



**Universidade do Minho**  
Escola de Engenharia

MESTRADO EM ENGENHARIA E GESTÃO DE SISTEMAS DE INFORMAÇÃO  
2022/2023

## **Técnicas de Inteligência Artificial na Previsão e Otimização em Sistemas Empresariais**

---

### **Relatório Final**

---

Nuno Moreira – PG50670

Paulo Barros – PG50678

Duarte Brandão – PG50348

**Grupo TP1-G12**

Professor Paulo Cortez



# Tabela de Conteúdos

<b>Tabela de Conteúdos.....</b>	<b>2</b>
<b>1. Introdução.....</b>	<b>3</b>
<b>2. Execução do Projeto.....</b>	<b>3</b>
2.1. Constituição do Grupo.....	3
2.2. Planeamento do Trabalho.....	3
2.3. Tarefas Realizadas.....	4
2.3.1. Duarte Brandão.....	4
2.3.2. Nuno Moreira.....	5
2.3.3. Paulo Barros.....	6
2.4. Autoavaliação.....	7
<b>3. Objetivo de Previsão.....</b>	<b>7</b>
Modelos Forecasting.....	7
Modelos Machine Learning.....	7
Tratamento dos Dados.....	8
Modelo Weekly Naive.....	8
Configuração dos Modelos e Métodos de Validação.....	9
Resultados.....	10
Time Ordered Holdout Split.....	10
Modelos Univariados Multi-Step Ahead.....	10
Modelos Multivariados.....	12
Rolling Window e Growing Window.....	12
Conclusão.....	16
<b>4. Objetivo de Otimização.....</b>	<b>16</b>
4.1. Função de avaliação (eval).....	17
4.2. Função de reparação de soluções inválidas (repair).....	19
4.3. Função de cálculo do limite superior (getUpperLimit).....	20
4.4. Método Monte Carlo.....	20
4.5. Método Simulated Annealing.....	22
4.6. Método Hill Climbing.....	23
<b>5. Demonstração do Sistema Desenvolvido.....</b>	<b>24</b>
<b>6. Conclusões.....</b>	<b>25</b>



# 1. Introdução

Nos últimos anos, o campo da inteligência artificial tem desempenhado um papel cada vez mais importante na análise de dados e previsão de tendências. Com o advento de técnicas de machine learning, tornou-se possível extrair informações valiosas de conjuntos de dados complexos e utilizá-las para tomar decisões informadas em várias áreas.

Este projeto foi dividido em duas partes, na primeira focamos-nos em prever as vendas das duas bebidas para a última semana, e a testar esses valores com métricas de avaliação.

Na segunda parte deste trabalho, focamos na otimização dos valores previstos utilizando diversos métodos e desenvolvemos uma implementação que integra as duas partes anteriores. Utilizamos técnicas de otimização para ajustar os parâmetros dos modelos de previsão e melhorar o lucro.

## 2. Execução do Projeto

### 2.1. Constituição do Grupo

- Duarte Brandão PG50348
- Nuno Moreira PG50670
- Paulo Barros PG50678

### 2.2. Planeamento do Trabalho

Inicialmente o grupo criou um repositório no GitHub de forma a podermos partilhar e sincronizar o trabalho entre os membros do grupo e mantermos sempre os nossos ficheiros atualizados em qualquer computador.

Seguidamente, à medida que íamos avançando no projeto e tendo em atenção as problemáticas discutidas por todos em reunião, cada elemento ficava semanalmente encarregue de uma ou mais tarefas.



---

## 2.3. Tarefas Realizadas

Nos subtópicos é detalhado, por cada elemento do grupo, o trabalho realizado pelo mesmo no decorrer do semestre.

### 2.3.1. Duarte Brandão

Neste projeto, começando pela parte da previsão, implementei os modelos univariados de forecasting Holt Winter, Arima, Ets e Nnetar e de machine learning Ctree, Cv.glmnet, Decision Tree e Knn. Relativamente aos modelos multivariados testei os modelos Arima, Var e MLP para 3 cenários diferentes. Num cenário utilizei as colunas precipitação e temperatura, noutro as duas bebidas, e para último através da criação de uma nova coluna “feriados” testei um cenário com essa coluna e uma coluna da bebida diferente da que queria prever resultados.

Sobre a segunda parte do projeto, na otimização ajudei o grupo a construir os métodos de otimização para maximizar o lucro, Monte Carlo, Hill Climbing e Sann, e ainda testei os restantes modelos que foram fornecidos pelo professor mas apenas aqueles 3 anteriormente referidos deram resultados válidos. Para o multiobjetivo, o objetivo 2, tentei aplicar o método rgba mas os resultados não eram bons e tinha erros de código e por falta de tempo não consegui obter resultados para o multiobjetivo.

Quanto à interface dei apoio ao Nuno na ligação dos resultados da previsão e dos modelos de otimização para a comunicação entre resultados.

Por fim, no relatório fiquei encarregado pela Introdução, Desenvolvimento do tema, Objetivos de Otimização e Conclusão.



---

### 2.3.2. Nuno Moreira

Inicialmente, e como representante do grupo, elaborei o documento do semanário conforme os guias presentes no enunciado. Para isso, criei também a pasta partilhada no Google Drive.

Na parte da previsão, implementei algumas receitas do professor de forma muito básica, relativas a modelos de ML como `ksvm`, `lssvm`, `mlp`, `mlpe` e `randomforest`. Também fiz alguma pesquisa sobre atributos a adicionar, tendo mesmo adicionado uma coluna com informação sobre feriados. Para usar esta coluna, tentei codificar a previsão multivariada, sem sucesso.

Na parte da otimização, codifiquei os modelos *Hill Climbing*, *Monte Carlo* e *Simulated Annealing*, através das receitas do professor. Devido à simplicidade desta implementação, cheguei aos mesmos resultados que os meus colegas que também implementaram os mesmos modelos. Decidi não tentar implementar mais modelos de otimização, em prol do desenvolvimento da interface

Finalmente, o momento em que estive mais atarefado foi na codificação da interface. Inicialmente, estive a “brincar” com o código fornecido na secção “Get Started” do site do Shiny para me familiarizar com a tecnologia. Depois, codifiquei uma simples interface que permitia ao utilizador escolher uma semana e calculava um planeamento dessa semana, com um plano definido também pelo utilizador.

De seguida, e com a integração do código dos modelos de previsão e otimização, comecei a desenvolver a solução final. Esta teve várias etapas:

- No início, apenas com 1 modelo de previsão e 1 de otimização, a interface usava *verbatimTextOutput* para mostrar exatamente o output pré-definido dos modelos para a consola:
- De seguida, codifiquei junto com os meus colegas funções para executar os diferentes modelos a partir da seleção do utilizador. Isto permitiu-nos integrar os 5 melhores modelos de previsão e todos os modelos de otimização.
- Finalmente, organizei os modelos e os outputs dos mesmos para permitir, no “backend” do shiny, criar tabelas e gráficos.



---

### 2.3.3. Paulo Barros

Na primeira semana, foquei-me em ler o enunciado e entender as várias fases do projeto, os seus objetivos e requisitos, como também compreender o código fornecido pelo professor para a primeira parte do trabalho, após o respetivo estudo dos slides correspondentes. Esta tarefa implicou 2 horas de estudo.

Na segunda semana, encarreguei-me de implementar o modelo Holt Winter utilizando o método Time Ordered Holdout Split. Esta tarefa implicou 2 horas de trabalho.

Na terceira semana, fui responsável por implementar os modelos xgboost, cubist, lm, mr, mars, pcr, plsr, cpps e rvm para o método de validação Time Ordered Holdout Split. Esta tarefa implicou 2 horas de trabalho.

Na quarta, quinta, sexta semana, implementei o modelo univariado 1-step ahead elman (que posteriormente foi descartado por não corresponder ao pretendido). Também foquei-me em implementar todos os modelos implementados pelo grupo, mas utilizando desta vez os métodos Growing e Rolling Window, com os respetivos gráficos comparando os valores previstos com o target e obtendo o valor da métrica NMAE. Além disso, preocupei-me por unificar e organizar o trabalho realizado até o momento, a fim de ter um ambiente de trabalho mais adequado. Nestas três semanas dediquei quase 14 horas a estas tarefas.

Na sétima semana, analisei os resultados da métrica NMAE em todos os modelos nos vários métodos de validação com o objetivo de concluir qual era o melhor modelo. Após a análise criei duas funções com esse modelo, uma para a previsão das vendas da bebida Stella e outra para as da bebida Bud, para serem utilizadas pela função eval(). Também implementei o modelo Weekly Naive, para comparar a métrica NMAE deste com as dos outros modelos e ver quão melhor são. Estas tarefas implicaram 2 horas de trabalho ao todo.

Na semana oito, tentei implementar, na segunda parte do projeto, os modelos de otimização Hill Climbing, Nelder-Mead, BFGS, CG, L-BFGS-B, SANN. Apesar do parâmetro upper estar parcialmente mal consegui configurar corretamente o resto. Esta tarefa implicou 4 horas de trabalho.

Na última semana, unicamente foquei no relatório, dedicando-lhe 3 horas.



## 2.4. Autoavaliação

Após discussão entre o grupo e tendo em atenção os obstáculos e a forma como a equipa de trabalho lidou com os mesmos achamos que a nota merecida são 18 valores.

Dentro da equipa trabalhamos de forma igual e a divisão do trabalho foi justa entre todos pelo que não pedimos diferenciação de notas entre os elementos do grupo.

## 3. Objetivo de Previsão

Nesta primeira fase do projeto, começamos por selecionar os modelos de forecasting, os de machine-learning, as métricas e os métodos de avaliação que consideramos mais adequados para avaliar a qualidade dos modelos.

Os métodos de validação utilizados são o "Time Ordered Holdout Split", o "Rolling Window" e o "Growing Window".

Quanto aos modelos, decidimos por implementar o maior número possível, tanto sejam de forecasting como de machine learning, alguns deles sugeridos pelo docente da cadeira e outros encontrados na nossa pesquisa por mais.

Os modelos estudados são os seguintes:

### Modelos Forecasting

- Holt Winter
- Arima
- Ets
- Nnetar

### Modelos Machine Learning

- Naive Bayes
- Ksvm
- Mlp
- Neural Network (mlpe)
- Ctree
- Cv.glmnet
- Dt
- Knn
- Xgboost
- Cubist
- Lm
- Mr
- Mars
- Pcr
- Plsr
- Cppls
- Rvm

Relativamente às métricas, apesar de termos utilizados MAE (Mean Absolute Error), RMSE (Root Mean Square Error) e RRSE (Root Relative Squared Error) na avaliação de vários



modelos, consideramos mais adequado guiar-nos unicamente pela métrica NMAE (Normalized Mean Absolute Error).

## Tratamento dos Dados

Perante a grande quantidade de zeros que se encontravam nas últimas duas semanas na coluna da bebida Stella optamos por eliminá-los do dataset.

Quanto ao resto dos dados, consideramos que não havia necessidade de os alterar.

Na fase dos modelos multivariados intentamos obter melhores valores de previsão da última semana através de acréscimo de dados que consideramos pertinentes (os feriados e os dias com jogos de futebol), porém os resultados finais eram piores. Logo, optamos por não os utilizar no cenário final.

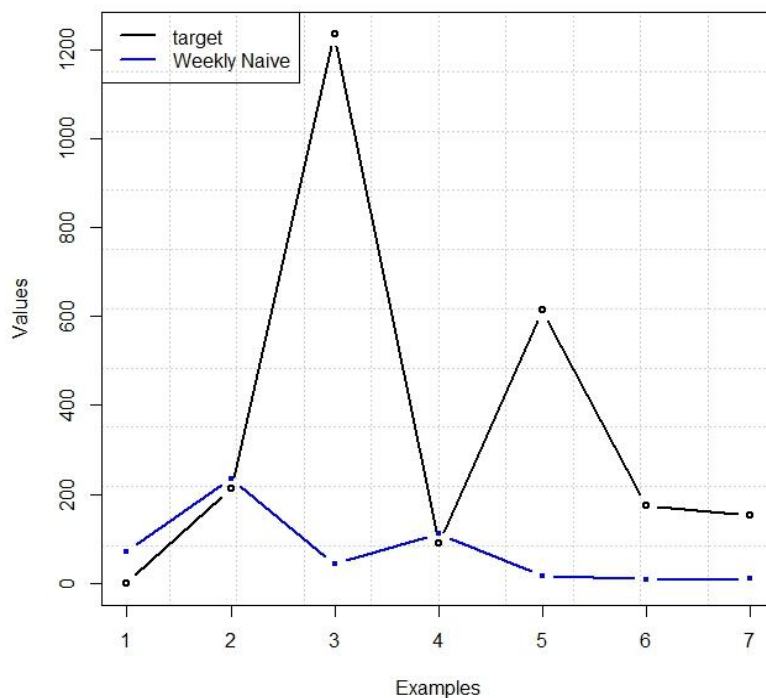
## Modelo Weekly Naive

Com o objetivo de avaliar a “qualidade” das previsões obtidas elaboramos o modelo Weekly Naive, uma vez que é uma boa forma de avaliar quão melhor são os resultados previstos pelos outros modelos comparando os resultados.

Utilizando o modelo Weekly Naive, o erro obtido pela métrica NMAE para os dados da bebida Stella é de um 33.64%, enquanto que a bebida Bud obteve um erro de 25.86%.

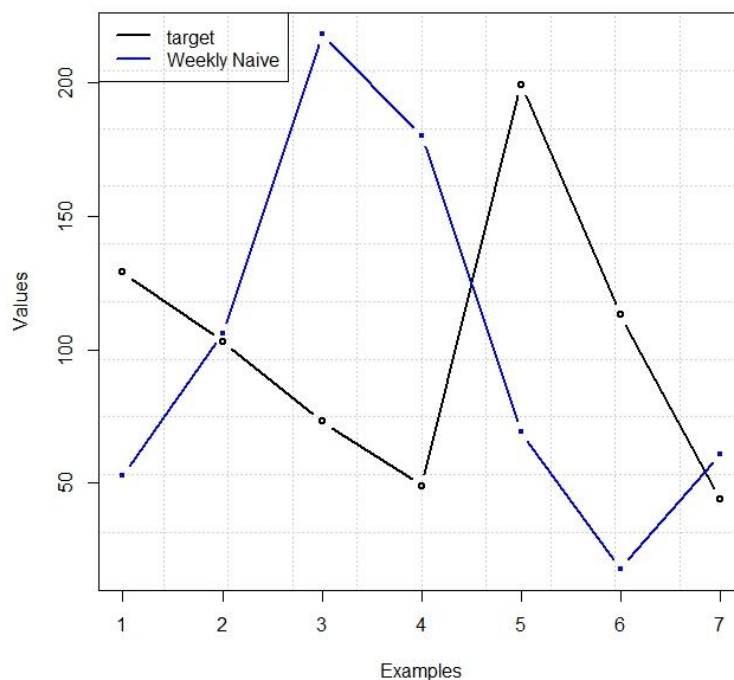
Os dois valores são significativamente elevados, confirmando a pouca fiabilidade do modelo.

A seguinte figura representa um gráfico de abscissas, para a bebida Stella, dos resultados previstos pelo modelo e a target:



A seguinte figura representa um gráfico de abscissas para a bebida Bud dos resultados previsto pelo modelo e a target:





## Configuração dos Modelos e Métodos de Validação

Para a implementação dos modelos, começamos por nos preocupar pela sua configuração e realizando pequenos testes, que consistiam em treinar com todos os dados excepto os da última semana e tentando fazer previsões para essa semana.

Com o objetivo de verificar que cada modelo de previsão foi bem construído e se fazia sentido, utilizamos gráficos de valores previstos versus desejados, a métrica NMAE e comparando os valores dados por esta em cada modelo com a do método Weekly Naive.

Após uma exaustiva fase de testing concluímos que a melhor configuração para os modelo de previsão utilizando o método de validação Time Ordered Holdout Split é a seguinte:

- A variável K tem que ser igual a 7, uma vez que a sazonalidade neste projeto é semanal quando muito;
- Para utilizar a versão sazonal nos modelos do pacote forecast atribuímos o valor da variável K, ou seja 7, ao parâmetro “frequency” na criação do objeto ts;
- Para beneficiar os modelos do pacote rminer de informação sazonal introduzimos na função CaseSeries o timelags c(1:7);

Enquanto à configuração dos modelos de previsão utilizando os métodos de Rolling e Growing Window estes foram mantidos, unicamente havendo configuração dos métodos em si, que são os seguintes:

- O valor da variável Runs é 20, que equivale a quantidade de iterações dos métodos Growing e Rolling Window;
- As diferentes medidas calculadas por cada iteração são agrupadas pela mediana;



## Resultados

### Time Ordered Holdout Split

Modelos Univariados Multi-Step Ahead

Tabela de Resultados para a bebida Bud:

Bebida	Modelo	Tipo de Modelo de Previsão	NMAE (em %)
Stella	Weekly Naive		33.64
	Naive	Forecast	3.11
	Arima	Machine-Learning	3.40
	Nnetar	Machine-Learning	7.72
	Ets	Machine-Learning	4.59
	HolWinters	Machine-Learning	3.60
	Ksvm	Forecast	3.67
	Mlp	Forecast	5.38
	NN	Forecast	5.92
	RandomForest	Forecast	4.20
	Ctree	Forecast	4.89
	Cv.glmnet	Forecast	3.09
	DecisionTree	Forecast	5.84
	Knn	Forecast	3.72
	Xgboost	Forecast	4.50
	Cubist	Forecast	3.44
	Lm	Forecast	3.48
	Mr	Forecast	3.48
	Mars	Forecast	3.33
	Pcr	Forecast	3.48
	Plsr	Forecast	3.48
	Cppls	Forecast	3.48
	Rvm	Forecast	3.89



Após a obtenção dos resultados de modelos univariados multi-step ahead utilizando o método de validação “Time Ordered Holdout Split” para a bebida Stella concluímos que o melhor modelo segundo a métrica NMAE é o modelo Cv.glmnet com um erro de 3.09%, 10.89 vezes melhor em comparação ao erro da previsão do modelo Weekly Naive.

Tabela de Resultados para a bebida Bud:

Bebida	Modelo	Tipo de Modelo de Previsão	NMAE
Bud	Weekly Naive		25.86
	Naive	Forecast	22.31
	Arima	Machine-Learning	22.94
	Nnetar	Machine-Learning	31.94
	Ets	Machine-Learning	21.91
	HolWinters	Machine-Learning	23.10
	Ksvm	Forecast	24.18
	Mlp	Forecast	20.49
	NN	Forecast	22.46
	RandomForest	Forecast	22.95
	Ctree	Forecast	20.61
	Cv.glmnet	Forecast	22.34
	DecisionTree	Forecast	24.36
	Knn	Forecast	22.45
	Xgboost	Forecast	26.46
	Cubist	Forecast	23.94
	Lm	Forecast	22.22
	Mr	Forecast	22.22
	Mars	Forecast	21.69
	Pcr	Forecast	22.22
	Plsr	Forecast	22.22
	Cppls	Forecast	22.22
	Rvm	Forecast	22.33



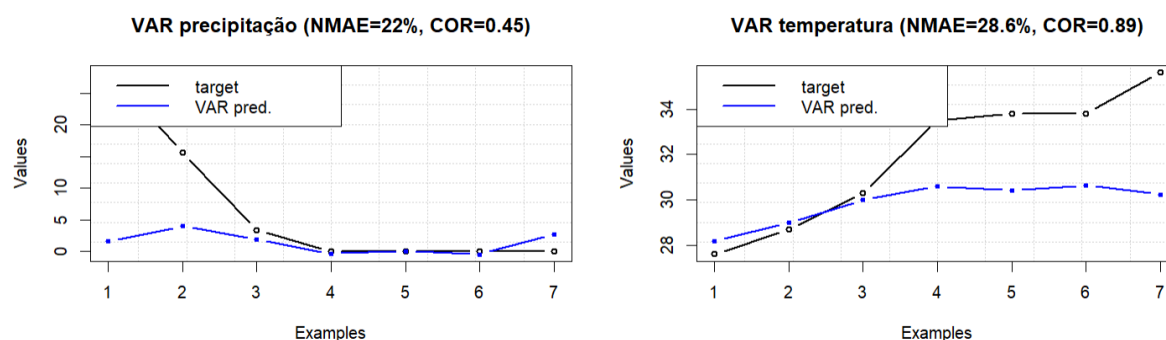
Após a obtenção dos resultados de modelos univariados multi-step ahead utilizando o método de validação “Time Ordered Holdout Split” para a bebida Bud concluímos que o melhor modelo segundo a métrica NMAE é o modelo Mlp com um erro de 20.49%, 1.26 vezes melhor em comparação ao erro da previsão do modelo Weekly Naive.

## Modelos Multivariados

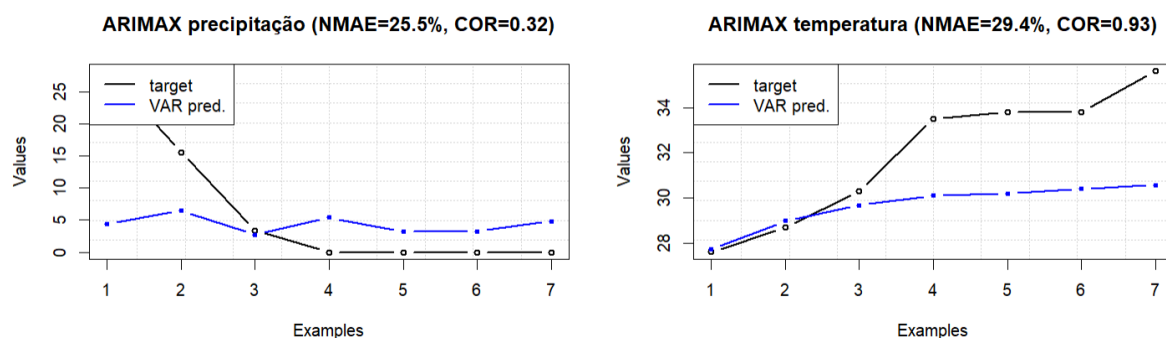
Nos modelos multivariados implementamos os modelos VAR, ARIMAX e MLP, mas este último modelo (MLP) não deu resultados tão bons e por isso decidimos apenas aplicar no relatório os outros dois. Para prever os valores, utilizamos as colunas da precipitação e temperatura para prever os valores das 2 bebidas.

Nos seguintes gráficos são mostrados os valores para a bebida Stella sendo que para a bud os resultados são idênticos:

### Var:



### Arimax:





## Rolling Window e Growing Window

Tabela de Resultados para a bebida Stella:

Bebida	Modelo	Método de Validação	NMAE
Stella	Weekly Naive		33.64
	Naive	Growing Window	5.05
		Rolling Window	5.05
	Arima	Growing Window	3.84
		Rolling Window	4.27
	Nnetar	Growing Window	7.05
		Rolling Window	9.43
	Ets	Growing Window	2.78
		Rolling Window	2.98
	HolWinters	Growing Window	3.23
		Rolling Window	2.91
	Ksvm	Growing Window	1.25
		Rolling Window	1.21
	Mlp	Growing Window	3.08
		Rolling Window	2.86
	NN	Growing Window	3.02
		Rolling Window	2.90
	RandomForest	Growing Window	4.57
		Rolling Window	4.10
	Ctree	Growing Window	4.40
		Rolling Window	4.40
	Cv.glmnet	Growing Window	4.89
		Rolling Window	4.84
	DecisionTree	Growing Window	5.33



	Knn	Rolling Window	5.33
		Growing Window	2.90
	Xgboost	Rolling Window	2.90
		Growing Window	1.95
	Cubist	Rolling Window	1.95
		Growing Window	1.69
	Lm	Rolling Window	1.69
		Growing Window	2.56
	Mr	Rolling Window	2.55
		Growing Window	2.55
	Mars	Rolling Window	3.00
		Growing Window	3.0
	Pcr	Rolling Window	2.55
		Growing Window	2.55
	Plsr	Rolling Window	2.55
		Growing Window	2.55
	Cppls	Rolling Window	2.55
		Growing Window	2.55
	Rvm	Rolling Window	3.19
		Growing Window	3.00

Após a obtenção dos resultados de modelos univariados multi-step ahead utilizando os métodos de validações “Rolling Window” e “Growing Window” para a bebida Stella concluímos que o melhor modelo segundo a métrica NMAE para o primeiro método é o modelo Ksvm com um erro de 1.21%, 27.80 vezes melhor em comparação ao erro da previsão do modelo Weekly Naive. Quanto ao segundo método, o melhor modelos segundo a métrica NMAE também é o modelo Ksvm com um erro de 1.25%, 26.91 vezes melhor em comparação ao erro da previsão do modelo Weekly Naive.



Tabela de Resultados para a bebida Bud:

Bebida	Modelo	Método de Validação	NMAE
Bud	Weekly Naive		25.86
	Naive	Growing Window	5.55
		Rolling Window	5.55
	Arima	Growing Window	8.10
		Rolling Window	7.93
	Nnetar	Growing Window	11.07
		Rolling Window	12.22
	Ets	Growing Window	11.41
		Rolling Window	11.91
	HolWinters	Growing Window	11.45
		Rolling Window	11.48
	Ksvm	Growing Window	3.04
		Rolling Window	3.04
	Mlp	Growing Window	4.20
		Rolling Window	3.98
	NN	Growing Window	3.94
		Rolling Window	4.45
	RandomForest	Growing Window	4.48
		Rolling Window	4.48
	Ctree	Growing Window	5.34
		Rolling Window	5.34
	Cv.glmnet	Growing Window	5.57
		Rolling Window	5.58
	DecisionTree	Growing Window	6.33
		Rolling Window	6.33
	Knn	Growing Window	4.75



		Rolling Window	4.75
	Xgboost	Growing Window	4.14
		Rolling Window	4.14
	Cubist	Growing Window	3.97
		Rolling Window	3.97
	Lm	Growing Window	5.37
		Rolling Window	5.37
	Mr	Growing Window	5.37
		Rolling Window	5.37
	Mars	Growing Window	5.00
		Rolling Window	5.00
	Pcr	Growing Window	5.37
		Rolling Window	5.37
	Plsr	Growing Window	5.37
		Rolling Window	5.37
	Cppls	Growing Window	5.37
		Rolling Window	5.37
	Rvm	Growing Window	4.30
		Rolling Window	4.84

Após a obtenção dos resultados de modelos univariados multi-step ahead utilizando os métodos de validações “Rolling Window” e “Growing Window” para a bebida Bud concluímos que o melhor modelo segundo a métrica NMAE para o primeiro método é o modelo Ksvm com um erro de 3.04%, 8.51 vezes melhor em comparação ao erro da previsão do modelo Weekly Naive. Quanto ao segundo método, o melhor modelo segundo a métrica NMAE também é o modelo Ksvm novamente com um erro de 3.04%, 8.51 vezes melhor em comparação ao erro da previsão do modelo Weekly Naive.

## Conclusão

Após a análise de todos os cenários possíveis concluímos que a melhor abordagem para prever a venda de ambas bebidas para a última semana é a utilização do modelo ksvm com método de validação rolling window, sendo que na bebida Bud não há diferença entre usar esse método ou o growing window, uma vez que devolve resultados idêntico segundo a métrica NMAE.





## 4. Objetivo de Otimização

Para esta segunda fase, o grupo começou por codificar as funções necessárias que suportam o objetivo de otimização. Estas encontram-se no ficheiro zip entregue em anexo a este documento, e são explicadas nos subtópicos a seguir.

### 4.1. Função de avaliação (eval)

Esta função foi codificada à imagem do slide 15 do enunciado do projeto e, por isso, resulta num output idêntico a este. Tal é comprovado na seguinte captura de ecrã, que recria os dados do slide referido.

```
> source("~/Documents/GitHub/TIAPPOSE/Nuno/otim.R")
Valores estimados para venda:
stella: 141 154 18 102 211 69 37 0
bud: 211 172 220 330 39 45 125 0
arm.: 6 3 0 1 1 0 1 > custo: 135
v1: 2 0 0 1 0 0 0
v2: 2 1 0 0 1 0 0
v3: 2 1 0 0 0 0 0
custo veiculos: 489
bebidas empacotadas e distribuidas:
stella: 160 8 0 52 20 0 0
bud: 200 200 0 0 30 0 0
- vendas stella: 141 27 0 52 20 0 0 > lucro: 1368
- vendas bud: 200 172 28 0 30 0 0 > lucro: 1892
- stock stella: 19 0 0 0 0 0 0 > custo: 19
- stock bud: 0 28 0 0 0 0 0 > custo: 28
Lucro Final: R$ 2589
Recursos: 22
```

A função eval percorre os 7 dias das previsões e calcula as vendas reais (tendo em conta as previsões e as bebidas disponíveis) e os custos cumulativos. Finalmente, retorna o lucro final.



```
# Loop through each day
for (i in 1:7) {

  # cerveja disponivel STELLA
  disponivel1 <- preparadas1[i] + stock1[i]

  # cerveja disponivel BUD
  disponivel2 <- preparadas2[i] + stock2[i]

  # STELLA -----
  # Calculate actual sales
  if (disponivel1 <= sales_pred1[i]) {
    actual_sales1[i] <- disponivel1
  } else {
    actual_sales1[i] <- sales_pred1[i]
    stock1[i+1] <- disponivel1 - sales_pred1[i]
  }
  # Calculate profit
  profit1 <- profit1 + (actual_sales1[i] * ganho1)
  # -----

  # BUD -----
  # Calculate actual sales
  if (disponivel2 <= sales_pred2[i]) {
    actual_sales2[i] <- disponivel2
  } else {
    actual_sales2[i] <- sales_pred2[i]
    stock2[i+1] <- disponivel2 - sales_pred2[i]
  }
  # Calculate profit
  profit2 <- profit2 + (actual_sales2[i] * ganho2)
  # -----

  # Custos cumulativos - fim de semana hardcoded
  if (i == 2 || i == 3) {
    soma_arm <- soma_arm + arm[i] * (custo_arm+5)
    soma_v1 <- soma_v1 + v1[i] * (custo_v1+5)
    soma_v2 <- soma_v2 + v2[i] * (custo_v2+5)
    soma_v3 <- soma_v3 + v3[i] * (custo_v3+5)
  } else {
    soma_arm <- soma_arm + arm[i] * custo_arm
    soma_v1 <- soma_v1 + v1[i] * custo_v1
    soma_v2 <- soma_v2 + v2[i] * custo_v2
    soma_v3 <- soma_v3 + v3[i] * custo_v3
  }
}

total_profit <- profit1 + profit2 - soma_arm - soma_v1 - soma_v2 - soma_v3 - sum(stock1) - sum(stock2)

resources <- sum(arm) + sum(v1) + sum(v2) + sum(v3)

return(total_profit)
```

Para além da função eval, foi codificada a função printeval, que efetua exatamente os mesmos cálculos mas retorna o plano detalhado na imagem no topo deste subtópico.



```
total_profit <- profit1 + profit2 - soma_arm - soma_v1 - soma_v2 - soma_v3 - sum(stock1) - sum(stock2)

resources <- sum(arm) + sum(v1) + sum(v2) + sum(v3)

# resposta:
output <- character()
output <- c(output,cat("Valores estimados para venda: \n"))
output <- c(output,cat("  stella: ",sales_pred1,"\n"))
output <- c(output,cat("    bud: ",sales_pred2,"\n"))
output <- c(output,cat("  arm.: ",arm," > custo: ",soma_arm))
output <- c(output,cat("\n    v1: ",v1))
output <- c(output,cat("\n    v2: ",v2))
output <- c(output,cat("\n    v3: ",v3))
output <- c(output,cat("\ncusto veiculos: ",soma_v1+soma_v2+soma_v3,"\n"))
output <- c(output,cat("bebidas empacotadas e distribuidas: \n"))
output <- c(output,cat("  stella: ",preparadas1,"\n"))
output <- c(output,cat("    bud: ",preparadas2,"\n"))
output <- c(output,cat("- vendas stella:",head(actual_sales1,-1)," > lucro: ",profit1))
output <- c(output,cat("\n- vendas bud:",head(actual_sales2,-1)," > lucro: ",profit2))
output <- c(output,cat("\n- stock stella:",tail(stock1,-1)," > custo: ",sum(stock1)))
output <- c(output,cat("\n- stock bud:",tail(stock2,-1)," > custo: ",sum(stock2),"\n"))
output <- c(output,cat("Lucro Final: R$", total_profit, "\n"))
output <- c(output,cat("Recursos: ", resources, "\n"))
output_final <- paste(output,"\n")

return(output_final)
```

## 4.2. Função de reparação de soluções inválidas (repair)

Esta função é chamada pela função eval e, quando são encontrados no plano valores de bebidas preparadas ilegais para os recursos disponíveis, resolve essa incoerência. Mais especificamente, calcula o valor máximo de bebidas possíveis de serem preparadas e divide-as pelas duas marcas. Em modo de exemplo, alteramos os recursos de armazenamento do exemplo exposto acima e apresentamos a seguir os resultados:

```
> printeval(s)
AVISO: Plano com valor ilegal encontrado nos recursos de armazém, no dia 1 - A resolver...
AVISO: Plano com valor ilegal encontrado nos recursos de armazém, no dia 2 - A resolver...
Valores estimados para venda:
  stella: 141 154 18 102 211 69 37 0
    bud: 211 172 220 330 39 45 125 0
  arm.: 1 1 0 1 1 0 1 > custo: 55
    v1: 2 0 0 1 0 0 0
    v2: 2 1 0 0 1 0 0
    v3: 2 1 0 0 0 0 0
custo veiculos: 489
bebidas empacotadas e distribuidas:
  stella: 36 8 0 52 20 0 0
    bud: 36 36 0 0 30 0 0
- vendas stella: 36 8 0 52 20 0 0 > lucro: 661.2
- vendas bud: 36 36 0 0 30 0 0 > lucro: 448.8
- stock stella: 0 0 0 0 0 0 0 > custo: 0
- stock bud: 0 0 0 0 0 0 0 > custo: 0
Lucro Final: R$ 566
Recursos: 15
```

A função repair foi implementada da seguinte forma:

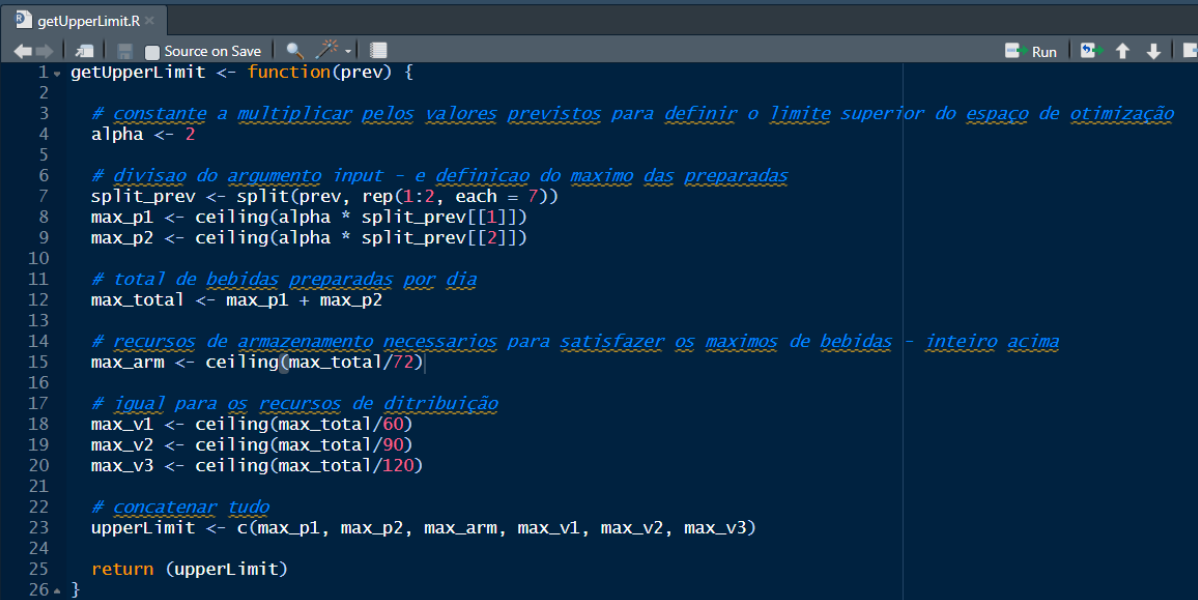


```
getUpperLimit <- function(prev) {  
  
  # constante a multiplicar pelos valores previstos para definir o limite superior do espaço de otimização  
  alpha <- 2  
  
  # divisao do argumento input - e definicao do maximo das preparadas  
  split_prev <- split(prev, rep(1:2, each = 7))  
  max_p1 <- ceiling(alpha * split_prev[[1]])  
  max_p2 <- ceiling(alpha * split_prev[[2]])  
  
  # total de bebidas preparadas por dia  
  max_total <- max_p1 + max_p2  
  
  # recursos de armazenamento necessarios para satisfazer os maximos de bebidas - inteiro acima  
  max_arm <- ceiling(max_total/72)  
  
  # igual para os recursos de ditribuição  
  max_v1 <- ceiling(max_total/60)  
  max_v2 <- ceiling(max_total/90)  
  max_v3 <- ceiling(max_total/120)  
  
  # concatenar tudo  
  upperLimit <- c(max_p1, max_p2, max_arm, max_v1, max_v2, max_v3)  
  
  return (upperLimit)  
}
```

### 4.3. Função de cálculo do limite superior (getUpperLimit)

Esta função é chamada pelos métodos de otimização para calcular os limites superiores das previsões das bebidas, recursos de armazenamento e recursos de distribuição.

Estes limites foram estabelecidos da seguinte forma:



```
getUpperLimit.R  
1 getUpperLimit <- function(prev) {  
2  
3   # constante a multiplicar pelos valores previstos para definir o limite superior do espaço de otimização  
4   alpha <- 2  
5  
6   # divisao do argumento input - e definicao do maximo das preparadas  
7   split_prev <- split(prev, rep(1:2, each = 7))  
8   max_p1 <- ceiling(alpha * split_prev[[1]])  
9   max_p2 <- ceiling(alpha * split_prev[[2]])  
10  
11   # total de bebidas preparadas por dia  
12   max_total <- max_p1 + max_p2  
13  
14   # recursos de armazenamento necessarios para satisfazer os maximos de bebidas - inteiro acima  
15   max_arm <- ceiling(max_total/72)  
16  
17   # igual para os recursos de ditribuição  
18   max_v1 <- ceiling(max_total/60)  
19   max_v2 <- ceiling(max_total/90)  
20   max_v3 <- ceiling(max_total/120)  
21  
22   # concatenar tudo  
23   upperLimit <- c(max_p1, max_p2, max_arm, max_v1, max_v2, max_v3)  
24  
25   return (upperLimit)  
26 }  
27
```

### 4.4. Método Monte Carlo

Para o método Monte Carlo, foi necessário definir os limites inferiores e superiores, lower e upper respetivamente, e o número de interações que neste caso foi estabelecido com o valor 1000.



```
otimizar(monte_carlo).R
1 source("blind.R") # fsearch is defined here
2 source("montecarlo.R") # mcsearch is defined here
3 source("teste-predict.R") #função predict stella
4 source("getUpperLimit.R")
5
6 # dimension
7 D=42
8
9 # evaluation function:
10 source("eval.R")
11
12 sales_pred1 <- predict_ksvm_stella()
13 sales_pred2 <- predict_ksvm_bud()
14
15 N=1000 # number of searches
16 # monte carlo search with D=2 and x in [-10.4,10.4]
17 lower=rep(0,D)
18 upper=getUpperLimit(c(sales_pred1,sales_pred2)) # upper bounds
19 MC=mcsearch(fn=eval,lower=lower,upper=upper,N=N,type="max")
20 cat("Best Solution:",MC$sol,"\n","Evaluation Function:",MC$eval,"(found at iteration:",MC$index,")\n")
21 |
```

21:1 (Top Level) R Script

```
Console Terminal Background Jobs
R 4.2.3 ~ /GitHub/novo/TIAPPOSE/projeto-final-tp1-g12/
> source("otimizar(monte_carlo).R")
[1] 2
[1] 89.35542 75.16464 48.92609 33.61919 21.59043 30.55042 47.64880
[1] 2
[1] 113.84670 111.76816 99.26259 61.77464 38.58846 41.02107 81.81809
Best Solution: 87.76519 87.05736 51.32509 28.539 9.962992 14.14106 26.28037 159.2417 79.2906 134.9344 87.23175 4.211904
16.11119 124.3047 5.767176 3.853385 4.949289 1.337896 0.7771534 0.3297644 3.822959 1.908698 5.012916 0.3468155 0.0862438
7 0.1751721 2.228719 3.81231 1.110055 1.0383 1.373835 0.4774218 0.518639 1.346348 0.4917318 0.4276778 1.274159 2.446752
1.805022 0.1952809 1.350016 1.330336
Evaluation Function: 2153.091 (found at iteration: 211 )
>
```



## 4.5. Método Simulated Annealing

Para o método Simulated Annealing (SANN) foi necessário alterar a função “CSANN”, adicionando o parâmetro “fnscale=-1” devido a problemas que tínhamos com resultados negativos. Para além disso definimos uma solução inicial “s0” que é aleatória e também definimos o lower, upper e o número de interações, tal como para o método anterior.

```
otimizar(sann).R
2 source("montecarlo.R") # mcsearch is defined here
3 source("teste-predict.R") #função predict stella
4 source("getUpperLimit.R")
5 source("eval.R")
6 source("hill.R")
7
8 N=1000 # searches
9 REPORT=N/10 # report results
10
11 sales_pred1 = predict_ksvm_stella()
12 sales_pred2 = predict_ksvm_bud()
13
14 max_previsto_stella = max(predict_ksvm_stella())
15 max_previsto_bud = max(predict_ksvm_bud())
16
17 D=42 # dimension
18 lower=rep(0,D) # lower bounds
19 upper=getUpperLimit(c(sales_pred1,sales_pred2)) # upper bounds
20
21 # slight change of a real par under a normal u(0,0.5) function:
22 rchange2=function(par) # change for hclimbing
23 { hchange(par, lower=lower, upper=upper, rnorm, mean=0, sd=0.5, round=TRUE) }
24
25 eval1=function(s) return(eval(s))
26 sann <- function(){
27
28   s0=c(rep(sample(0:max_previsto_stella +max_previsto_stella*0.5, 1),7),
29         rep(sample(0:max_previsto_bud + max_previsto_bud*0.5,1),7),
30         rep(sample(0:72, 1),7),rep(sample(0:60, 1),7),rep(sample(0:90, 1),7),rep(sample(0:120, 1),7))
31   cat("Simulated Annealing search D=",D,"(iters=",N,")\n")
32   CSANN=list(maxit=N,temp=2000,trace=TRUE, fnscale= -1)
33   SA=optim(par=s0,fn=eval1,method="SANN",gr=rchange2,control=CSANN)
34   #L=-eval1(SA$par)
35   #if(L>best) { BESTSA=SA; best=L;}
36   #s0 = s0[7+41:length(s0)]
37   #}
38   cat("Best Solution:",SA$par,"\n","Evaluation Function",SA$value,"\n")
39
40 }
41 sann()
```

```
> source("otimizar(sann).R")
[1] 2
[1] 83.41133 73.25847 47.02047 29.61442 21.98900 31.13186 47.54237
[1] 2
[1] 119.44120 113.62130 95.77157 61.67696 37.63202 35.66713 76.64413
[1] 2
[1] 87.02666 74.84847 48.78220 32.30581 21.90054 30.83524 47.66847
[1] 2
[1] 122.64493 119.01687 94.39503 61.55000 35.91662 33.78843 78.35052
Simulated Annealing search D= 42 (iters= 1000 )
sann objective function values
initial      value 31960.766544
iter        999 value -2272.702291
final        value -2272.702291
sann stopped after 999 iterations
Best Solution: 98.51333 121.5133 92 52 44 51 91 138.3225 148.3225 157.3225 121 63 56 146 5 6 4 1 2 0 2 5 2 0 0 0 2 0 0 1 1 2
2 0 1 2 1 0 1 0 0 2
Evaluation Function 2272.702
>
```



## 4.6. Método Hill Climbing

Para o método Hill Climbing apenas foi necessário definir a função uma solução inicial aleatória no espaço de valores, o lower, upper e o número de iterações como nos métodos analisados anteriormente.

```
otimizar(hill_climb).R x
Source on Save
Run Source
1 source("hill.R") # hclimbing is defined here
2 source("montecarlo.R") # mcsearch is defined here
3 source("teste-predict.R") #função predict stella
4 source("getUpperLimit.R")
5
6 # dimension
7 D=42
8
9 # evaluation function:
10 source("eval.R")
11
12 sales_pred1 <- predict_ksvm_stella()
13 sales_pred2 <- predict_ksvm_bud()
14
15 max_previsto_stella = max(predict_ksvm_stella())
16 max_previsto_bud = max(predict_ksvm_bud())
17
18
19 N=1000 # number of searches
20 REPORT=N/20 # report results
21
22 # monte carlo search with D=2 and x in [-10.4,10.4]
23 lower=rep(0,D)
24 upper=getUpperLimit(c(sales_pred1,sales_pred2)) # upper bounds
25
26 rchange1=function(par,lower,upper) # change for hclimbing
27 { hchange(par, lower=lower, upper=upper, rnorm, mean=0, sd=0.25, round=FALSE) }
28
29 s0=c(rep(sample(0:max_previsto_stella +max_previsto_stella*0.5, 1),7),
30      rep(sample(0:max_previsto_bud + max_previsto_bud*0.5,1),7),rep(sample(0:72, 1),7),
31      rep(sample(0:60, 1),7),rep(sample(0:90, 1),7),rep(sample(0:120, 1),7))
32
33 cat("hill climbing search D=",D,"(iters=",N,")\n")
34 # initial solution:
35 HC=hclimbing(par=s0,fn=eval,change=rchange1,lower=lower,upper=upper,type="max",
36             control=list(maxit=N,REPORT=REPORT,digits=2))
37
38 cat("Best Solution:",HC$sol,"\n","Evaluation Function",HC$eval,"\n")
39
40 57.65 45.56 64 85.98 87.01 87.92 87.33 90.26 75.55 71.75 90.01 3.1 2.64 2.64 2.67 0 1.16 2.88 1.11 3.71 0.74 0 0.94 0.15 0
41 1.08 0.3 0.53 1.04 0 0.24 0 0.12 0 0 0.2 0.03 0.99 1.44 f: 2650.85
42 i: 950 s: 85.14 83.38 86.61 56.74 44.89 64 86.05 86.84 87.22 87.9 90.25 74.8 71.89 90.36 2.81 2.99 2.62 2.41 0.27 0.83 3.15
43 1.03 3.57 0.56 0 0.53 0.34 0.07 1.39 0.45 0.38 1.31 0 0.53 0.53 0.24 0.01 0 0.61 0.22 0.36 1.12 f: 2743.67 s' 85.42 83.52 8
44 6.97 57.25 44.71 64 86.44 86.53 87.24 87.93 90.33 74.75 71.98 90.39 3.09 2.65 2.29 2.75 0.52 0.74 3.67 1.16 3.61 0.78 0.26
45 0.68 0.2 0 1.8 0.65 0.4 1.14 0.31 0.26 0.87 0.05 0.37 0.09 0.57 0.15 0.04 1.3 f: 2636.13
46 i: 1000 s: 85.37 83.42 86.6 56.76 44.47 64 85.31 87.19 87.63 87.32 89.62 74.87 72.03 90.22 2.26 3.4 1.84 2.49 0.09 1.04 3.6
47 1 1.1 3.48 0.76 0.23 0.03 0.3 0 1.21 0.04 0.42 1.89 0.01 0.68 0.91 0 0.12 0.39 0 0.3 0.51 0.54 f: 2767.91 s' 85.44 83.34 8
48 6.44 56.76 44.62 63.97 85.05 87.06 87.63 86.97 89.41 75.04 71.51 90.37 1.89 3.23 1.62 2.56 0 0.67 3.7 0.88 3.36 0.65 0 0
49 48 0.21 1.11 0.46 0.27 1.8 0 0.81 1.08 0.19 0.55 0.1 0 0.48 0.75 0.56 f: 2552.95
50 best: 85.37 83.42 86.6 56.76 44.47 64 85.31 87.19 87.63 87.32 89.62 74.87 72.03 90.22 2.26 3.4 1.84 2.49 0.09 1.04 3.61 1.1
51 3.48 0.76 0.23 0.03 0.3 0 1.21 0.04 0.42 1.89 0.01 0.68 0.91 0 0.12 0.39 0 0.3 0.51 0.54 f: 2767.91
52 Best Solution: 85.37302 83.41537 86.60367 56.7585 44.46774 64 85.30846 87.18807 87.62948 87.31801 89.6181 74.87163 72.03349
53 90.21868 2.258944 3.402 1.836537 2.486042 0.09251174 1.043471 3.608135 1.103992 3.4782 0.7629932 0.2341723 0.02506256 0.298
54 8851 0 1.205162 0.04085788 0.4205615 1.888938 0.01335219 0.6826424 0.913307 0 0.12148 0.3884131 0 0.3031275 0.5083943 0.539
55 1944
56 Evaluation Function 2767.907
```



## 5. Demonstração do Sistema Desenvolvido

Para demonstrar a solução final integrada desenvolvida pelo grupo, foi gravado um pequeno vídeo da utilização da mesma. Este vídeo encontra-se publicado e disponível no seguinte link:

- <https://youtu.be/BS7BGSpFRwc>





---

## 6. Conclusões

Em jeito de conclusão, o nosso projecto universitário centrou-se na previsão dos valores das vendas para duas bebidas e na sua optimização para obter melhores lucros. Através de uma investigação e análise rigorosa, utilizamos várias técnicas e metodologias para desenvolver um modelo robusto que pudesse prever com precisão as vendas e fornecer informações valiosas para a optimização dos lucros.

Ao longo do projecto, deparámo-nos com vários desafios, relativos à implementação dos modelos e interligar tudo por uma interface.. No entanto, através de muito estudo e pesquisa conseguimos ultrapassar estes obstáculos e desenvolver uma estrutura fiável para a previsão de vendas e optimização de lucros.

Em conclusão, o nosso projecto universitário abordou aspectos importantes da maneira que conseguimos perceber a importância da ciência de dados para otimizar os recursos das empresas.