

## **1 - Cenário Específico – Neurologia - Detecção Cefaleia:**

Na área Neurológica há uma defasagem do uso de Inteligência Artificial (IA) em relação a outras áreas da medicina que, apesar de ainda muito recentes e com potencial de crescimento, há uma emergência de novas ferramentas de auxílio à saúde. Tendo em vista esse atraso e a possibilidade de facilitar o estudo neurológico, o trabalho é desenvolvido para criar uma ferramenta que auxilie profissionais no diagnósticos de Cefaleia, popularmente conhecida como dor de cabeça que, além de muito comum entre a população, estando entre as quatro doenças neurológicas mais sofridas, é perigosa a ponto de ser um fator limitante para atividades comuns do indivíduo.

Segundo o Protocolo Nacional para Diagnóstico e Manejo das Cefaleias nas Unidades de Urgência do Brasil, de 2018, feito por pesquisadores da Academia Brasileira de Neurologia – Departamento Científico de Cefaleia Sociedade Brasileira de Cefaleia, entre o ano de 2017 e 2018, cerca de 70% das pessoas apresentaram o sintoma, sendo a permanência da doença nos homens de 94% e 99% nas mulheres. Já em um estudo exposto pela Sociedade Internacional de Cefaleia, também em 2018, são reconhecidos 150 tipos diferentes de dores de cabeça, e são avaliados sintomas comuns à diversas condições médicas. Realizar o diagnóstico correto é extremamente importante para entender a causa dos problemas de saúde, em alguns casos, a dor de cabeça é o principal sintoma. A revista científica Ciência e Cultura, já em 2011, havia publicado que cerca de 40% das pessoas têm cefaleia com certa regularidade

*(fonte: Cienc. Cult. Vol.63 no.2 São Paulo Apr. 2011).*

O objetivo é o início de estudos que buscam abrir caminhos para um diagnóstico preciso de cefaleia por meio da IA. Estamos propondo uma solução na área neurológica devido a uma defasagem do uso da IA como ferramenta de auxílio. Logo, a nossa solução será voltada para a área da saúde, mais especificamente para os neurologistas que, além de usarem dessa ferramenta, a alimentarão fornecendo dados históricos para treinamentos.

Estamos propondo uma solução para a detecção de dois tipos de cefaleia mais comuns: Sem Aura (Enxaqueca comum) e Com Aura (Enxaqueca Clássica).

## **Fluxo de Criação da Inteligência**

Separando por módulos de desenvolvimento:

1º módulo: Inserir dados extraídos das características de cefaleias do livro Classificação Internacional de Cefaleias.

Neste módulo, lemos dois tipos de cefaleia presentes, a Migrânea sem aura, descrita como:

Critérios diagnósticos:

A. Ao menos cinco crises<sup>1</sup> preenchendo os critérios de B a D

B. Crises de cefaleia durando 4-72 horas (sem tratamento ou com tratamento ineficaz)<sup>2,3</sup>

C. A cefaleia possui ao menos duas das seguintes características:

1. localização unilateral

2. caráter pulsátil

3. intensidade da dor moderada ou forte

4. exacerbada por ou levando o indivíduo a evitar atividades físicas rotineiras (por exemplo: caminhar ou subir escadas)

D. Durante a cefaleia, ao menos um dos seguintes:

1. náusea e/ou vômito

2. fotofobia e fonofobia

E. Não melhor explicada por outro diagnóstico da ICHD-3.

A Migrânea com aura, descrita como:

Critérios diagnósticos:

A. Ao menos duas crises preenchendo os critérios B e C

B. Um ou mais dos seguintes sintomas de aura plenamente reversíveis:

1. visual
2. sensorial
3. fala e/ou linguagem
4. motor
5. tronco cerebral
6. retiniano

C. Ao menos três das seis seguintes características:

1. ao menos um sintoma de aura alastra-se gradualmente por  $\geq 5$  minutos
  2. dois ou mais sintomas de aura ocorrem em sucessão
  3. cada sintoma de aura individual dura 5-60 minutos<sup>1</sup>
  4. ao menos um sintoma de aura é unilateral<sup>2</sup>
  5. ao menos um sintoma de aura é positivo<sup>3</sup>
  6. a aura é acompanhada, ou seguida dentro de 60 minutos, por cefaleia
- D. Não melhor explicada por outro diagnóstico da ICHD-3.

Obtivemos as características de cada doença:

Migrânea sem aura: Tempo de dor, localização, caráter, Intensidade, Exarcebação com atividade cotidiana, náuseas, vômito, fotofobia e fonofobia, cinco ou mais vezes a combinação de B+D, duas ou mais vezes:

Tabela de Características da Migrânea sem Aura		
critério	característica	Medida
B	tempo	Hora
C	Localização	unilateral ou bilateral
C	caráter	pulsátil ou gradual
C	intensidade	forte, moderada ou fraca
C	Exacerbação	sim ou não
D	náuseas	sim ou não
D	vômito	sim ou não
D	fotofobia e fonofobia	sim ou não
$A = 5(B+D)$	5 vezes ou mais de combinação B+D	sim ou não
$\geq 2C$	2 ou mais	sim ou não

Migrânea com aura: localização, aura visual, aura sensorial, aura de fala e/ou linguagem, aura motor, aura tronco cerebral, aura retiniano, ao menos uma aura alastra-se gradualmente por 5 min ou mais, dois ou mais sintomas de aura sucessivos, sintoma unilateral, sintoma positivo de aura, aura acompanhada ou seguida dentro de 60 min por cefaleia.

2º módulo: Separar classificação para cada características, ou seja, as entradas.

3º módulo: Inserir base de dados de pacientes com doenças de cefaleia;

4º módulo: Treinar os dados para diagnóstico da doença.

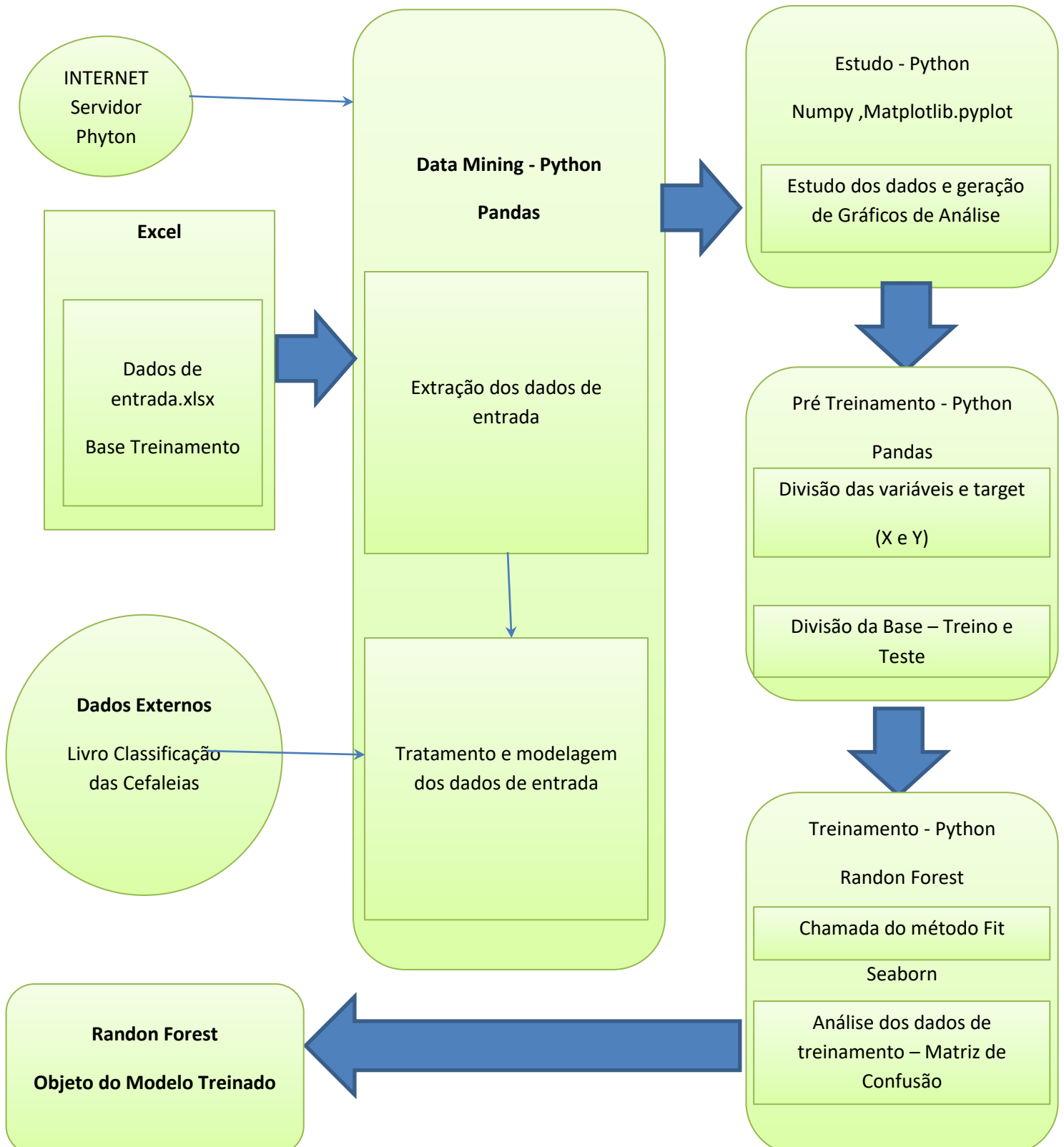
## 2 - Arquitetura Proposta para a Solução

Utilizando a linguagem *python* devido à sua flexibilidade e compatibilidade com ferramentas de treinamento de dados. A apresentação

dos resultados do modelo predito expostos em forma gráfica e de texto através da ferramenta *DataVIS*.

O fluxo da arquitetura se deve pela inserção dos dados é mostrado a seguir:

### A - Treinamento do modelo



**Dados de entrada** - Podem ser em arquivo em *Excel* e em *.csv* – uma base de dados com 43 colunas.

**Extração de dados e *Data Mining*** – utilizando o *Python* e o *Integrated Development Environment* (IDE) *Jupyter* , com a biblioteca *Pandas* para realizar o *Data Mining* e os estudos em cima do banco de dados importado.

**Visualização de Gráficos de estudo da Base** – utilizando a biblioteca *matplotlib.pyplot* e *Pandas* para visualizar os dados do estudo realizado na base e estatísticas retiradas da base de forma gráfica.

**Aplicação de técnicas de *Machine Learning*** – Com a variedade de funções da biblioteca *sklearn* para *machine learning*, utilizando os módulos para *machine learning* e utilizando a biblioteca *seaborn* para a extração das matrizes de confusão para auxiliar nas análises e performance dos treinamentos.

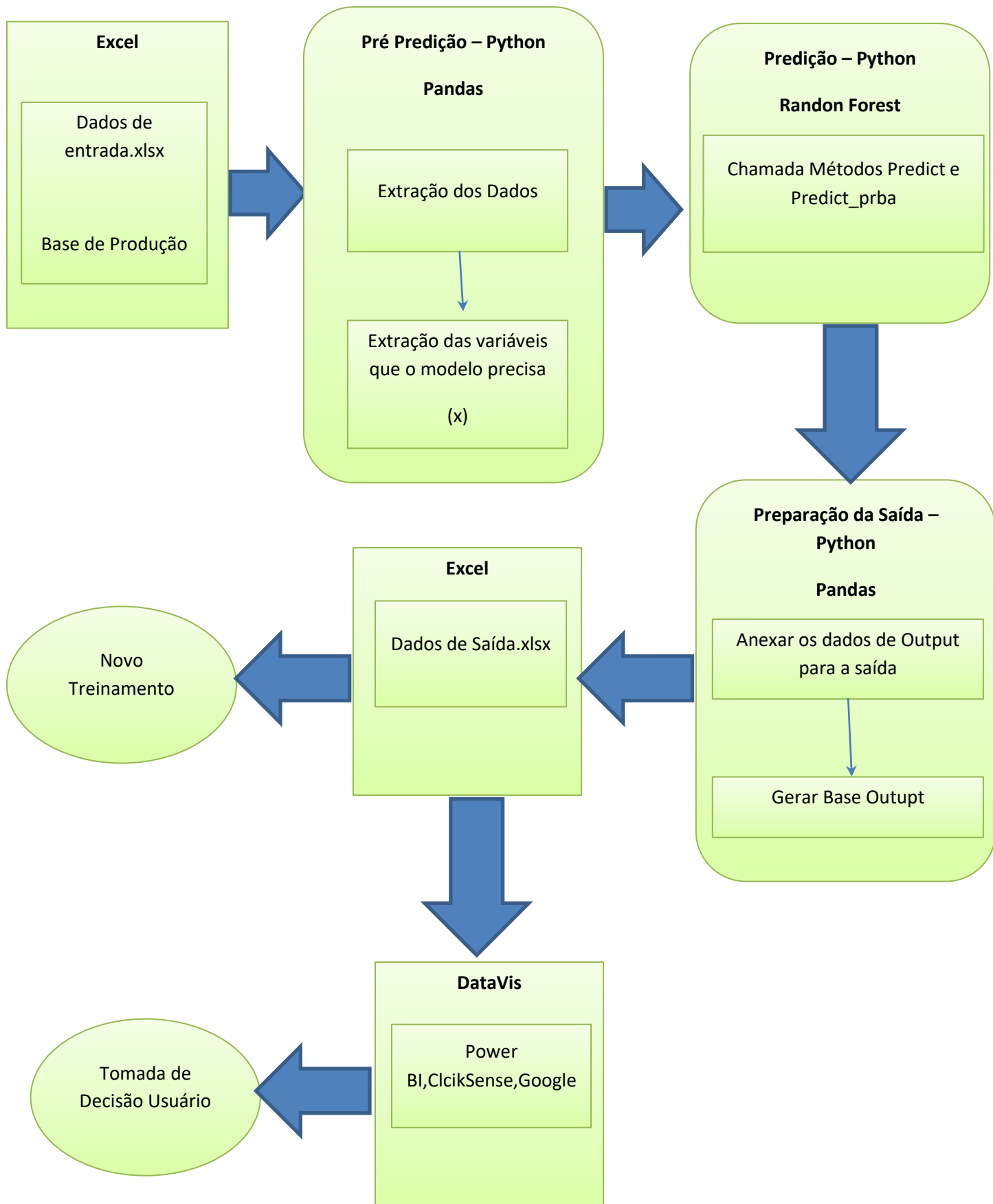
**Treinamento *Random Forest*** – Aplicação a Biblioteca *RandomForestClassifier* para o Treinamento do modelo através do método *fit*.

**Retorno o objeto Treinado** – Após o treinamento o objeto *RandomForestClassifier* é retornado

```
# Random Forest

random_forest = RandomForestClassifier(n_estimators=100)
random_forest.fit(X_train, y_train)
#Predict Output
rf_predicted = random_forest.predict(X_test)
```

## B – Predição do modelo



**Dados de entrada** - Podem ser em arquivo em *Excel* e em .csv, uma base de dados com 43 colunas.

**Aplicação da função de predição** – Após a recepção dos dados de entrada através da biblioteca *panas* do *Phyton* é separado os dados que serão usados na predição (variável *X\_INPUT*) e é chamada a função de predição através do objeto treinado (*predict* e *predict\_proba* do objeto *RandonForestClassifier*)

**Retorno dos Dados** – Por fim é retornada uma base com os scores e o retorno da predição do modelo preditivo

**Apresentação dos dados** - Os dados de retorno podem ser apresentados para o usuário uma ferramenta de *data Vis* (*Power BI*, *Click Sense*, etc), lendo os dados do retorno dos dados treinados com as bibliotecas de IA .Eles também podem ser utilizados como dados de treinamento para novos treinamentos da inteligência

### **3 - Técnica Utilizada – Machine Learning – Randon Forest**

Aprendizagem Supervisionada: De acordo com as regras para a detecção dos dois tipos de cefaleia mais comuns que retiramos do livro, iremos treiná-las com a base para marcar qual registro tem ou não cefaleia e de qual tipo.

Dentro desse modelo , submetemos a base gerada a dois tipos de técnicas, a fim de avaliar a mais vantajosa e performática para o nosso modelo de dados: Regressão Logística e o modelo de Árvore de decisão *Randon Forest*.

#### **Regressão Logística**

É utilizado para problemas de classificação binária. A função logística transforma qualquer valor no intervalo de 0 a 1, aplicando uma regra à saída da função logística para ajustar valores para 0 e 1 e prever um valor de classe.



## Regressão logística binominal

Aplicado ao modelo de detecção de cefaleia, cada sintoma existente deve ser classificado entre tem ou não tem, sendo representado respectivamente por 1 e 0.

## Random Forest

Os algoritmos Random Forest são criados por árvores de decisão e utilizando a técnica de combinação de modelos. Trazendo uma resposta de tendências para os dados apresentados, no caso, possíveis chances de possuir ou não uma doença.

Tendo em vista a nossa disponibilidade de variáveis inseridas para treinamento dos dados, a técnica utilizada de *Machine Learning* que mais se adequa é a Random Forest, de aprendizagem supervisionada, pois em comparação com a regressão logística ela apresentou melhores resultados (Percentual de acuracidade : 92.6 contra 67.4 da Regressão Linear) e pelo fato de funcionar bem no nosso escopo e projeto de análise com combinações e sintomas para identificar doenças.

Adotamos também o uso do Random Forest pelo fato dele funcionar bem com a quantidade de dados de treino que nós temos. Realizamos testes também com menos dados(100 registros) e o resultado de acuracidade foi menor.

Comparação entre os modelos submetendo o treinamento com 5000 registros:

	Model	Score	Test Score
2	Random Forest	100.00	92.60
1	Gaussian Naive Bayes	74.43	74.33
0	Logistic Regression	70.23	67.40

#### 4 – Ferramentas Envolvidas no projeto

**A - Dados de entrada** – Uma base com os sintomas do paciente , gerada de maneira sintética através das regras retiradas do Livro “Classificação Internacional das Cefaleias”

Foi criado um algoritmo para a geração desses dados conforme a classificação dos sintomas do livro:

Campos	Características
Idade	de 5 até 80 anos
Gênero	Masculino e Feminino
Tempo de Dor	Em Minutos
Localização	1 - Unilateral 2- bilateral 3 - frontal 4 - traseiro
Caráter	0 - Não ,1- Sim
Intensidade	1- Fraca ,2- Moderada, 3- Forte
Exacerbação	0 - Não ,1- Sim
Náuseas	0 - Não ,1- Sim
Fotofobia	0 - Não ,1- Sim
Tempo de Adormecimento Durante a Crise	Em Minutos
Alodinia cutânea	0 - Não ,1- Sim
autônomo craniano	0 - Não ,1- Sim
Menstruação	0 - Não ,1- Sim
Tempo Aura visual	Em Minutos
Tempo Aura Sensorial	Em Minutos
Tempo Aura Fala Linguagem	Em Minutos
Tempo Aura Motor	Em Minutos
Tempo Aura Tronco Cerebral	Em Minutos
Tempo Aura Retiniano	Em Minutos
Aura alastra-se gradualmente por 5 min ou mais	0 - Não,1- Sim
Aura Unilateral	0 - Não ,1- Sim
Aura Positivo	0 - Não ,1- Sim
Cefaleia	0 - Não ,1- Sim

Cefaleia acompanhada dentro de 60 min	0 - Não ,1- Sim
Afasia (unilateral)	0 - Não,1- Sim
Disartria(unilateral)	0 - Não ,1- Sim
Disartria(bilateral)	0 - Não ,1- Sim
reversibilidade plena	0 - Não ,1- Sim
Fraqueza Muscular	0 - Não ,1- Sim
Vertigem	0 - Não ,1- Sim
zumbido	0 - Não ,1- Sim
Hipoacusia (perda de audição)	0 - Não ,1- Sim
Diplopia (alteração da posição dos olhos que causa visão dupla)	0 - Não ,1- Sim
ataxia não atribuível ao déficit sensorial	0 - Não ,1- Sim
diminuição no nível da consciência	0 - Não ,1- Sim
sintomas de migrânea hemiplégica	0 - Não ,1- Sim
Mutação no gene CACNA1A	0 - Não ,1- Sim
Mutação no gene ATP1A2	0 - Não ,1- Sim
Mutação no gene SCN1A	0 - Não ,1- Sim
desenho do paciente de um defeito de campo monocular	0 - Não ,1- Sim
exame clínico de campo visual	Resultado do Exame

## B - Processamento para classificar qual registro da base que tem cefaleia positiva e para qual tipo (Com Aurea e Sem Aurea)

Após extrair a base dentro do *python*, foi criado um processamento de Data Mining dentro do Python com a biblioteca Pandas para tratar e classificar seguindo as regras do livro para a detecção de cefaleia. Foram trabalhados 2 tipos de cefaleia mais comuns, conhecida como: Enxaqueca Comum (Cefaleia Sem Aurea) e Enxaqueca Clássica (Cefaleia com Aurea).

Após essa Classificação foram criados mais 5 campos no DataSet:

**CefaleiaTipo1SimNao** – 0- Não, 1- Sim – Detecta ou não a presença da cefaleia tipo 1- Sem Aurea

**CefaleiaTipo2SimNao** – 0- Não, 1- Sim – Detecta ou não a presença da cefaleia tipo 2- Com Aurea

**CefaleiaSimNao** – 0- Não , 1- Sim – Detecta ou não a presença da cefaleia (Se um ou outro campo acima estiver marcado )

**Cefaleia2TiposSimNao** – 0- Não , 1- Sim – Indicador para a possibilidade da pessoa poder ter os 2 tipos a partir dos sintomas acima

**CefaleiaClassificacao** – Campo para a aplicação da seguinte classificação das categorias em um campo só. Essa classificação é baseada no resultado dos campos acima conforme abaixo:

- 0 - Sem cefaleia
- 1 - Tipo1 (Sem aura – Enxaqueca comum)
- 2- Tipo2 (Sem aura – Enxaqueca clássica)
- 3 - Tipo1 ou tipo2

Por esse campo é caracterizada cefaleia se o valor for maior que 0.

```
df.loc[
    (df['Tempo de Dor'] >= 240 ) &
    (
        (df['Localizacao'] == 1) |
        (df['Carater'] == 1) |
        (df['Exacerbação'] == 1) |
        (df['Intensidade'].isin([2,3])
    )) &
    (
        (df['Náuseas'] == 1) |
        (df['Fotofobia'] == 1)
    ),
    'CefaleiaTipo1SimNao'] = 1

#Aplicando o Critério do Livro para Cefaleia Clássica Com Aurea
df.loc[
    (df['Aura alastra-se gradualmente por 5 min ou mais'] == 1) &
    (
        (df['Tempo Aura visual']).between(5, 60, inclusive=False) |
        (df['Tempo Aura Sensorial']).between(5, 60, inclusive=False) |
        (df['Tempo Aura Fala Linguagem']).between(5, 60, inclusive=False) |
        (df['Tempo Aura Motor']).between(5, 60, inclusive=False) |
        (df['Tempo Aura Tronco Cerebral']).between(5, 50, inclusive=False) |
        (df['Tempo Aura Retiniano']).between(5, 60, inclusive=False)
    ) &
    (
        (df['Aura Unilateral'] == 1) |
        (df['Aura Positivo'] == 1) | (df['Cefaleia'] == 1) |
        (df['Cefaleia acompanhada dentro de 60 min'] == 1)
    )
]
```

```
(
    (df['Aura Unilateral'] == 1) |
    (df['Aura Positivo'] == 1) | (df['Cefaleia'] == 1) |
    (df['Cefaleia acompanhada dentro de 60 min'] == 1)
),
'CefaleiaTipo2SimNao'] = 1

#Se Qualquer dos indicadores forem verdadeiro , está caracterizada a cefaleia
df.loc[(df['CefaleiaTipo1SimNao'] == 1) | (df['CefaleiaTipo2SimNao'] == 1) , 'CefaleiaSimNao'] = 1

#Avaliando Possibilidade de ser
df.loc[(df['CefaleiaTipo1SimNao'] == 1) & (df['CefaleiaTipo2SimNao'] == 1) , 'Cefaleia2TiposSimNao'] = 1

[87] > %> ML

#Aplicando classificação das categorias em um campo só 0 - Sem cefaleia 1- Tipo1 2- Tipo2 - 3 - Tipo1 ou tipo2
df.loc[(df['CefaleiaSimNao'] == 0) , 'CefaleiaClassificacao'] = 0
df.loc[(df['CefaleiaSimNao'] == 1) & (df['CefaleiaTipo1SimNao'] == 1) & (df['Cefaleia2TiposSimNao'] == 0) ,
'CefaleiaClassificacao'] = 1
df.loc[(df['CefaleiaSimNao'] == 1) & (df['CefaleiaTipo2SimNao'] == 1) & (df['Cefaleia2TiposSimNao'] == 0) ,
'CefaleiaClassificacao'] = 2
df.loc[(df['Cefaleia2TiposSimNao'] == 1) , 'CefaleiaClassificacao'] = 3

[88] > %> ML

print(df['CefaleiaClassificacao'].value_counts())

0    2849
1     1182
2     1135
3       634
Name: CefaleiaClassificacao, dtype: int64
```

## C- Realizar estatísticas em relação às informações da base gerada e classificada com as informações

Após a detecção desses campos. Fizemos algumas estatísticas na base e tiramos alguns indicadores como, por exemplo:

- Total de pacientes com Cefaleia geral e por tipo;
- Agrupamento por idade e por Gênero;
- Relação da cefaleia entre tempo de dor e idade.

## D- Aplicação de ferramentas de Machine Learning

Após a geração destas estatísticas iremos aplicar as ferramentas de Machine Learning (Random Forest) de acordo com as características de nossa base e realizando a separação entre base de treinamento e teste e aplicando o treinamento correspondente e gerando o modelo de Apoio à decisão .

Para o treinamento , realizamos a retirada das variáveis e saída(campos criados para classificação dentro do dataset) e gênero da base para a variável X e para a variável Y colocamos o campo final de classificação da cefaleia (CefaleiaClassificacao) conforme imagens:

```
X = df.drop(['Genero', 'CefaleiaSimNao', 'CefaleiaTipo1SimNao', 'CefaleiaTipo2SimNao', 'Cefaleia2TiposSimNao',
'CefaleiaClassificacao'], axis=1)
X.head(3)
```

	Idade	Tempo de Dor	Localizacao	Carater	Intensidade	Exacerbação	Náuseas	Fotofobia	Adormecimento Durante a Crise	alodinia cutânea	...	randomlistAH	ataxia não atribuível ao déficit sensorial	dim no cons
0	52	48	3	1	3	1	1	0	101	0	...	0	1	
1	74	50	4	0	2	0	0	0	79	1	...	0	1	
2	35	101	1	1	2	0	1	1	53	0	...	1	0	

```
3 rows x 42 columns
```

```
y = df['CefaleiaClassificacao']
liver_corr = X.corr()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=101)
```

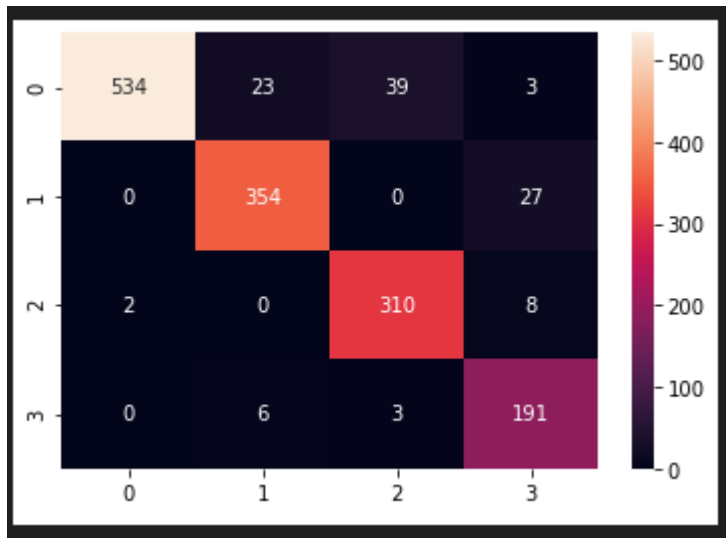
```
liver_corr = X.corr()
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=101)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=101)
```

The image shows a screenshot of a Jupyter Notebook environment within the Visual Studio Code (VS Code) editor. The interface is divided into several panels. On the left, the 'EXPLORER' panel shows a file tree with various files, including notebooks and data files. The main editor area displays a Jupyter Notebook with a single cell containing Python code for a Random Forest classifier. The code imports necessary libraries, loads data, splits it into training and testing sets, trains a Random Forest model, and evaluates its performance. Below the code cell, the output of the notebook is visible, showing the results of the model's performance metrics. The bottom status bar indicates the Python version (3.8.3) and the bitness (64-bit). The system tray at the very bottom shows the date and time (02/12/2020, 18:45).

Aplicamos também a métrica para a criação das matrizes de confusão para avaliar os resultados e comparar com as informações de classificação:



Gerando o modelo de apoio à decisão apoiada nesse modelo supervisionado e com o devido treinamento aplicado a base para o retorno ao usuário.

### E- Geração dos dados de Saída

Para o processamento dos dados de produção realizamos o recebimento dos dados e o tratamento e tratamento de separação de variáveis de entrada e realizamos a chamada do método para realizar o predict das informações e retorno das informações :

```
BaseImput = pd.read_excel(r'C:\Users\MARCOSVINICIUS\Documents\Aulas Pós\SAD\baseCefaleia2500Prd.xlsx')
ScoreBase(BaseImput,random_forest)
X_input = BaseImput.drop(['Genero'], axis=1)
len(X_input)
```

```
def ScoreBase(BaseInput,ObjetoTreinado):

    X_input = BaseInput.drop(['Genero'], axis=1)

    y_Prob = ObjetoTreinado.predict_proba(X_input)
    y_output = ObjetoTreinado.predict(X_input)

    BaseInput["CefaleiaNao"] = list(y_Prob[:,0])
    BaseInput["CefaleiaSim"] = list(y_Prob[:,1] + y_Prob[:,2] + y_Prob[:,3])
    BaseInput["CefaleiaTipo1"] = list(y_Prob[:,1])
    BaseInput["CefaleiaTipo2"] = list(y_Prob[:,2])
    BaseInput["Cefaleia2Tipos"] = list(y_Prob[:,3])

    BaseInput["CefaleiaTipo1"] = ((BaseInput["CefaleiaTipo1"] * 100)/BaseInput["CefaleiaSim"])/100
    BaseInput["CefaleiaTipo2"] = ((BaseInput["CefaleiaTipo2"] * 100)/BaseInput["CefaleiaSim"])/100
    BaseInput["Cefaleia2Tipos"] = ((BaseInput["Cefaleia2Tipos"] * 100)/BaseInput["CefaleiaSim"])/100

    BaseInput["ClassificacaoCefaleia"] = y_output #Aplicando classificação das categorias em um campo só 0 -
Sem cefaleia 1- Tipo1 2- Tipo2 - 3 - Tipo1 ou tipo2

    BaseInput.to_excel('C:/Users/MARCOSVINICIUS/Documents/Aulas Pós/SAD/model_output.csv',columns=['Idade',
'Genero','Tempo de Dor','Localizacao','Carater','Exacerbação','Intensidade','Náuseas','Fotofobia','Aura
alastra-se gradualmente por 5 min ou mais','Tempo Aura visual','Tempo Aura Sensorial','Tempo Aura Fala
Linguagem','Tempo Aura Motor','Tempo Aura Tronco Cerebral','Tempo Aura Retiniano','Aura Unilateral','Aura
Positivo','Cefaleia acompanhada dentro de 60 min','CefaleiaNao','CefaleiaSim','CefaleiaTipo1','CefaleiaTipo2',
'Cefaleia2Tipos','ClassificacaoCefaleia'])
```

Em relação as informações de retorno basicamente são retornados 5 campos de probabilidade e 1 campo com a classificação retornada:

- CefaleiaNao - Percentual de probabilidade de não ser cefaleia;
- CefaleiaSim - Percentual de probabilidade de ser cefaleia;
- CefaleiaTipo1 - Percentual de probabilidade de ser cefaleia do tipo 1  
Sem Aura ,em cima do percentual da probabilidade de ser cefaleia;
- CefaleiaTipo2 - Percentual de probabilidade de ser cefaleia do tipo 2  
Com Aura ,em cima do percentual da probabilidade de ser cefaleia;
- Cefaleia2Tipos - Percentual de probabilidade de ser um ou outro tipo de cefaleia . em cima do percentual de probabilidade de ser cefaleia
- ClassificaçãoCefaleia - Campo retornado pela função “predict” do Randon forest com o classificador retornado.

Para o cálculo dos campos CefaleiaTipo1 , CefaleiaTipo2 e Cefaleia2Tipos é realizado um tratamento na saída do resultado predito aplicando uma regra de três através do valor do campo CefaleiaSim.

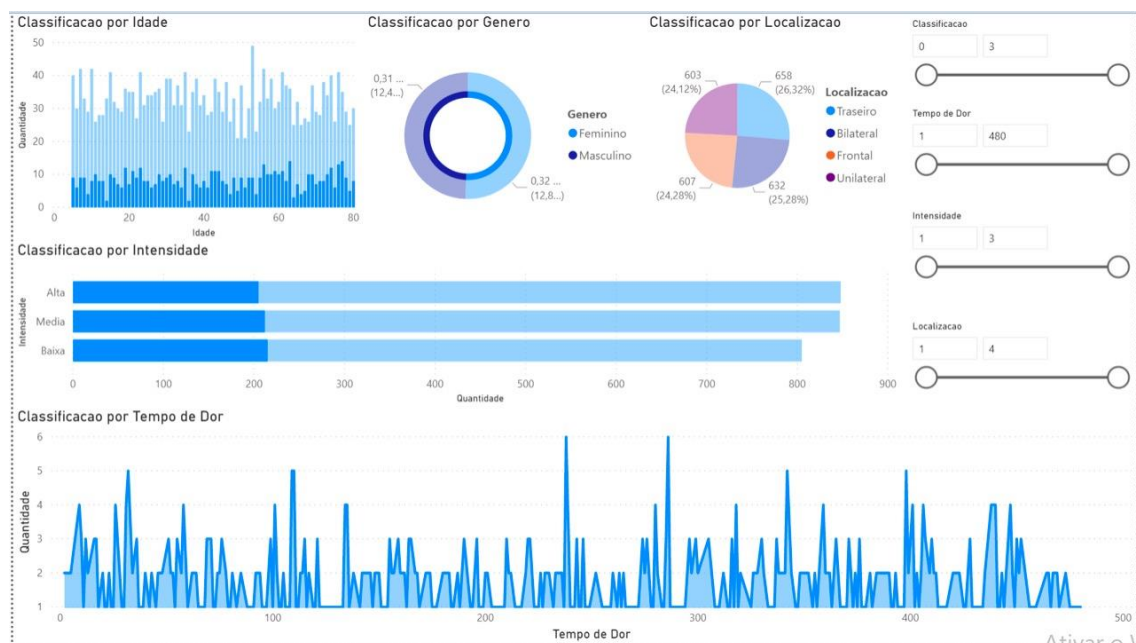


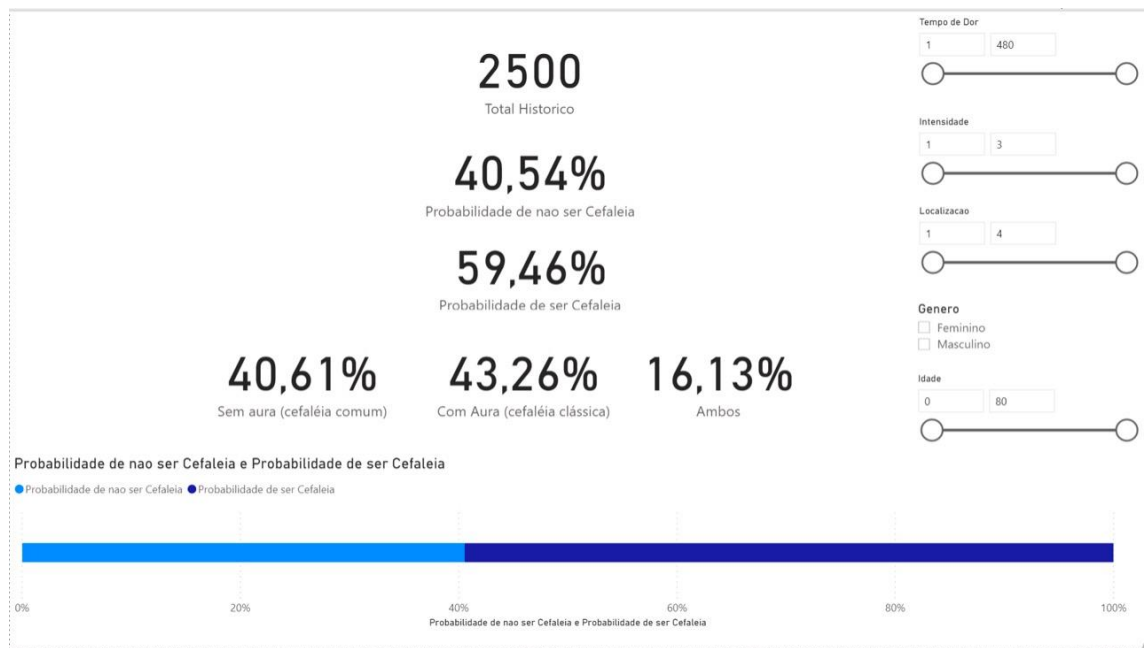
	U	V	W	X	Y	Z	
60 min	CefaleiaNao	CefaleiaSim	CefaleiaTipo1	CefaleiaTipo2	Cefaleia2Tipos	ClassificacaoCefaleia	
0	0,19	0,81	0,913580247	0,037037037	0,049382716	1	
0	0,78	0,22	0,545454545	0,363636364	0,090909091	0	
0	0,28	0,72	0,041666667	0,819444444	0,138888889	2	
0	0,07	0,93	0,655913978	0,053763441	0,290322581	1	
0	0,17	0,83	0,891566265	0,012048193	0,096385542	1	
0	0,85	0,15	0,466666667	0,466666667	0,066666667	0	
0	0,67	0,33	0,787878788	0,121212121	0,090909091	0	
1	0,83	0,17	0,529411765	0,352941176	0,117647059	0	
1	0,76	0,24	0,333333333	0,625	0,041666667	0	
0	0,83	0,17	0,588235294	0,294117647	0,117647059	0	
0	0,32	0,68	0,044117647	0,911764706	0,044117647	2	
1	0,12	0,88	0,090909091	0,602272727	0,306818182	2	
0	0,15	0,85	0,329411765	0,282352941	0,388235294	3	

O retorno será em base de dados (Excel ou CSV) com as variáveis de entrada e com o “append” das informações de saída acima para que possa ser gerado Dashboards através de alguma ferramenta de DATAVIS (Power BI) a consulta das informações deste modelo preditivo e de classificação.

Os dados de saída estão estruturados para também servir de base de entrada para treinamentos posteriores.

Exemplo de visualização de retorno no Power BI:





**Grupo 5:**

**Luize Casaretti**

**Marcos Sousa**

**Rafael Reis**

**Mauro Yida Nunes**