Gold Recovery	Linear Regression	0.2570	0.0346	0.0059	0.0767	Untuned

Best Gold Ounces Model: Artificial Neural Network (R-squared: 0.9851; RMSE: 8.9568)
Best Gold Recovery Model: XGBoost Regressor (R-squared: 0.7181; RMSE: 0.0472)
Even if the Untuned CatBoost Model has slightly higher R-squared values, we will still use the tuned XGBoost model since this has already gone through Cross-Validation.

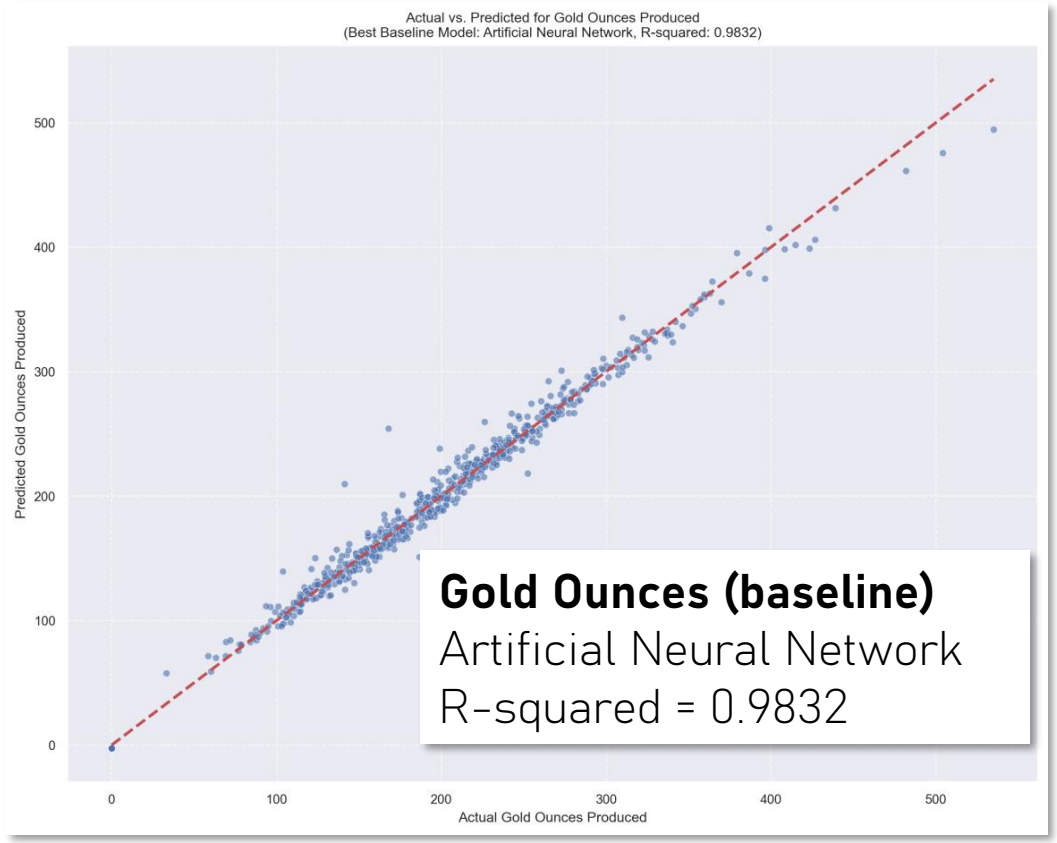
Hyperparameter Tuned Model Results: Gold Recovery



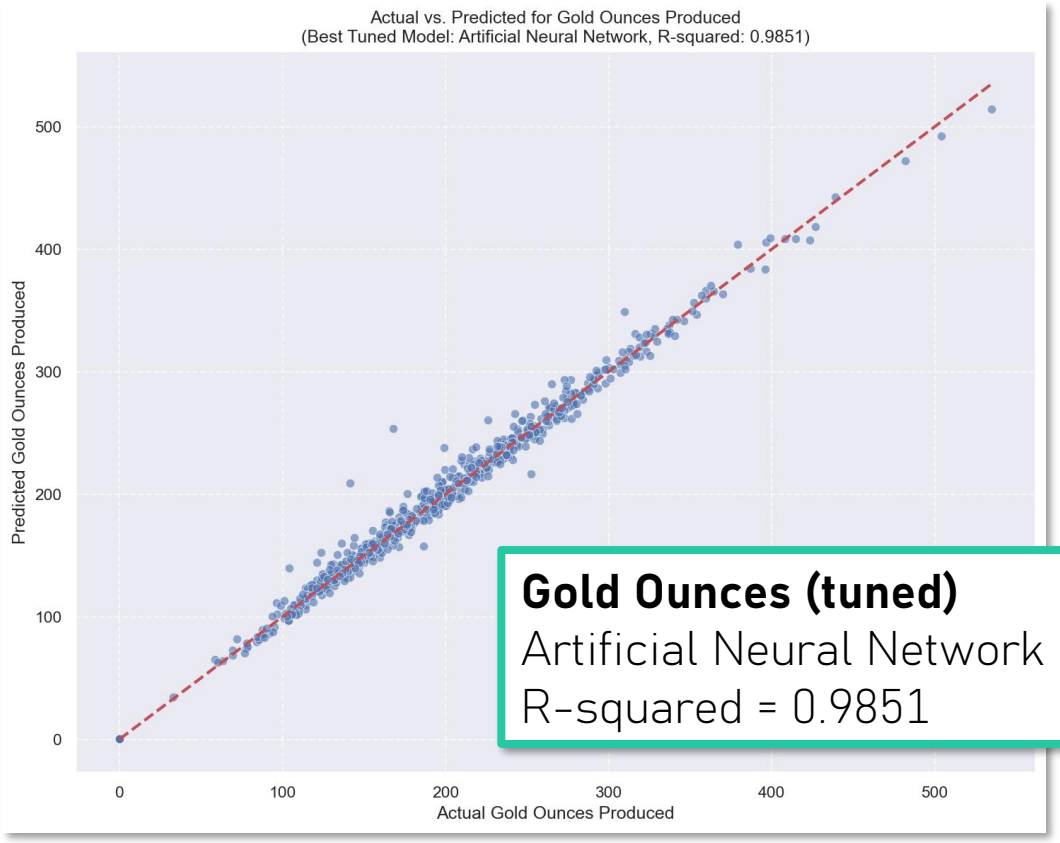


Gold Recovery did not seem to have a significant performance improvement compared to the best baseline model. We will choose the tuned XGBoost model since it has already undergone Cross-Validation during tuning.

Hyperparameter Tuned Model Results: Gold Ounces



Gold Ounces (baseline)
Artificial Neural Network
R-squared = 0.9832



Gold Ounces (tuned)
Artificial Neural Network
R-squared = 0.9851

Gold Ounces also did not seem to have a significant performance improvement compared to the best baseline model. The tuned ANN for Gold Ounces was chosen due to having undergone cross-validation in GridSearchCV.

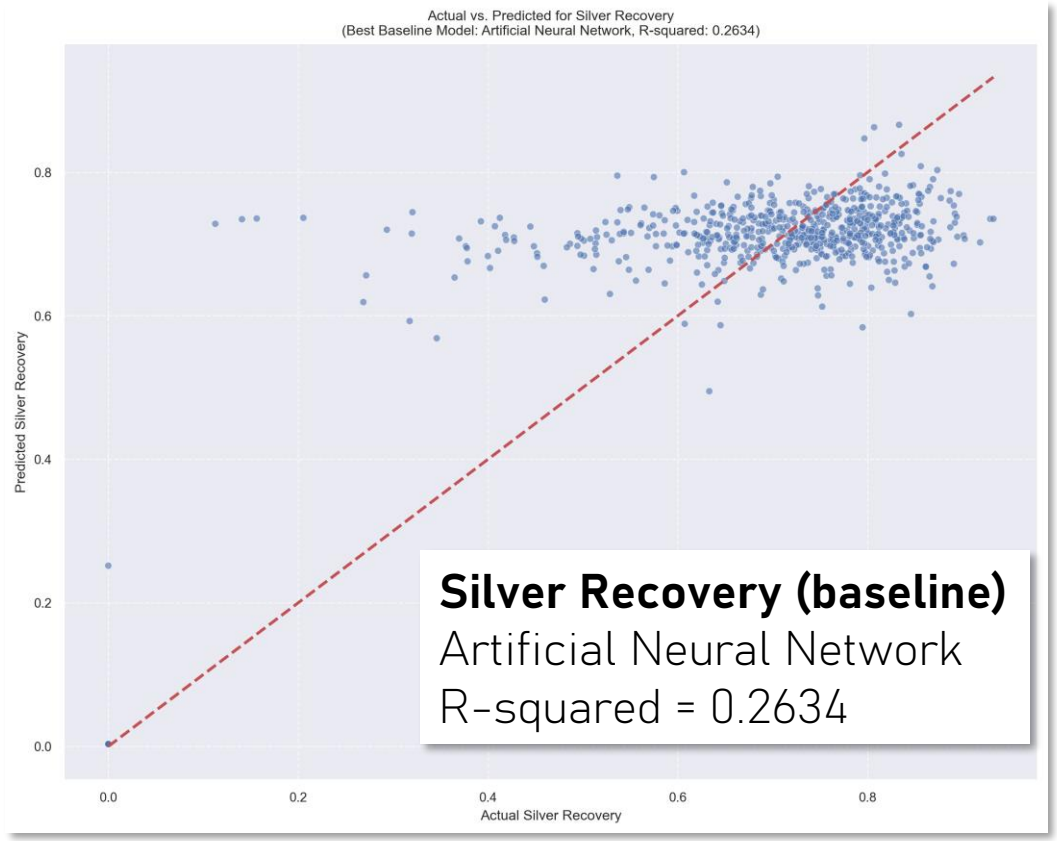
Tuned vs Baseline Complete Model Comparisons: Silver Ounces and Silver Recovery

Silver Ounces: minor improvement between best Baseline and Tuned models, which may entail that the model is already optimal prior to hyperparameter tuning.
Silver Recovery: only minor improvement was observed compared to the previous best Baseline and Tuned models

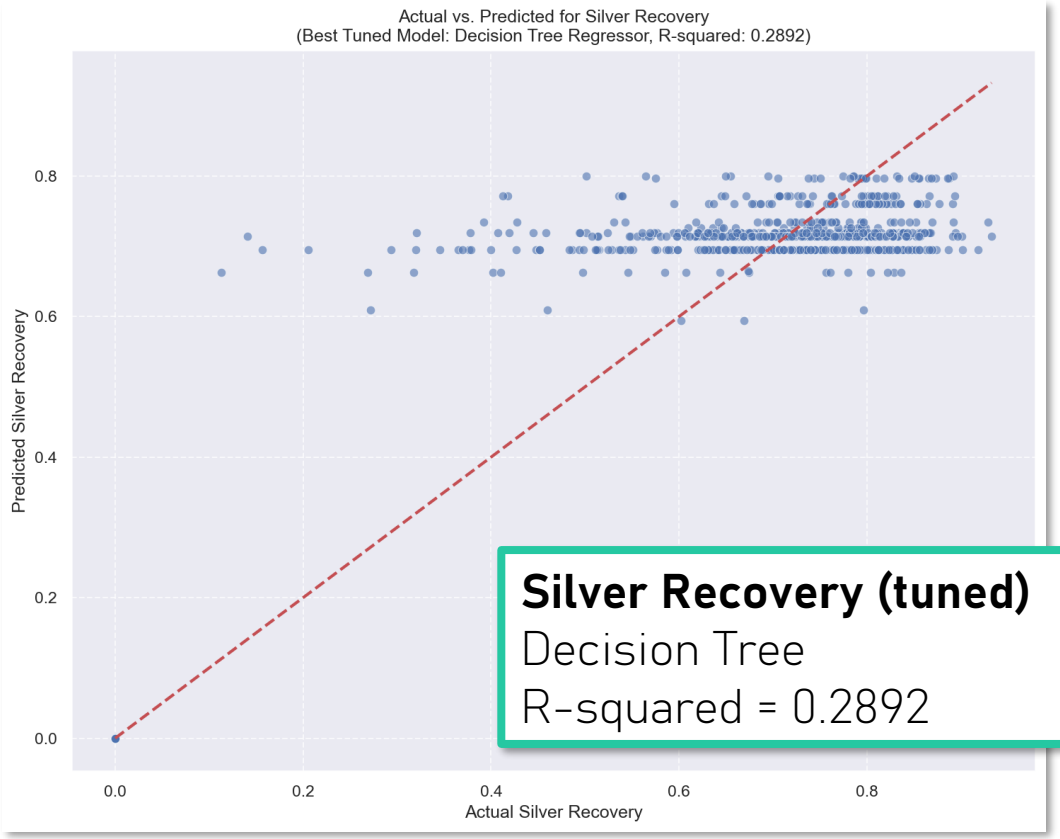
Target	Model	R-squared	MAE	MSE	RMSE	Tuning Status
Silver Ounces Produced	**Artificial Neural Network**	**0.8020**	**107.2076**	**22939.0438**	**151.4564**	**Tuned**
Silver Ounces Produced	CatBoost	0.7933	108.0777	23940.9791	154.7287	Untuned
Silver Ounces Produced	CatBoost	0.7931	108.0324	23968.1402	154.8165	Tuned
Silver Ounces Produced	Artificial Neural Network	0.7926	108.9027	24021.3693	154.9883	Untuned
Silver Ounces Produced	XGBoost Regressor	0.7851	110.0901	24896.2907	157.7856	Tuned
Silver Ounces Produced	XGBoost Regressor	0.7851	110.0901	24896.2907	157.7856	Untuned
Silver Ounces Produced	LightGBM	0.7838	111.2168	25048.8248	158.2682	Tuned
Silver Ounces Produced	LightGBM	0.7828	111.1808	25161.0882	158.6225	Untuned
Silver Ounces Produced	Decision Tree Regressor	0.7455	121.9353	29477.1357	171.6891	Tuned
Silver Ounces Produced	Linear Regression	0.6978	126.9735	35001.9138	187.0880	Tuned
Silver Ounces Produced	Linear Regression	0.6978	126.9735	35001.9138	187.0880	Untuned
Silver Ounces Produced	Decision Tree Regressor	0.5869	154.6529	47853.1352	218.7536	Untuned
Silver Ounces Produced	Support Vector Machine	0.5709	132.6564	49703.4430	222.9427	Tuned
Silver Ounces Produced	Support Vector Machine	0.3361	182.8102	76904.0706	277.3158	Untuned
Silver Recovery	**Decision Tree Regressor**	**0.2892**	**0.0850**	**0.0138**	**0.1173**	**Tuned**
Silver Recovery	Artificial Neural Network	0.2634	0.0877	0.0143	0.1194	Untuned
Silver Recovery	CatBoost	0.2557	0.0856	0.0144	0.1201	Untuned
Silver Recovery	Support Vector Machine	0.2492	0.0877	0.0145	0.1206	Tuned
Silver Recovery	Support Vector Machine	0.2492	0.0877	0.0145	0.1206	Untuned
Silver Recovery	LightGBM	0.2458	0.0874	0.0146	0.1209	Tuned
Silver Recovery	Artificial Neural Network	0.2457	0.0893	0.0146	0.1209	Tuned
Silver Recovery	XGBoost Regressor	0.2418	0.0865	0.0147	0.1212	Untuned
Silver Recovery	CatBoost	0.2405	0.0868	0.0147	0.1213	Tuned
Silver Recovery	LightGBM	0.2390	0.0878	0.0147	0.1214	Untuned
Silver Recovery	XGBoost Regressor	0.2350	0.0872	0.0148	0.1217	Tuned
Silver Recovery	Linear Regression	0.1011	0.0947	0.0174	0.1319	Tuned
Silver Recovery	Linear Regression	0.1011	0.0947	0.0174	0.1319	Untuned
Silver Recovery	Decision Tree Regressor	-0.4155	0.1227	0.0274	0.1656	Untuned

Best Silver Ounces Model: Artificial Neural Network (R-squared: 0.8020; RMSE: 151.4564)
Best Silver Recovery Model: XGBoost Regressor (R-squared: 0.2892; RMSE: 0.1173)

Hyperparameter Tuned Model Results: Silver Recovery



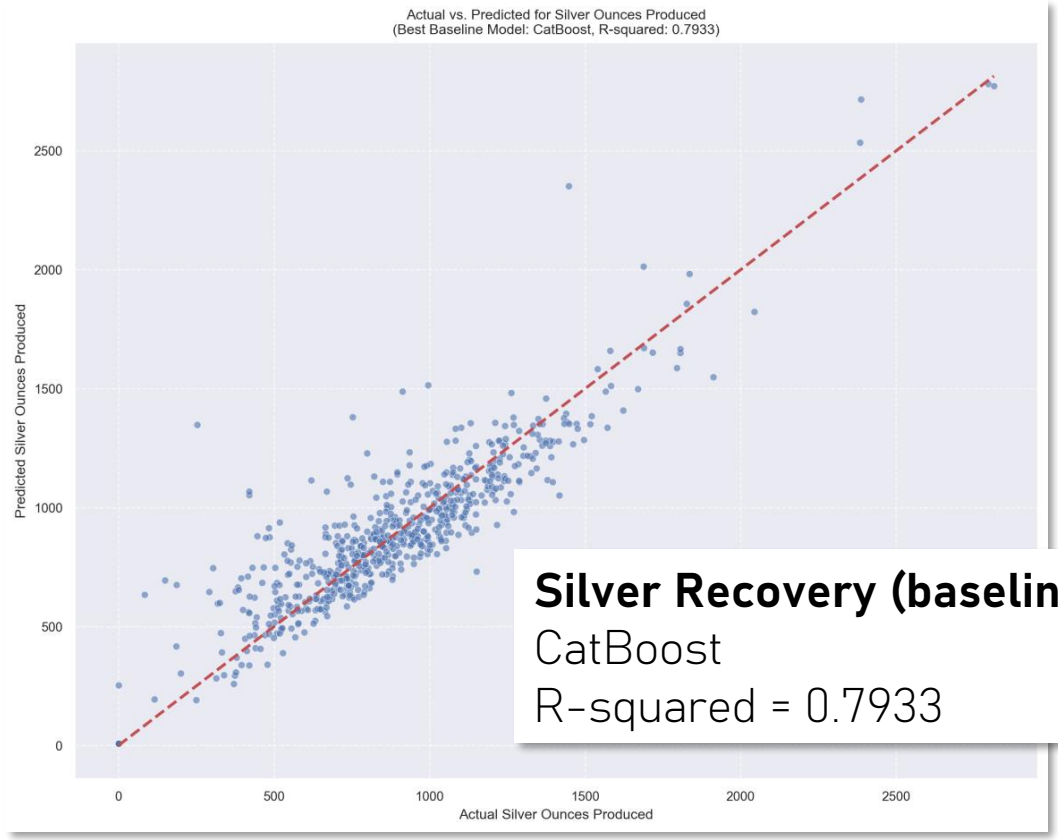
Silver Recovery (baseline)
Artificial Neural Network
R-squared = 0.2634



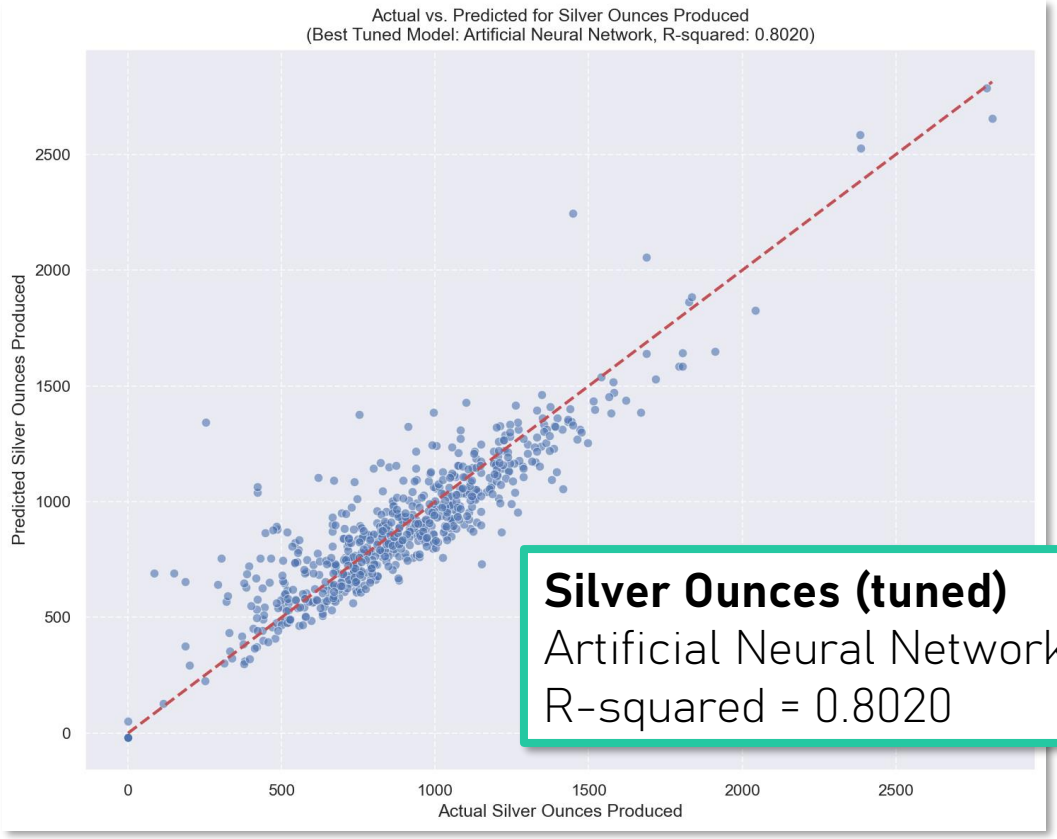
Silver Recovery (tuned)
Decision Tree
R-squared = 0.2892

Silver Recovery had a slight performance improvement compared to the best baseline model. The tuned Decision Tree coincidentaly, the decision tree model is also the most improved model after tuning.

Hyperparameter Tuned Model Results: Silver Ounces



Silver Recovery (baseline)
CatBoost
R-squared = 0.7933



Silver Ounces (tuned)
Artificial Neural Network
R-squared = 0.8020

Silver Ounces had a minimal performance improvement compared to the best baseline model. The tuned ANN has been selected because of the higher R-squared and since it has already been cross-validated.

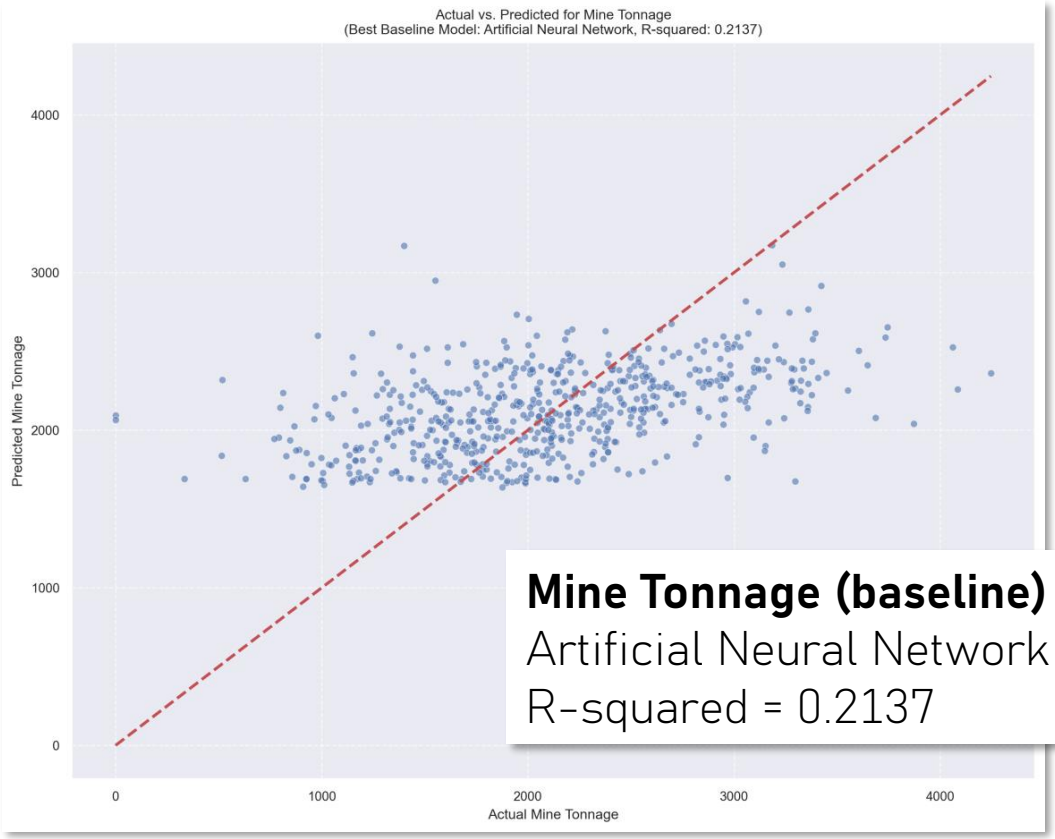
Tuned vs Baseline Complete Model Comparisons: Mine Tonnage

Mine Tonnage: minor improvement between best Baseline and Tuned models, which may entail that the prior best baseline model is already optimal prior to hyperparameter tuning. A larger performance improvement was observed between the baseline Catboost model and the tuned CatBoost model (which was selected in this target variable).

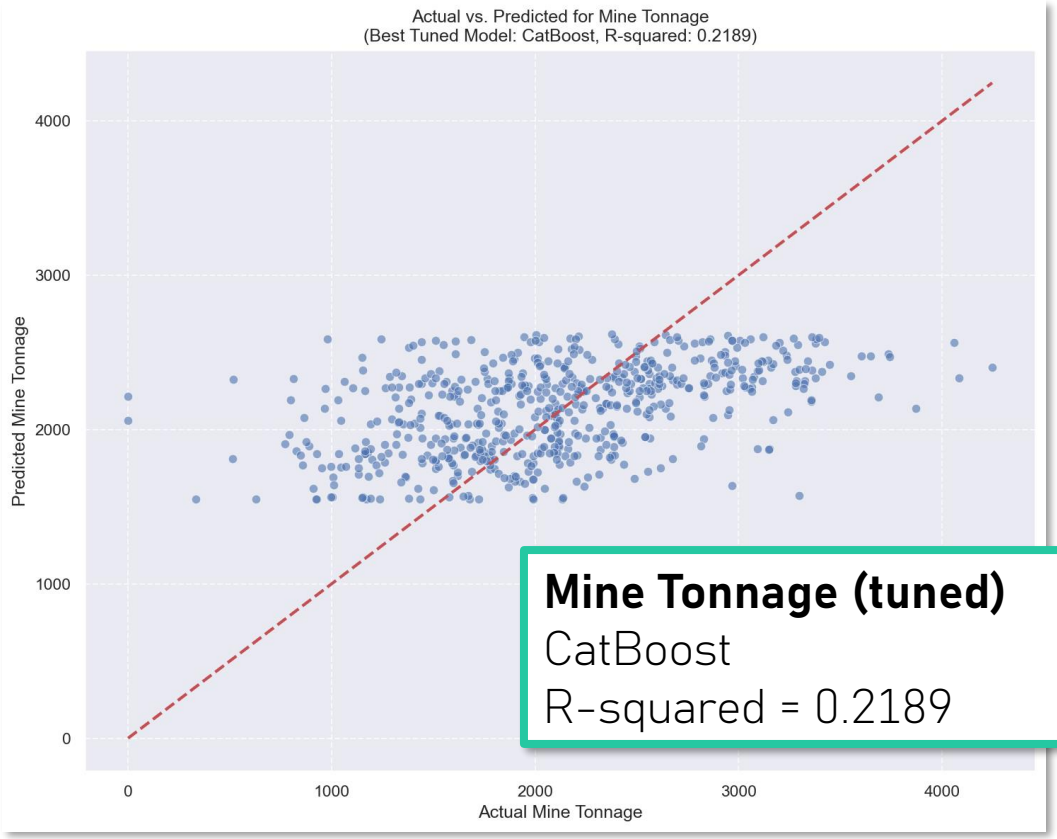
Target	Model	R-squared	MAE	MSE	RMSE	Tuning Status
Mine Tonnage	**CatBoost**	**0.2189**	**469.5555**	**353071.9989**	**594.1986**	**Tuned**
Mine Tonnage	Artificial Neural Network	0.2137	471.0984	355453.8864	596.1995	Untuned
Mine Tonnage	Artificial Neural Network	0.2126	470.9128	355947.5545	596.6134	Tuned
Mine Tonnage	LightGBM	0.2091	474.7449	357540.7370	597.9471	Tuned
Mine Tonnage	Linear Regression	0.2060	474.0768	358928.3828	599.1063	Tuned
Mine Tonnage	Linear Regression	0.2060	474.0768	358928.3828	599.1063	Untuned
Mine Tonnage	XGBoost Regressor	0.2058	475.8304	359035.2771	599.1955	Tuned
Mine Tonnage	Support Vector Machine	0.1967	475.7553	363121.3146	602.5955	Tuned
Mine Tonnage	CatBoost	0.1965	476.7977	363232.1265	602.6874	Untuned
Mine Tonnage	XGBoost Regressor	0.1911	478.2959	365673.1277	604.7091	Untuned
Mine Tonnage	Decision Tree Regressor	0.1726	487.3849	374001.5472	611.5567	Tuned
Mine Tonnage	LightGBM	0.1591	486.1358	380103.4234	616.5253	Untuned
Mine Tonnage	Support Vector Machine	0.1081	503.6970	403181.8826	634.9660	Untuned
Mine Tonnage	Decision Tree Regressor	-0.6279	690.7254	735900.7238	857.8466	Untuned

Best Mine Tonnage Model: CatBoost (R-squared: 0.2189; RMSE: 594.1986)
The tuned CatBoost Model has slightly higher R-squared values, than the previous baseline ANN model

Hyperparameter Tuned Model Results: Mine Tonnage



Mine Tonnage (baseline)
Artificial Neural Network
R-squared = 0.2137

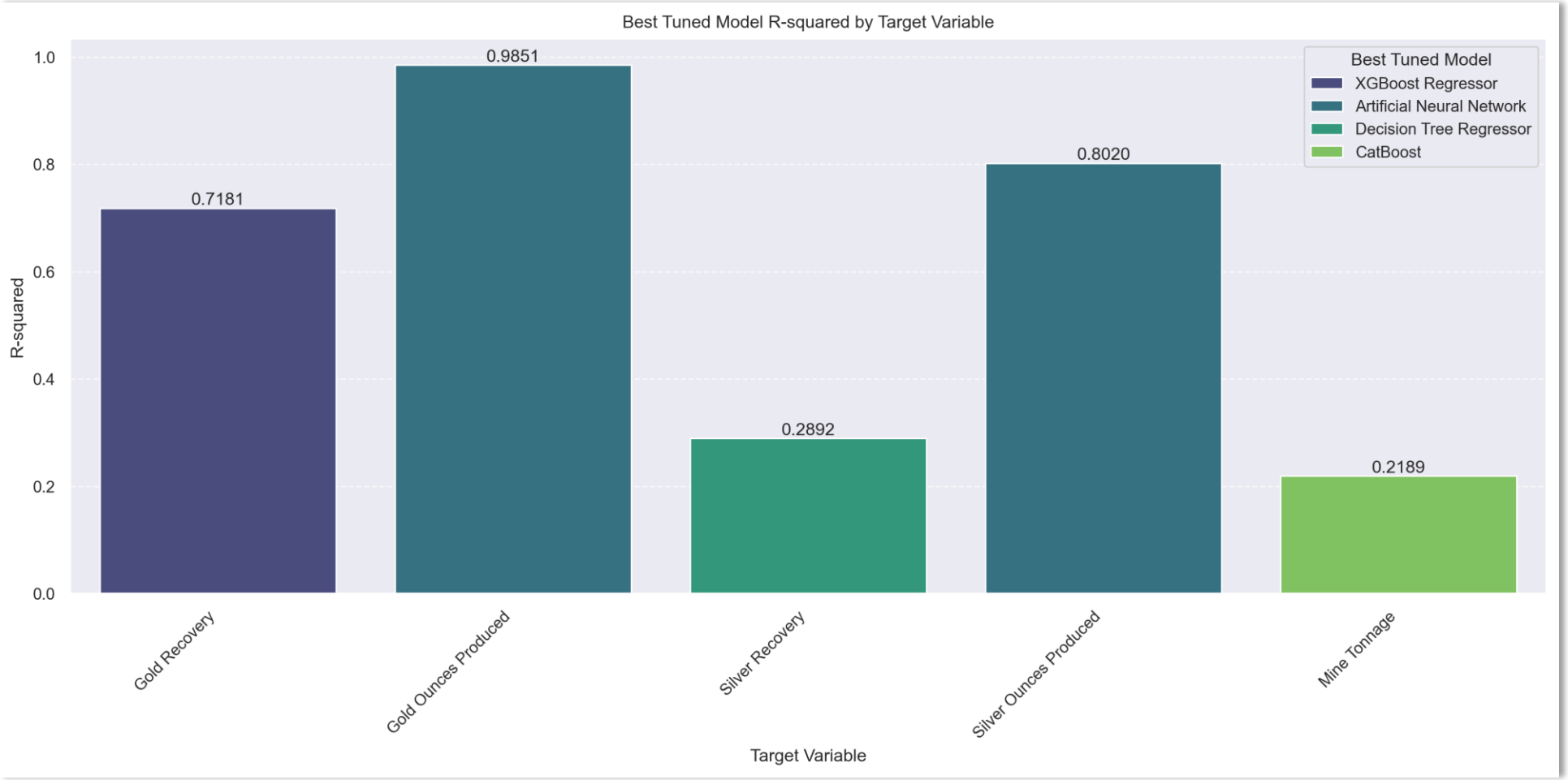


Mine Tonnage (tuned)
CatBoost
R-squared = 0.2189

Mine Tonnage had a minimal performance improvement compared to the best baseline model. The tuned CatBoost has been selected because of the higher R-squared and since it has already been cross-validated.

Final Model Selection

Target	Regression Model	R-squared	RMSE
Gold Ounces Produced	Artificial Neural Network (Multi Layer Perceptron)	**0.9851**	**8.9568**
Gold Recovery	XGBoost Regressor	**0.7181**	**0.0472**
Silver Ounces Produced	Artificial Neural Network	**0.8020**	**151.4564**
Silver Recovery	Decision Tree Regressor	**0.2892**	**0.1173**
Mine Tonnage	CatBoost	**0.2189**	**594.1986**



SHAP model interpretations

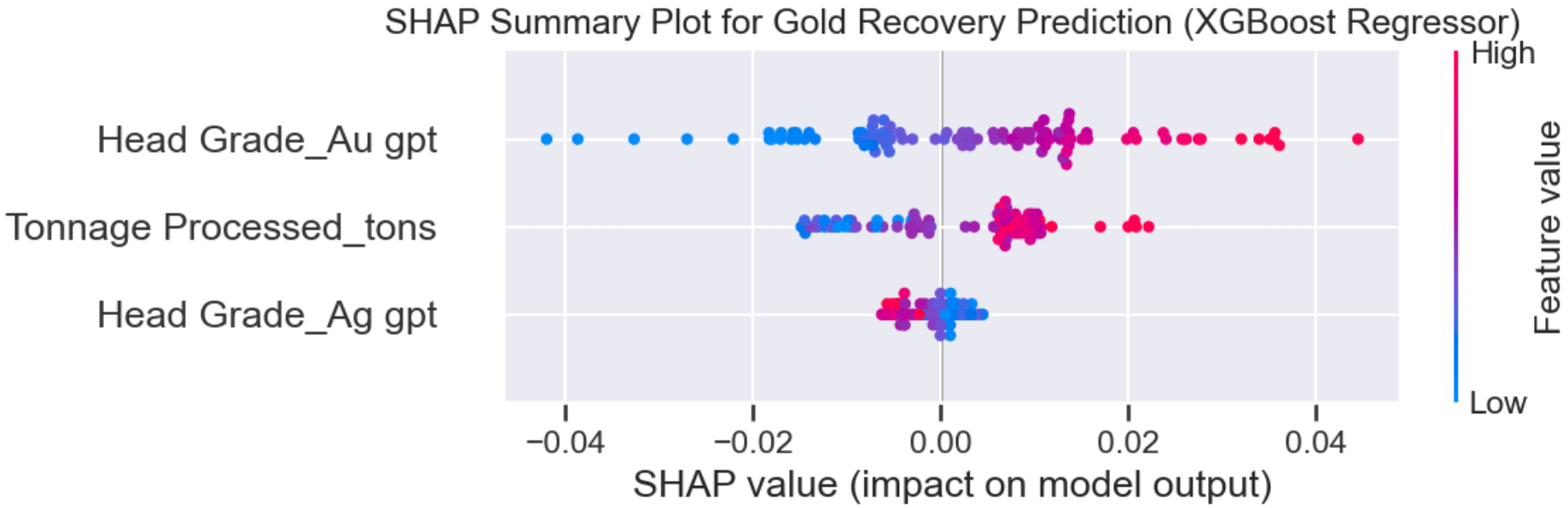
- Based on the official documentation for shap:

"SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions"

- Using the SHAP library we should be able to provide an outlook to how our Machine Learning Models are affected by each feature.

SHAP model for Gold Recovery

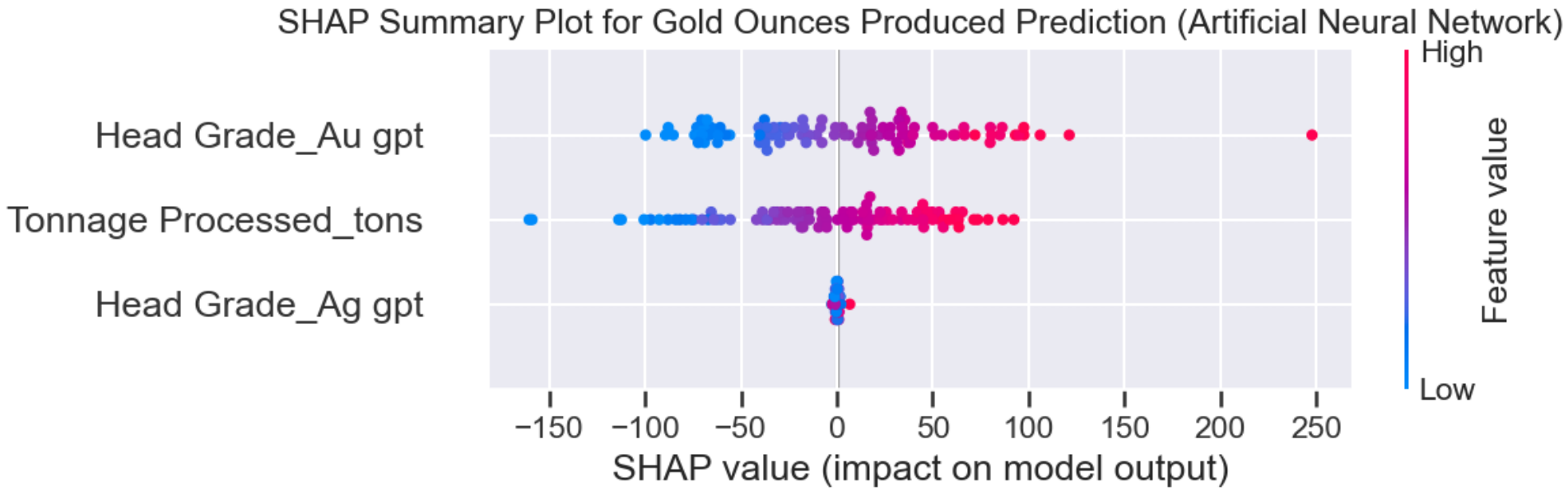
Gold Recovery (tuned)
XGBoost
R-squared = 0.7181



- **Most Influential Feature:** 'Head Grade_Au gpt' appears to be the most influential feature
- Higher 'Head Grade_Au gpt' generally leads to higher gold recovery.

SHAP model for Gold Ounces

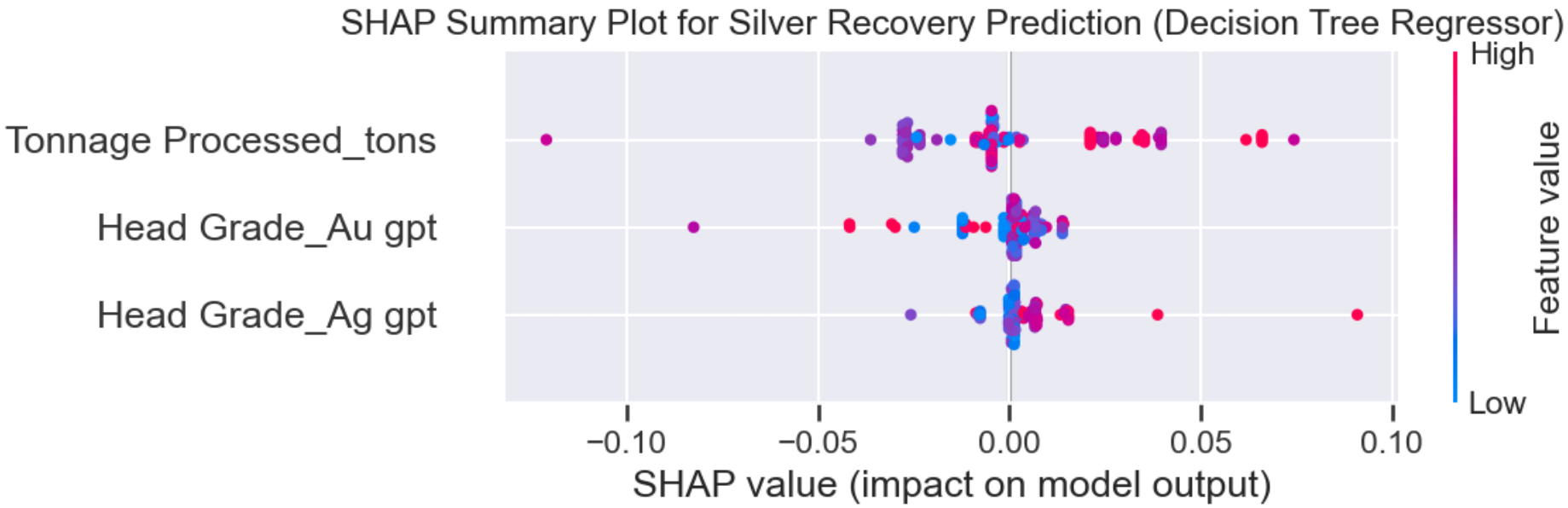
Gold Ounces (tuned)
Artificial Neural Network
R-squared = 0.9851



- **Most Influential Feature:** All three features ('Head Grade_Au gpt', 'Tonnage Processed_tons', 'Head Grade_Ag gpt') are highly important.
- 'Head Grade_Au gpt' and 'Tonnage Processed_tons' show a strong positive correlation with the output, meaning higher values of these features lead to higher predicted gold ounces produced.

SHAP model for Silver Recovery

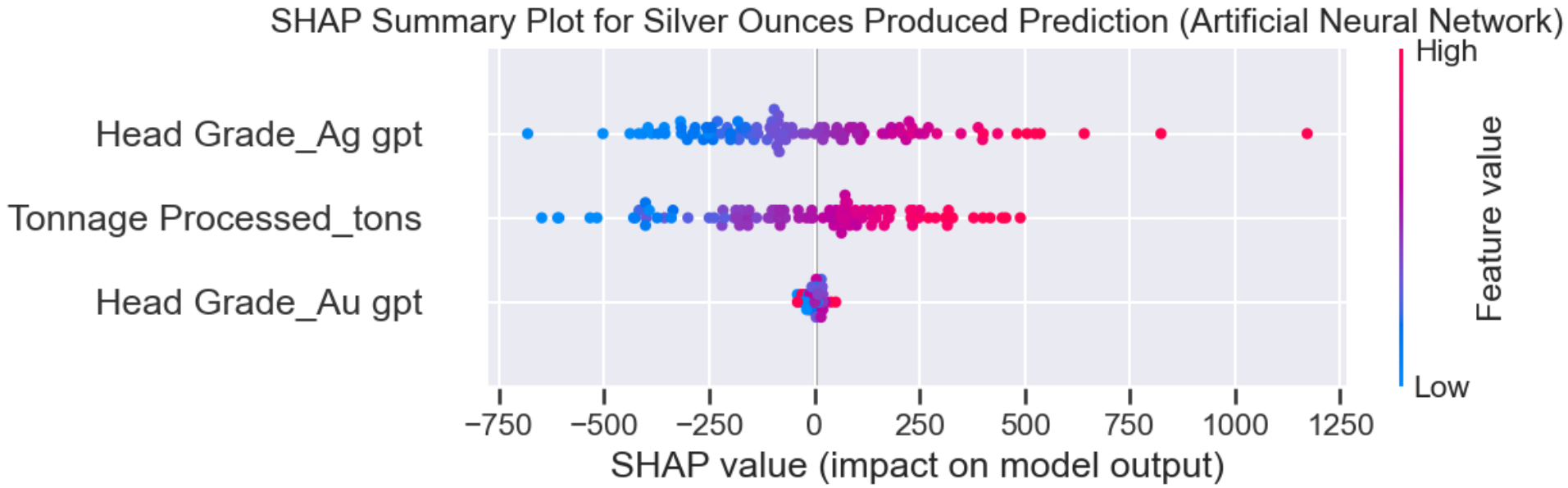
Silver Recovery (tuned)
Decision Tree
R-squared = 0.2892



- **Most Influential Feature:** 'Head Grade_Ag gpt' is the most impactful feature, which is intuitive. 'Head Grade_Au gpt' and 'Tonnage Processed_tons' have less impact.
- Deviations in 'Head Grade_Ag gpt' primarily influence the predicted silver recovery. Given the lower R-squared, the model's reliance on 'Head Grade_Ag gpt' suggests that other uncaptured factors likely play a significant role.

SHAP model for Silver Ounces

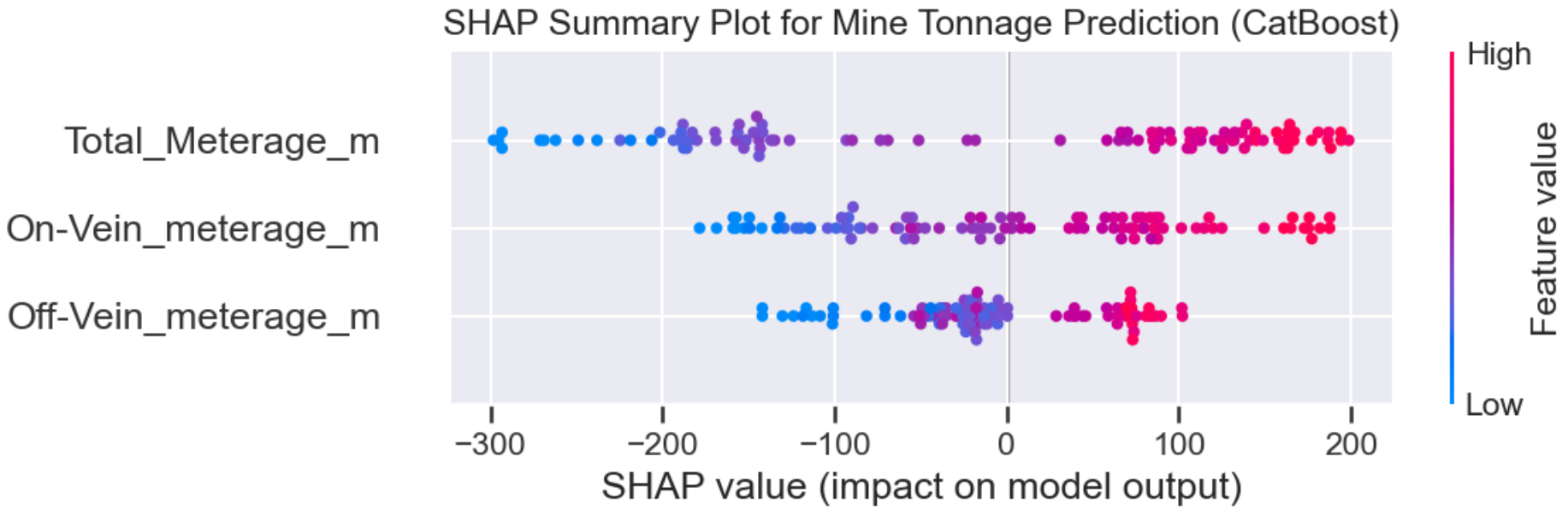
Silver Ounces (tuned)
Artificial Neural Network
R-squared = 0.8020



- **Most Influential Feature:** Similar to gold ounces, all three features contribute significantly.
- 'Head Grade_Ag gpt' and 'Tonnage Processed_tons' are particularly dominant, with higher values leading to increased silver ounces produced.

SHAP model for Mine Tonnage

Mine Tonnage (tuned)
CatBoost
R-squared = 0.2189



- **Most Influential Feature:** 'Total_Meterage_m' and 'Off-Vein_meterage_m' appear to be the most important features. However, the overall low R-squared for this target variable suggests that even these features have limited predictive power.
- 'Total_Meterage_m' contribute the often modest push or pull on the prediction. This reaffirms the earlier observation that the current feature set is insufficient for accurate mine tonnage prediction.

Conclusion and Recommendation: Key Findings and Best Models

- **Gold Ounces Produced:**
  - **Best Model (Tuned):** Artificial Neural Network (R-squared: **0.9851**, RMSE: 8.9568)
  - This target variable is highly predictable, indicating strong relationships with the input features. Both ANNs and ensemble methods (CatBoost, XGBoost) achieved excellent performance.
  - **SHAP Insights:** 'Head Grade_Au gpt' and 'Tonnage Processed_tons' are highly important, showing a strong positive correlation with the output.
  - Out of all the target variables, gold ounces can be predicted most accurately with a RMSE of ±8.9568 oz Au which is low enough error for prediction and can already be used reliably for forecasting purposes.

Conclusion and Recommendation: Key Findings and Best Models

- **Gold Recovery**:
  - **Best Model (Tuned):** XGBoost Regressor (R-squared: **0.7181**, RMSE: 0.0472)
  - Good predictive power was achieved. Ensemble methods (XGBoost, CatBoost, LightGBM) consistently outperformed traditional models.
  - **SHAP Insights:** 'Head Grade_Au gpt' is the most influential feature, positively impacting gold recovery, followed by 'Tonnage Processed_tons' and 'Head Grade_Ag gpt'.
  - Gold recovery can be predicted relatively accurate with a RMSE of ±0.0472% which can still be improved for higher prediction but can already be used reliably as guidance when needed.

Conclusion and Recommendation: Key Findings and Best Models

- **Silver Ounces Produced**:
  - **Best Model (Tuned):** Artificial Neural Network (R-squared: **0.8020**, RMSE: 151.7891)
  - Moderate predictability was observed, better than silver recovery but not as reliable as gold ounces.
  - **SHAP Insights:** 'Head Grade_Ag gpt' and 'Tonnage Processed_tons' are dominant features, leading to increased silver ounces produced when their values are higher.
  - Silver ounces performed can be predicted accurate enough with a RMSE of ±151.7891 oz Ag which can still be improved for higher predictive power but can already be used reliably as guidance when needed.

Conclusion and Recommendation: Key Findings and Best Models

- **Silver Recovery:**
  - **Best Model (Tuned):** Decision Tree Regressor (R-squared: **0.2892**, RMSE: 0.1173)
  - This proved to be the most challenging prediction task. Even after tuning, R-squared values remained low, suggesting the current feature set explains only a small portion of its variance. Traditional models like Linear Regression and Decision Trees performed surprisingly well after tuning compared to some ensemble methods for this specific target.
  - **SHAP Insights:** 'Head Grade_Ag gpt' is the most impactful feature. However, the low R-squared suggests other uncaptured factors play a significant role.
  - Silver Recovery performed not as well as the other target variables only achieving an RMSE of ±0.1173% which can still be improved for higher predictive power but can already be used reliably as guidance when needed
  - A deeper investigation into additional, more relevant features is crucial. This could involve consulting domain experts to identify factors like geological formations, ore body characteristics, equipment operational parameters, or temporal trends that might influence these metrics.

Conclusion and Recommendation: Key Findings and Best Models

- **Mine Tonnage**:
  - **Best Model (Tuned):** Support Vector Machine (R-squared: **0.1995,** RMSE: 588.6655)
  - All models struggled significantly with this prediction. The R-squared values were very low (around 0.20), indicating that the selected predictors (`Off-Vein_meterage_m`, `On-Vein_meterage_m`, `Total_Meterage_m`) explain very little of the variance in 'Mine Tonnage_tons'.
  - **Therefore it is recommended that the current available feature set be expanded to other mine production metrics: such as blast width, stoping height, etc..**
  - **SHAP Insights:** 'Total_Meterage_m' and 'Off-Vein_meterage_m' were identified as important, but the overall low R-squared indicates limited predictive power from the current feature set for this target.
  - Mine Tonnage performed the worst out of all the models achieving an RMSE of ±588.66 tons which is too high of an error to predict it accurately. This only means the model can still be improved a lot by adding other features.
  - A deeper investigation into additional, more relevant features is crucial. This could involve consulting domain experts to identify factors like other mining metrics, equipment operational parameters, or temporal trends that might influence these metrics

Conclusion and Recommendation: Major Data Gaps

- **Time Series Analysis**
  - Time-Series analysis was not considered due to the **broken datasets especially during COVID times** wherein proper data-recording was not done at that time.
  - Time-Series analysis was also considered unsuitable due to the tendency of ARIMA and LSTM models to possibly overshoot projections because it is **not bound by limiting factors** such as number of Mining Equipment, Milling Equipment, Mechanical Availability, etc.
- **Mine Production and Mine Advance Data**
  - Majority of data available are **not synced at the same time-period as the other daily data in the mill**, but it can possibly be summarized or generalized to fit the same time frame as the data used in this study.
- **Data Granularity**
  - As much as some datasets are recorded per-shift, some where not recorded with any time or date stamps therefore they are much harder to index and incorporate in this dataset.
  - Data granularity was set to daily since it is the **most common report interval between the mine and the mill**. A more detailed granularity of this report will probably increase model predictive performance especially for the lower performing models in this capstone project.

Conclusion and Recommendation: Major Insights Gained

- The predictability of gold-related metrics is generally higher than silver-related metrics with the current feature set.
- Predicting 'Mine Tonnage' is significantly difficult with the given features, suggesting they are not the primary drivers of tonnage variations.
- SHAP explanations confirmed the intuitive importance of 'Head Grade_Au gpt' for gold, and 'Head Grade_Ag gpt' for silver, but also highlighted the role of 'Tonnage Processed_tons' across multiple targets. For Mine Tonnage, SHAP reaffirmed the limited influence of the available meterage features.

Conclusion and Recommendation: Major Insights Gained

- The predictive power of a Machine Learning model is dependent on **the strength of the relationships** between the predictor variables with respect to the target variable.
- A **good balance** between model accuracy and the **required predictor variables** are required to come up with a decently accurate prediction.
  - i.e. A model with 10 inputs may give out a better or more accurate prediction than one requiring only 2 to 3 inputs, but having to record 10 inputs will be **much more cumbersome or possibly too expensive and time consuming** as to the weight or impact of the prediction to a company's process.
- The **timeliness of a prediction** must also be considered. If a prediction can only be made too close to when the official results are obtained, decreases the necessity for the prediction since it doesn't give ample time for the company to adapt or change its parameters to improve the outcome of the target variable.
  - This also means that the predictor variables used to forecast **the predictor variable should already have been recorded much earlier** in the process for it to have any meaningful impact to the company.
  - Therefore, a timely prediction for % Recovery will only be meaningful if the recorded input metrics or features **occur prior** to when the target metal/s have already been completely extracted from the ore.

Conclusion and Recommendation: Items for Consideration

- **Feature Engineering for Challenging Predictions:**
  - For 'Silver Recovery' and especially 'Mine Tonnage', a deeper investigation into additional, more relevant features is crucial. This could involve consulting domain experts to identify factors like geological formations, ore body characteristics, equipment operational parameters, or temporal trends that might influence these metrics.
- **Explore Time-Series Models:**
  - Given that all datasets are time-indexed, exploring time-series specific models (e.g., ARIMA, Prophet, LSTMs for sequential data) could capture temporal dependencies that traditional regression models might miss, especially for recovery rates and mine tonnage.
- **Advanced Hyperparameter Optimization:**
  - While `GridSearchCV` was used, other advanced optimization techniques like `RandomizedSearchCV` with broader distributions or Bayesian Optimization may be employed for a more exhaustive search of the hyperparameter space, especially for ANNs and complex ensemble models

Conclusion and Recommendation: Items for Consideration

- **Ensemble Stacking/Blending:**
  - Combining predictions from several best-performing models (e.g., CatBoost, XGBoost, ANN) through stacking or blending could potentially yield even better performance for each target variable.
- **Further SHAP Analysis:**
  - Investigate SHAP dependence plots for critical features to understand how a feature's value affects the prediction when interacting with other features. This could provide deeper insights into the underlying process dynamics.
- **Data Quality and Granularity:**
  - Assess if more granular or higher-quality data points are available, particularly for the challenging predictions. The low R-squared values for 'Mine Tonnage' might suggest that the current features are too high-level or miss crucial information that will improve model accuracy.

Thank You.

Jose Norbiel G. Florendo

AIM-PGDAIML

josenorbielflorendo@gmail.com