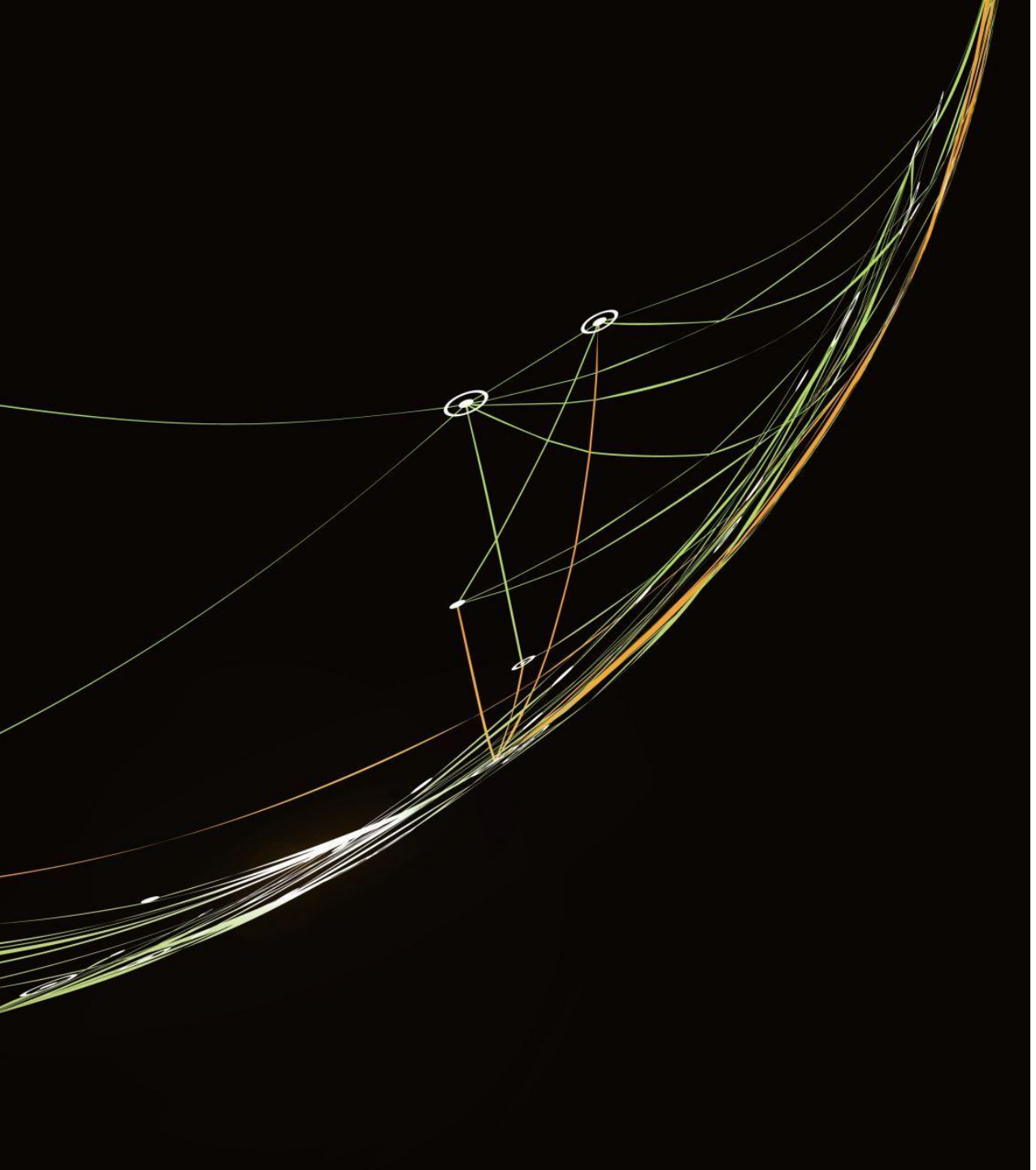


Identification of Poor Performing Products for Dilliard's

By Group 4:

- Jason Huang
- Rohit Sharma
- Kexian Wu
- Kate Yee

Executive Summary

Predicted the amount spent in future transactions in order to pinpoint products with low profit percentages.

Focused our analysis on transactions:
- within Texas
- max original price threshold of $49.5
- categorized by original price into two groups: 'Cheap' and 'Normal.'

Performed a ROI analysis predicting that when we exclude the bottom 30% of products from each store, we make a significant increase in profit of 2.93 times as compared to our baseline results.

Introduction

- Dilliard's is a prominent department store chain, which sells clothing, lifestyle, and home essentials products across the United States
- Point of sale data provided in five tables: strinfo, skstinfo, skuinfo, trnsact, and deptinfo
- Project Steps:
  - Perform exploratory data analysis
  - Choose a profit based machine learning question
  - Devise models to answer our chosen question
  - Conduct an ROI analysis

Data Processing

- Point of sale data supplied as a collection of csv files
- Imported data into a PostgreSQL database using pgAdmin for centralized data access and queries
- Resolved issues with skuinfo vendor names containing commas by filtering out around 8000 rows to maintain data integrity
- Used pandas and psycopg2 within python Jupyter notebooks to access the datasets

Exploratory Data Analysis by Location

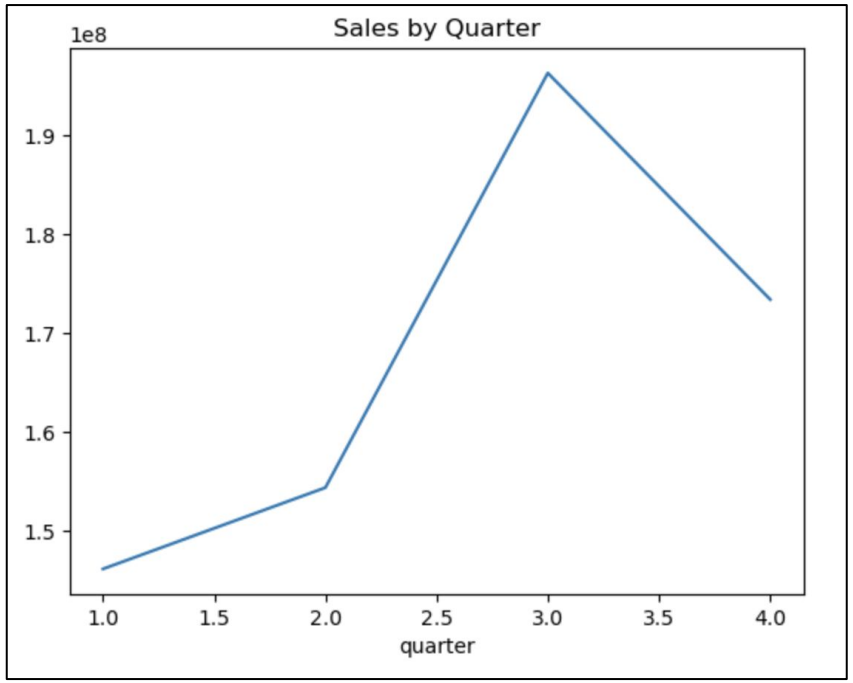
- Began by exploring the trnsact table
  - difficult to query due to 120,916,896 rows
  - solved by randomly sampling one million rows for exploratory analysis
- Merged our sampled trnsact dataframe with the strinfo table to view how stores are distributed across states, cities, and zip codes
- Discovered that Texas was the state with the greatest sum of transaction 'amt's
- Decided to focus our analysis on Texas transactions to segment transactions into a digestible amount, while still analyzing the largest state demographic

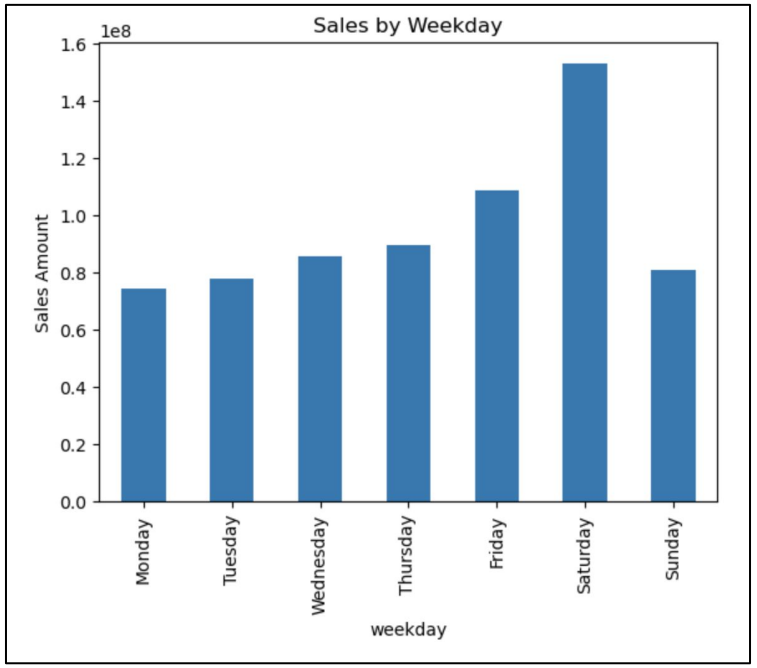
Sampled Total Transaction Amount by State

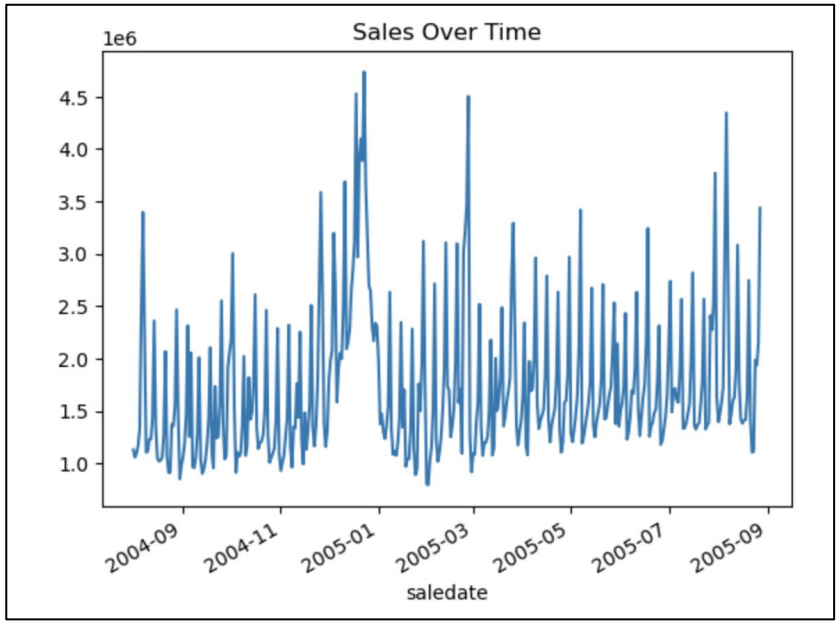
state	
TX	5564009.12
FL	3435391.28
LA	1420495.11
OH	1324056.37
AZ	1255785.47
TN	952154.98
OK	881879.30
MO	877461.90
AR	840495.58
AL	779322.11

Exploratory Data Analysis by Saledate

- Saw a significant spike around December graphing saledate distribution over time
- Grouped 'saledate' by weekday, month, and quarter
- Kept those 'salesdate' classifications in mind for later feature engineering





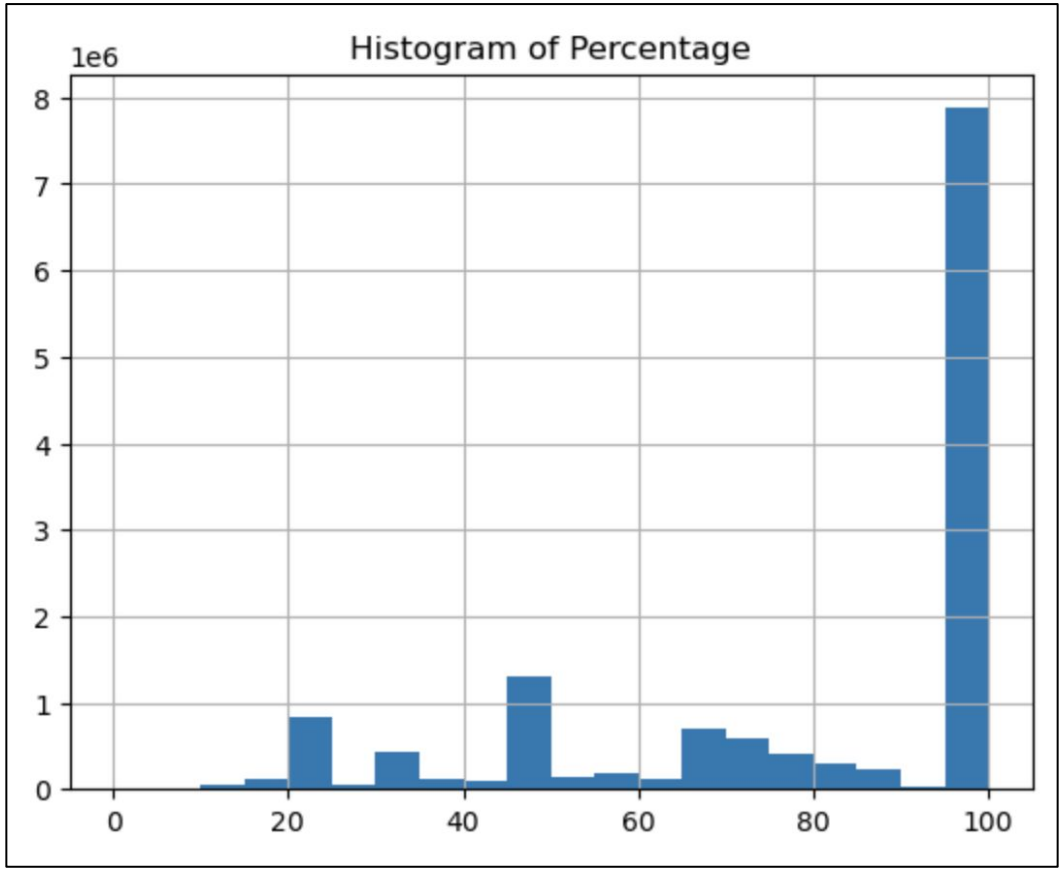


Exploratory Data Analysis by Discount

$$'discount\ percentage' = \frac{'amt'}{'orgprice'} \times 'quantity' \times 100$$

$$'profit' = ('amt' - 'cost') \times 'quantity'$$

- Created discount percentage and profit columns
- Removed rows that were missing necessary values
- Decided to exclude transactions with excessively high discounts, i.e. selling items at prices less than 3% of the original price
- Observed many instances of transactions with exceptionally high discount rates
- Considered the possibility of predicting the profit margin before stocking these items

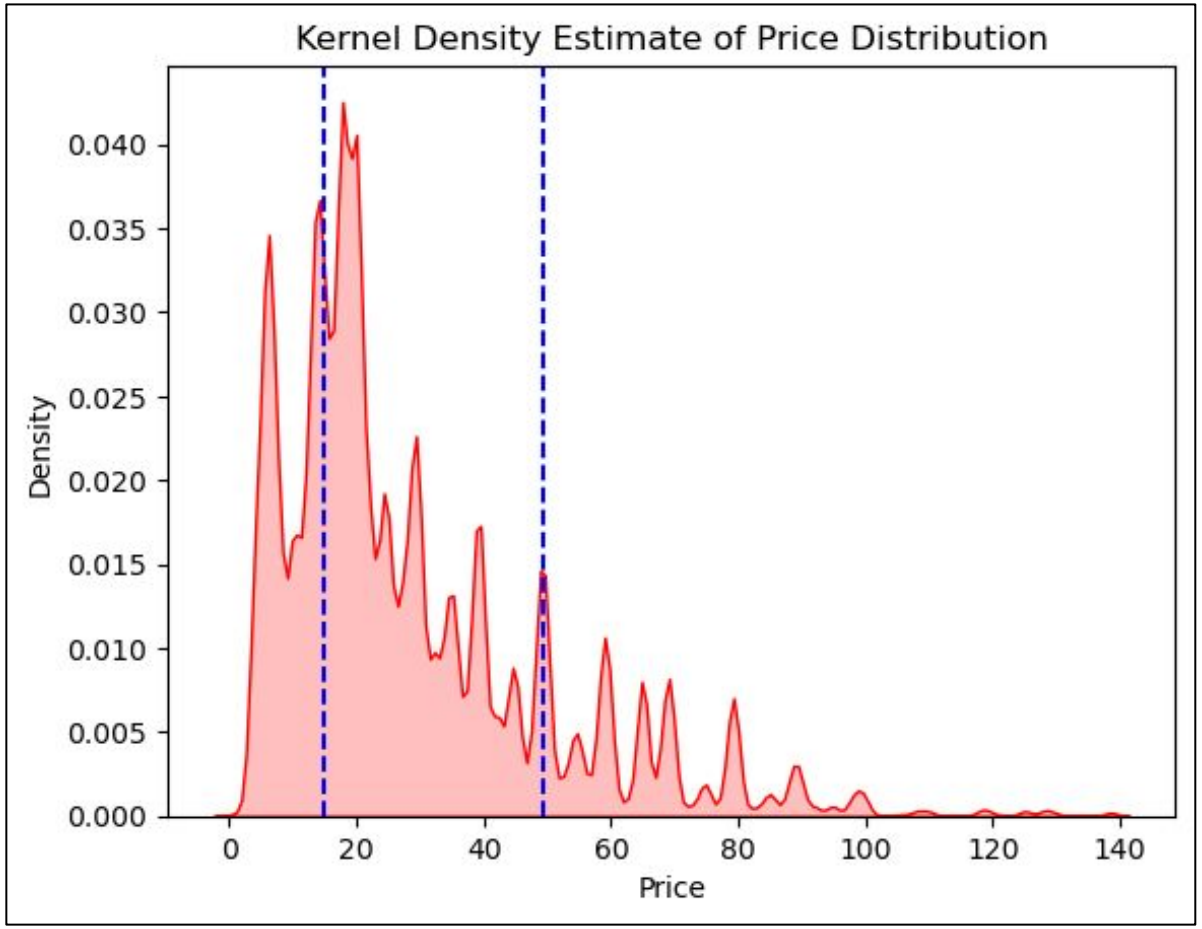


Final Data Cleaning

- Removed all rows of 'stype' R to only focus on purchase transactions
- Removed outliers
  - Filtered by 'quantity', 'orgprice', 'packsize', and 'cost' on a .99 quantile
  - Excluded rows where a single transaction is associated with a specific ('sku', 'store') combination
- Pickled the resulting dataframe to be used for the feature engineering and modeling steps of our process

Feature Engineering and Pipeline

- Dataset trimmed using a $49.5 threshold for the 'org-price' feature, focusing on Dilliard's stronger presence in lower-priced product categories
- Redundant features dropped to prevent overfitting; department description used due to its consistency
- 'saledate' replaced with derived features like year, month, weekday, and holidays for better prediction
- Lagged features for 'amt' and 'orgprice' included to capture historical pricing trends
- Feature engineering pipeline established, standardizing numerical variables and employing one-hot encoding for categorical data



Modeling

- Main objective: employ SKU features and transaction history as key variables to predict the target variable, 'amt'
- Our hypothesis: by identifying and eliminating less profitable products ahead of time, we can help the company reduce the cost of purchasing those products and the loss of selling them, as they will not contribute significantly/positively to the company's profitability

Modeling Process

Built machine-learning models to predict sales:
- Linear Regression
- Lasso Regression
- Random Forest

Incorporated a grid search process to optimize the models

Random Forest models included a grid search to determine the optimal number of trees and the maximum depth of the trees

Evaluated the models based on:
- Mean Squared Error
- R-squared
- Mean Absolute Error

Results and Comparative Analysis

- **Lasso Regression**: Controls overfitting and aids in feature selection by adjusting alpha and fit_intercept parameters
- **Random Forest Implementation**: Selected for handling complex datasets, optimized with grid search for n_estimators and tree depth.
- **Category-Specific Models**: Developed individual models for 'Cheap' (orgprice <= $15) and 'Normal' ($15 < orgprice < $49.5) price categories due to low R-squared value in the single model approach.
- **Variable Performance of Linear Regression**: High R-squared values in 'Cheap' category, indicating good fit.
- **Lasso Regression's Superiority**: Reduced overfitting and identified significant features, performing better in both price categories.
- **Random Forest vs. Linear Models**: Unexpectedly, Random Forest didn't outperform linear models despite its robustness in handling complex relationships.
- **Best Model**: Individual Lasso models for 'cheap' and 'normal' categories showed the best performance, used for estimating profit margin.

	Model	Mean Squared Error	R-squared	Mean Absolute Error
0	Linear Regression (all)	30.558	0.669	3.283
1	Linear Regression (cheap)	2.260	0.862	0.928
2	Linear Regression (normal)	42.932	0.490	4.229
3	Lasso Regression (all)	30.523	0.670	3.275
4	Lasso Regression (cheap)	2.251	0.862	0.919
5	Lasso Regression (normal)	42.842	0.491	4.219
6	Random Forest (all)	40.048	0.567	3.244
7	Random Forest (cheap)	2.539	0.845	0.792
8	Random Forest (normal)	54.201	0.356	4.145

ROI Analysis

- Our ROI analysis perfectly examines the financial outcomes of inventory management decisions and uses the results from our lasso regression model
- Our analysis reveals a clear pivot point at the 0.3 quantile threshold, where profit shifts from negative to positive
- Our baseline model estimates performance of previously non-profitable products and involves eliminating the bottom % of underperforming items in each store by targeting pairs during training, and applying the same criteria in test sets
- With our proposed model we concluded a rise in profit of 2.93 folds, as indicated in our ROI table, with better returns at an earlier threshold compared to the baseline model values
- Our targeted approach promises a better ROI by achieving profitability at a lower quantile threshold and minimizes opportunity cost

Baseline Model	**Profitability cutoff value is 0.4**				
Lasso Regression Model	**Profitability cutoff value is 0.3**				
Threshold	**0.3**	To evaluate returns based on our proposed model			
RETURNS					
	BASELINE MODEL		**PROPOSED MODEL**		
Test set comparison					
Cost of underperforming items	$ 229,662		$ 981,396		
Revenue of underperforming items	$ 217,017		$ 1,005,915		
Percent increase in revenue between baseline and proposed models			363.52%		
Profit of underperforming items	$ (12,645)		$ 24,520		
Difference (proposed - baseline)			$ 37,165		
Percent increase in profit between baseline and proposed models			293.91%		
Previous month cost	$ 3,052,840		$ 2,071,445		
Previous month profit	$ 1,632,529		$ 1,608,009		
Based on this analysis, we observe a significant percent profit increase of 2.93 folds between the baseline and our lasso model, which suggest that excluding the bottom 30% of products for each store results in higher revenue and minimized opportunity costs					
INVESTMENTS				1/12 Years	$ 14,515
LABOR COSTS					
	Annual wage		**FTE**	**Project duration (years)**	**Total**
Data Scientist	$ 120,000.00		1	0.25	$ 30,000.00
Manager	$ 175,000.00		1	0.25	$ 43,750.00
Model maintenance					
Data Engineer	$ 100,000.00		1	1.00	$ 100,000.00
INFRASTRUCTURE COST					
	Cloud hourly rate		**Computing hours/day**	**Days computing***	**Total**
AWS	$ 0.10		24	180	$ 432.00