

Challenge September 2024

Boat Rental

A solution with IBM Watson AI (CPLEX and LLM Llama3 by Meta)
by Alex Fleischer afleischer@fr.ibm.com

CPLEX and Llama3 by Meta are available within IBM AI platform
watsonx.AI

We can start with the following prompt:

Floataway tours has \$420,000 that may be used to purchase new rental boats for hire during the summer. The boats can be purchased from two different manufacturers. Floataway tours would like to purchase at least 50 boats and would like to purchase the same number from Sleekboat as from Racer to maintain goodwill. Also, Floataway Tours wishes to have a total capacity of at least 200. Data about the boats is summarized below:

Boat Speedhawk is manufactured by Sleekboat and its cost is 6000 It provides 3 seats and lead to profit 70 .

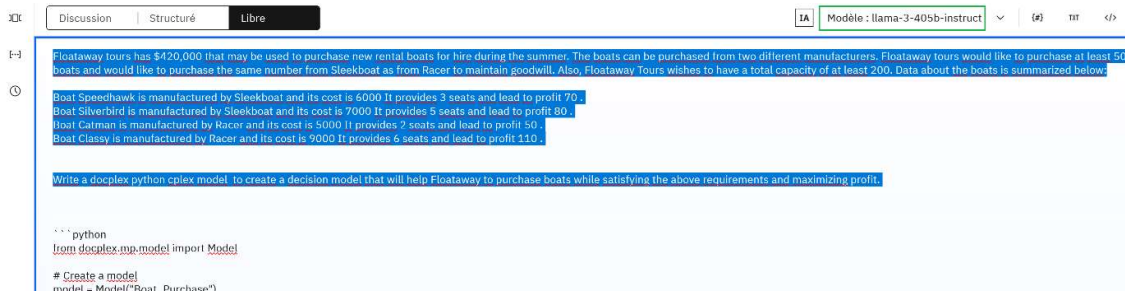
Boat Silverbird is manufactured by Sleekboat and its cost is 7000 It provides 5 seats and lead to profit 80 .

Boat Catman is manufactured by Racer and its cost is 5000 It provides 2 seats and lead to profit 50 .

Boat Classy is manufactured by Racer and its cost is 9000 It provides 6 seats and lead to profit 110 .

Write a docplex python cplex model to create a decision model that will help Floataway to purchase boats while satisfying the above requirements and maximizing profit.

Which can be seen in the watsonx.AI environment as



The screenshot shows a web interface with a top navigation bar containing 'Discussion', 'Structuré', and 'Libre'. On the right, there is a dropdown menu showing 'Modèle : llama-3-405b-instruct'. The main content area contains a problem statement in French about boat purchases for 'Floataway tours'. Below the problem statement, there is a prompt: 'Write a docplex python cplex model to create a decision model that will help Floataway to purchase boats while satisfying the above requirements and maximizing profit.' At the bottom, a snippet of the generated Python code is visible, starting with 'python' and 'from docplex.mp.model import Model'.

```
python
from docplex.mp.model import Model

# Create a model
model = Model("Boat_Purchase")
```

And the generated python model is

```
from docplex.mp.model import Model
```

```
# Create a model
```

```
model = Model("Boat_Purchase")
```

```
# Define the variables
```

```
Speedhawk = model.integer_var(name="Speedhawk", lb=0)
```

```
Silverbird = model.integer_var(name="Silverbird", lb=0)
```

```
Catman = model.integer_var(name="Catman", lb=0)
```

```
Classy = model.integer_var(name="Classy", lb=0)
```

```
# Define the objective function
```

```
model.maximize(70 * Speedhawk + 80 * Silverbird + 50 * Catman + 110 * Classy)
```

```
# Define the constraints
```

```
model.add_constraint(6000 * Speedhawk + 7000 * Silverbird + 5000 * Catman +  
9000 * Classy <= 420000, "Budget")
```

```
model.add_constraint(Speedhawk + Silverbird + Catman + Classy >= 50,  
"Minimum_Boats")
```

```
model.add_constraint(Speedhawk + Silverbird == Catman + Classy,  
"Equal_From_Each_Manufacturer")
```

```
model.add_constraint(3 * Speedhawk + 5 * Silverbird + 2 * Catman + 6 * Classy >=  
200, "Minimum_Capacity")
```

```
# Solve the model
```

```
solution = model.solve()
```

```
# Print the solution
```

```
print("Solution:")
```

```
print("Speedhawk:", solution.get_value(Speedhawk))
```

```
print("Silverbird:", solution.get_value(Silverbird))
```

```
print("Catman:", solution.get_value(Catman))
```

```
print("Classy:", solution.get_value(Classy))
```

which gives

Number of Speedhawk boats to purchase: 28.0

Number of Silverbird boats to purchase: 0

Number of Catman boats to purchase: 0

Number of Classy boats to purchase: 28.0

And a generated documentation

This model defines the variables as the number of each type of boat to purchase.

The objective function is to maximize the total profit. The constraints are:

- * The total cost of the boats cannot exceed the budget of \$420,000.
 - * The total number of boats purchased must be at least 50.
 - * The number of boats purchased from Sleekboat must be equal to the number of boats purchased from Racer.
 - * The total capacity of the boats must be at least 200.
- The model is then solved using the `solve` method, and the solution is printed to the console.