

Introduction.....	3
Linear Programming and Data Analysis for Optimal Revenue.....	3
Key Findings and Insights	5
Optimisation Model Results	5
Historical Data Analysis Results.....	6
Revenue Management Implementation Strategy.....	7
Conclusion.....	8
Appendices.....	9

Introduction

The development of a Revenue Management (RM) strategy is highly applicable to industries with complex operational challenges and constraints, including cyclical demand fluctuations, logistical complexity, perishable inventory and diverse customer segments across industries. BuildMax Rentals a leading provider in Europe and North America, operates a fleet of cranes, excavators, and bulldozers across 80 branches. While the company does not operate with perishable goods in the traditional sense, idle equipment incurs opportunity costs like vacant hotel rooms or empty airline seats. Car rental companies face similar fleet management and pricing challenges, despite having greater flexibility in redistributing vehicles across locations. However, BuildMax Rentals must contend with longer rental durations, servicing downtime, and location-specific inventory constraints, making fleet optimisation more complex. Serving construction firms, municipal projects, and industrial operations, BuildMax must balance short-term availability with long-term commitments while adapting to diverse customer preferences and price sensitivities.

Implementing RM at BuildMax Rentals presents several challenges. Seasonality and demand variability play a significant role, as construction equipment rentals are influenced by cyclical demand patterns linked to construction projects and broader economic conditions, unlike airlines and hotels, where demand is driven by tourism and events. Inventory constraints pose a challenge, as equipment availability cannot be instantly increased like additional flights or hotel rooms. Fleet allocation complexity, unlike airline seating where capacity is fixed, BuildMax must actively reposition, maintain, and manage its fleet, making real-time revenue management decisions more intricate.

This report assesses the effectiveness of the current strategy by evaluating its potential to maximise revenue through a linear programming (LP) model using Python while simultaneously analysing operational performance. The analysis focuses on demonstrating how RM principles optimise key areas, including pricing, seasonality, fleet utilisation and allocation. By combining techniques with data-driven insights, the report highlights the impact of RM on overall efficiency and profitability while identifying areas for further refinement and strategic improvement.

Linear Programming and Data Analysis for Optimal Revenue

The analysis combines linear programming optimisation with historical data evaluation to examine demand patterns, including seasonal fluctuations and equipment-specific preferences.

It also identifies utilisation challenges such as inventory constraints and the trade-offs between short-term and long-term rentals.

A LP model was developed to optimise fleet allocation and maximise revenue while accounting for capacity constraints and demand fluctuations (Appendix 1). The model ensures that rental decisions strategically balance short-term and long-term bookings, improve fleet utilisation and reduce lost revenue from unfulfilled requests. With this, BuildMax can make data-driven allocation decisions, ensuring profitability without compromising equipment availability.

A) Decision Variables

The LP model defines key decision variables representing rental allocation and pricing decisions:

- $X_{i,t,j}$: The units of equipment type i accepted for a t -week rental order in week j .
- $I_{i,j}$: The inventories available for equipment type i at week j .

B) Objective Function

The primary objective of the model is to maximise total revenue across all equipment types and rental durations. The revenue function is defined as follows:

$$\max R = \sum_{j=1}^{52} \sum_i \sum_t P_{i,t,j} \times 7 \times X_{i,t,j} \times t$$

- Constant $P_{i,t,j}$: the price per day of each equipment i for a t -week rental order in week j .

C) Constraints

To ensure practical feasibility, the constraints allow the model to control fleet allocation and reflect BuildMax's operational and logistical limitations, ensuring that the revenue potential is maximised with real-conditioned limitations.

1. Demand Constraint

The units of equipment BuildMax accepted to rent out cannot exceed the demand of each type of equipment for each duration.

$$X_{i,t,j} \leq D_{i,t,j} \quad \forall i, t, j$$

2. Capacity Constraints

The units of equipment BuildMax accepted to rent out cannot exceed the available inventory (start of week inventory) for each week.

$$\sum_t X_{i,t,j} \leq I_{i,j} \forall i, t, j$$

Inventory Balance Equation

To ensure the inventory at the start of the week is updated correctly, the equation is applied to explain the dynamic changing process.

Week 1: the start of week inventories for week 1 is the initial inventory.

Week j from 2 to 52: The start of week inventory at week j is equal to the start of week inventory at week j-1, subtract the sum of units rented out at week j-1 for all types of equipment, plus the total units returned timely in week j that are rented out at week j-t given that j-t ≥ 1.

$$I_{i,j} = I_{i,j-1} - \sum_t X_{i,t,j-1} + \sum_t X_{i,t,j-t} \forall i, t, j$$

3. Non-Negativity Integer Constraint

$$X_{i,t,j}, I_{i,j} \geq 0, \text{ integer}$$

Key Findings and Insights

Optimisation Model Results

The model allocates rentals to maximise total revenue and improve fleet utilisation efficiency. The model achieved a total revenue of £178,478,636, representing a 11.22% increase from the original revenue of £160,473,614 (Appendix 3). This outcome highlights the substantial impact of capacity-constrained RM in boosting profitability while ensuring that fleet availability remains balanced. Additionally, ROI improved by 15.47%, demonstrating that revenue gains were achieved without the need for fleet expansion, shifting focus on better allocation of existing resources.

Furthermore, revenue distribution across equipment types indicates notable shifts. Cranes generate the highest revenue per unit (RP), significantly outperforming that of Excavators' and

Bulldozers'. The increase of accepting rate (Appendix 2) indicates that equipment types with higher RPU should be accepted more to increase utilisation rate and to guarantee better benefits as far as the inventories allows.

The trade-off between long-term and short-term leases is a significant insight. The accepting rate for long-term lease requests highly increases. The 1-week rental accepting rate (-0.38%) is minimal, while the three long-term rentals have an average of 10% increase. Contractors prefer long-term leases, and the model guides BuildMax to accept more long-term requests to optimise revenue. However, short-term rental provides more flexibility for inventories to avoid accidents such as late returns or sharp decrease in inventories due to homochromous long-term rentals.

Historical Data Analysis Results

An analysis of BuildMax's rental data reveals clear pricing inefficiencies and demand trends that impact fleet allocation and revenue potential. Despite higher daily rates, short-term rentals are underutilised, as the business prioritises long-term leases to stabilise fleet utilisation. While this strategy ensures consistent income, it limits inventory turnover, particularly during peak seasons when short-term demand surges. Seasonal trends indicate that demand peaks in Q2 and Q3, driven by increased construction activity, whereas Q1 and Q4 experience lower rental volumes due to weather conditions and project slowdowns. The optimisation model reinforced this pattern, demonstrating that increasing short-term rental availability during peak periods would allow BuildMax to capitalise on higher-margin opportunities, while off-peak discounts on long-term rentals would sustain year-round utilisation.

Fleet utilisation (Appendix 4) varies across equipment types. Excavators maintain the highest utilisation rate at 80%, reflecting strong demand but also an overcommitment to long-term leases, restricting their availability for high-turnover rentals. Cranes, despite generating the highest revenue, operate at only 68.95% utilisation, indicating potential for improved fleet allocation. Bulldozers, with a lower utilisation rate of 49.85%, appear oversupplied in some locations, suggesting that reallocation to high-demand regions would enhance efficiency. Additionally, servicing delays and uneven maintenance schedules contribute to underutilisation, with equipment availability fluctuating based on unpredictable servicing cycles rather than planned fleet turnover.

Revenue Management Implementation Strategy

BuildMax should consider integrating advanced forecasting, dynamic pricing, and real-time fleet tracking into its operations. Implementing a machine learning-enabled demand forecasting model will improve rental predictions, allowing for seasonal price adjustments and real-time fleet reallocation based on projected demand fluctuations. By increasing short-term rental rates during peak seasons and offering long-term discounts in off-peak months, BuildMax can maximise revenue potential while maintaining fleet stability.

The implementation strategy should follow a phased rollout, beginning with eight pilot locations representing 10% of BuildMax's total branches. These locations should be selected based on high rental volume, strong seasonal fluctuations, and fleet imbalances, allowing for controlled testing before full-scale deployment. Standardising pricing strategies, fleet allocation methods, and forecasting models across locations will ensure a consistent and scalable RM framework that aligns with BuildMax's operational goals.

While the RM model optimises revenue allocation, certain limitations must be addressed. High-value ad-hoc rental requests may still be rejected due to fleet constraints, requiring manual intervention to assess strategic opportunities. Additionally, contractors may resist dynamic pricing adjustments, necessitating gradual implementation and transparent communication to ease adaptation. Forecasting uncertainty remains a challenge, as external factors such as economic downturns and extreme weather may impact model accuracy. Continuous monitoring and iterative refinements will be necessary to ensure that the system evolves with changing market conditions.

To ensure the successful adoption of Revenue Management, BuildMax must integrate a structured People, Process, and Technology (PPT) framework. Staff training is also essential in equipping employees to interpret RM outputs, make data-driven rental decisions, and manage fleet allocation effectively. Standardising processes across all branches will streamline dynamic pricing, fleet tracking, and predictive maintenance scheduling, ensuring operational consistency.

Conclusion

BuildMax Rentals has the opportunity to achieve significant revenue growth and operational efficiency through the implementation of a Revenue Management framework. The integration of linear programming and data analytics demonstrates clear potential to optimise revenue, fleet utilisation, and rental allocations, while refinements are needed to address servicing delays, fleet imbalances, and the management of ad-hoc rental requests. By strategically reallocating high-margin equipment, refining pricing strategies, and streamlining logistics, BuildMax can maximise profitability. A phased pilot rollout will ensure smooth adoption, projecting a 15% first-year revenue uplift and reinforcing market leadership.

Appendices

Appendix 1: Python Code for Linear Optimisation Model

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Created on Thu Mar  6 12:47:37 2025
5
6  @author: Group 8
7  """
8
9  import numpy as np
10 import pandas as pd
11 from pyomo.environ import *
12
13 # Load dataset
14 file_path = "BuildMax_Rentals_Dataset_Updated.xlsx"
15 df = pd.read_excel(file_path, sheet_name="Sheet1")
16
17 # Define equipment types and rental durations
18 equipment_types = ["Excavators", "Cranes", "Bulldozers"]
19 durations = [1, 4, 8, 16] # Rental durations in weeks
20 num_weeks = 52
21
22 # Extract initial inventory
23 initial_inventory = {eq: df[f'{eq} - Start of Week Inventory'].iloc[0] for eq in equipment_types}
24
25 # Extract prices, demand, and returns with (i, t, j) indexing
26 prices = {(eq, t, j+1): df[f'{eq} - {t}-week Price per day (€)'].iloc[j]
27             if t == 1 else df[f'{eq} - {t}-weeks Price per day (€)'].iloc[j]
28             for eq in equipment_types for t in durations for j in range(num_weeks)}
29
30 demand = {(eq, t, j+1): df[f'{eq} - {t}-week Demand (units)'].iloc[j]
31             if t == 1 else df[f'{eq} - {t}-weeks Demand (units)'].iloc[j]
32             for eq in equipment_types for t in durations for j in range(num_weeks)}
33
34 # Define model
35 model = ConcreteModel()
36
37 # Decision Variables
38 model.X = Var(equipment_types, durations, range(1, num_weeks+1), domain=NonNegativeIntegers) #the units accepted to optimize revenue
39 model.I = Var(equipment_types, range(1, num_weeks+1), domain=NonNegativeIntegers) # Inventory tracking
40
41 # Objective Function: Maximize Revenue
42 def revenue_rule(model):
43     return sum(
44         t * prices[i, t, j] * model.X[i, t, j]
45         for i in equipment_types for t in durations for j in range(1, num_weeks+1)
46     )
47
48 model.obj = Objective(rule=revenue_rule, sense=maximize)
49
50 # Inventory Balance Constraint (Returns = Rentals from t Weeks Ago)
51 def inventory_rule(model, i, j):
52     if j == 1:
53         return model.I[i, j] == initial_inventory[i] # Week 1: Start with initial inventory
54     else:
55         # Rentals accepted in the previous week (j-1) are subtracted
56         # Returns occur for rentals accepted t weeks ago
57         returns_total = sum(model.X[i, t, j-t] for t in durations if (j-t) >= 1)
58
59     return model.I[i, j] == model.I[i, j-1] - sum(model.X[i, t, j-1] for t in durations) + returns_total
60
61 model.inventory_constraint = Constraint(equipment_types, range(1, num_weeks+1), rule=inventory_rule)
62
63 # Demand Constraint
64 def demand_rule(model, i, t, j):
65     return model.X[i, t, j] <= demand[i, t, j]
66
67 model.demand_constraint = Constraint(equipment_types, durations, range(1, num_weeks+1), rule=demand_rule)
68
69 # Capacity Constraint
70 def capacity_rule(model, i, j):
71     return sum(model.X[i, t, j] for t in durations) <= model.I[i, j]
72
73 model.capacity_constraint = Constraint(equipment_types, range(1, num_weeks+1), rule=capacity_rule)
74
75 # Solve model
76 solver = SolverFactory("glpk")
77 solver.solve(model)
78
79 # Print results in the required format
80 print(f"Total optimized revenue: {model.obj():.2f}")
81
82 for i in equipment_types:
83     for t in durations:
84         total_rentals = sum(model.X[i, t, j] for j in range(1, num_weeks+1))
85         print(f"{i} - {t}-week rentals: {total_rentals:.1f}")
86

```

```

87  # ----- ANALYSIS SECTION ----- #
88
89 # Revenue per Equipment Type
90 revenue_equipment_type = {eq: sum(
91     t * 7 * prices[eq, t, j] * model.X[eq, t, j]()
92     for t in durations for j in range(1, num_weeks+1))
93     for eq in equipment_types}
94
95 for eq in equipment_types:
96     print(f"Total Revenue {eq}: {revenue_equipment_type[eq]:,.2f}")
97
98 # Revenue per Duration and Equipment Type
99 revenue_per_duration_eq = {
100     (eq, t): sum(
101         t * 7 * prices[eq, t, j] * model.X[eq, t, j]()
102         for j in range(1, num_weeks+1))
103     for eq in equipment_types for t in durations}
104
105 # Print Revenue per Duration and Equipment Type
106 print("\nRevenue per Duration and Equipment Type:")
107 for (eq, t), revenue in revenue_per_duration_eq.items():
108     print(f"{eq} - {t}-week rentals: {revenue:,.2f}")
109
110 # Compute Weekly Fleet Utilization Rate
111 weekly_utilization = {
112     eq: [
113         (sum(model.X[eq, t, j] for t in durations) / model.I[eq, j]) * 100
114         if model.I[eq, j] > 0 else 0 # Avoid division by zero
115         for j in range(1, num_weeks+1)
116     ]
117     for eq in equipment_types
118 }
119
120 # Compute Average Fleet Utilization Rate across all weeks
121 avg_fleet_utilization = {
122     eq: sum(weekly_utilization[eq]) / num_weeks
123     for eq in equipment_types
124 }
125
126 # Print Results
127 print("\n--- Fleet Utilization Rate ---")
128 for eq in equipment_types:
129     print(f"{eq}: {avg_fleet_utilization[eq]:.2f}%")
130
131 # Compute Total Rentals and Total Inventory for each equipment type
132 total_rentals = {
133     eq: sum(sum(model.X[eq, t, j] for t in durations) for j in range(1, num_weeks+1))
134     for eq in equipment_types
135 }
136
137 total_inventory = {
138     eq: sum(model.I[eq, j] for j in range(1, num_weeks+1))
139     for eq in equipment_types
140 }
141

```

```

142 # Compute Overall Fleet Utilization Rate
143 total_rentals_all_eq = sum(total_rentals[eq] for eq in equipment_types)
144 total_inventory_all_eq = sum(total_inventory[eq] for eq in equipment_types)
145
146 overall_utilization_rate = (total_rentals_all_eq / total_inventory_all_eq) * 100
147
148 # Print Results
149 print(f"Overall Fleet Utilization Rate: {overall_utilization_rate:.2f}%")
150
151 # Revenue per Unit (RPU)
152 rpu = {
153     eq: revenue_equipment_type[eq] / initial_inventory[eq]
154     for eq in equipment_types
155 }
156
157 for eq in equipment_types:
158     print(f"Revenue per {eq}: £{rpu[eq]:.2f}")
159
160 # Define equipment purchase costs for each equipment type (in £)
161 equipment_costs = {
162     "Excavators": 12000,
163     "Cranes": 15000,
164     "Bulldozers": 25000
165 }
166
167 original_total_revenue = {
168     "Excavators": 45274005,
169     "Cranes": 61785689,
170     "Bulldozers": 53413920
171 }
172
173 # Calculate Original RPU for each equipment type
174 original_rpu = {
175     eq: original_total_revenue[eq] / initial_inventory[eq]
176     for eq in equipment_types
177 }
178
179 # Print the Original RPU for each equipment type
180 for eq in equipment_types:
181     print(f"Original RPU for {eq}: £{original_rpu[eq]:.2f}")
182
183 # Calculate Total Equipment Cost for each equipment type (Inventory * Cost)
184 total_equipment_cost = {
185     eq: initial_inventory[eq] * equipment_costs[eq]
186     for eq in equipment_types
187 }
188
189 # Calculate Original Total Revenue (Sum of original revenues)
190 original_total_revenue_all = sum(original_total_revenue.values())
191
192 # Calculate Optimized Total Revenue (Revenue from model)
193 optimized_total_revenue_all = sum(revenue_equipment_type.values())
194
195 # Calculate Original ROI using the formula: (Total Revenue - Total Equipment Cost) / Total Equipment Cost * 100
196 original_roi_all = original_total_revenue_all - sum(total_equipment_cost.values())
197 original_roi_percentage = (original_roi_all / sum(total_equipment_cost.values())) * 100
198
199 # Calculate Optimized ROI using the formula: (Optimized Revenue - Total Equipment Cost) / Total Equipment Cost * 100
200 optimized_roi_all = optimized_total_revenue_all - sum(total_equipment_cost.values())
201 optimized_roi_percentage = (optimized_roi_all / sum(total_equipment_cost.values())) * 100
202
203 # Calculate ROI Improvement
204 roi_improvement_percentage = optimized_roi_percentage - original_roi_percentage
205
206 # Print Results
207 print(f"\nOriginal Total Revenue: £{original_total_revenue_all:.2f}")
208 print(f"Optimized Total Revenue: £{optimized_total_revenue_all:.2f}")
209 print(f"Total Equipment Cost: £{sum(total_equipment_cost.values()):.2f}")
210 print(f"Original ROI: {original_roi_percentage:.2f}%")
211 print(f"Optimized ROI: {optimized_roi_percentage:.2f}%")
212 print(f"ROI Improvement: {roi_improvement_percentage:.2f}%")

```

LP Model Results

Total optimized revenue: 178,478,636.00
 Excavators - 1-week rentals: 1104.0
 Excavators - 4-week rentals: 1178.0
 Excavators - 8-week rentals: 1435.0
 Excavators - 16-week rentals: 1437.0
 Cranes - 1-week rentals: 1183.0
 Cranes - 4-week rentals: 1192.0
 Cranes - 8-week rentals: 1442.0
 Cranes - 16-week rentals: 1561.0
 Bulldozers - 1-week rentals: 1412.0
 Bulldozers - 4-week rentals: 1385.0
 Bulldozers - 8-week rentals: 1324.0
 Bulldozers - 16-week rentals: 1476.0

Total Revenue Excavators: £49,866,264.00
Total Revenue Cranes: £69,471,738.00
Total Revenue Bulldozers: £59,140,634.00

Revenue per Duration and Equipment Type:
Excavators - 1-week rentals: £1,795,388.00
Excavators - 4-week rentals: £7,134,036.00
Excavators - 8-week rentals: £15,665,496.00
Excavators - 16-week rentals: £25,271,344.00
Cranes - 1-week rentals: £2,404,570.00
Cranes - 4-week rentals: £9,744,224.00
Cranes - 8-week rentals: £20,664,896.00
Cranes - 16-week rentals: £36,658,048.00
Bulldozers - 1-week rentals: £2,642,962.00
Bulldozers - 4-week rentals: £9,720,592.00
Bulldozers - 8-week rentals: £16,487,352.00
Bulldozers - 16-week rentals: £30,289,728.00

--- Fleet Utilization Rate ---

Excavators: 79.99%
Cranes: 68.95%
Bulldozers: 49.85%
Overall Fleet Utilization Rate: 47.23%
Revenue per Excavators: £65,613.51
Revenue per Cranes: £83,700.89
Revenue per Bulldozers: £65,711.82
Original RPU for Excavators: £59,571.06
Original RPU for Cranes: £74,440.59
Original RPU for Bulldozers: £59,348.80

Original Total Revenue: £160,473,614.00
Optimized Total Revenue: £178,478,636.00
Total Equipment Cost: £44,070,000.00
Original ROI: 264.13%
Optimized ROI: 304.99%
ROI Improvement: 40.86%

Appendix 2: Demand Related Tables

Demand Table

	Excavator	Crane	Bulldozer	Total
1-Week	1397	1350	1553	4300
4-Week	1273	1242	1420	3935
8-Week	1485	1442	1324	4251
16-Week	1453	1578	1476	4507
Total	5608	5612	5773	16993

Actual Accepted Units Table

	Excavator	Crane	Bulldozer	Total
1-Week	1186	1159	1368	3713
4-Week	1074	1036	1213	3323
8-Week	1253	1230	1164	3647
16-Week	1325	1408	1360	4093
Total	4838	4833	5105	14776

Actual Accepted Rate Table

	Excavator	Crane	Bulldozer	Total
1-Week	84.90%	85.85%	88.09%	86.35%
4-Week	84.37%	83.41%	85.42%	84.45%
8-Week	84.38%	85.30%	87.92%	85.79%
16-Week	91.19%	89.23%	92.14%	90.81%
Total	86.27%	86.12%	88.43%	86.95%

Optimised Accepted Units

	Excavator	Crane	Bulldozer	Total
1-Week	1104	1183	1412	3699
4-Week	1178	1192	1385	3755
8-Week	1435	1442	1324	4201
16-Week	1437	1561	1476	4474
Total	5154	5378	5597	16129

Optimised Accepted Rate

	Excavator	Crane	Bulldozer	Total

1-Week	79.03%	87.63%	90.92%	86.02%
4-Week	92.54%	95.97%	97.54%	95.43%
8-Week	96.63%	100.00%	100.00%	98.82%
16-Week	98.90%	98.92%	100.00%	99.27%
Total	91.90%	95.83%	96.95%	94.92%

Accepted Rate Comparison

Difference rate= (Optimised accepted rate- Actual accepted rate)/Actual rate

	Excavator	Crane	Bulldozer	Total
1-Week	-6.91%	2.07%	3.22%	-0.38%
4-Week	9.68%	15.06%	14.18%	13.00%
8-Week	14.53%	17.24%	13.75%	15.19%
16-Week	8.45%	10.87%	8.53%	9.31%
Total	6.53%	11.28%	9.64%	9.16%

Appendix 3: Revenue Related Tables

Actual Revenue Table

	Excavator	Crane	Bulldozer	Total
1-Week	£1,916,537.00	£2,358,937.00	£2,522,184.00	£6,797,658.00
4-Week	£6,427,988.00	£8,612,632.00	£8,475,264.00	£23,515,884.00
8-Week	£13,657,560.00	£17,668,280.00	£14,450,968.00	£45,776,808.00
16-Week	£23,271,920.00	£33,145,840.00	£27,965,504.00	£84,383,264.00
Total	£45,274,005.00	£61,785,689.00	£53,413,920.00	£160,473,614.00
RPU	£59,571.06	£74,440.59	£59,348.80	

Optimised Revenue Table

	Excavator	Crane	Bulldozer	Total
1-Week	£1,795,388.00	£2,404,570.00	£2,642,962.00	£6,842,920.00
4-Week	£7,134,036.00	£9,744,224.00	£9,720,592.00	£26,598,852.00
8-Week	£15,665,496.00	£20,664,896.00	£16,487,352.00	£52,817,744.00
16-Week	£25,271,344.00	£36,658,048.00	£30,289,728.00	£92,219,120.00
Total	£49,866,264.00	£69,471,738.00	£59,140,634.00	£178,478,636.00
RPU	£65,613.51	£83,700.89	£65,711.82	

Revenue Comparison

	Excavator	Crane	Bulldozer	Total
1-Week	-6.32%	1.93%	4.79%	0.67%
4-Week	10.98%	13.14%	14.69%	13.11%
8-Week	14.70%	16.96%	14.09%	15.38%
16-Week	8.59%	10.60%	8.31%	9.29%
Total	10.14%	12.44%	10.72%	11.22%

Appendix 4:

Weekly fleet utilization rate = $(\sum_t x_{i,t,j} \div I_{i,j}) \times 100\%$

Average fleet utilization rate = $\frac{(\sum_{j=1}^{52} \text{weekly fleet utilization rate})}{52}$

Excavator	Crane	Bulldozer	Overall
79.99%	68.95%	49.85%	47.23%