Trabalho da Disciplina SCC0275 - Ciência de Dados

Alexandre Norcia Medeiros	NUSP: 10295583
Caio Abreu de Oliveira Ribeiro	NUSP: 10262839
Daniel Penna Chaves Bertazzo	NUSP: 10349561
Vinicius Torres Dutra Maia da Costa	NUSP: 10262781

ICMC USP SÃO CARLOS

Análise de Classificadores para o dataset MoCap

- Análise e pré-processamento dos dados brutos
- Definição dos modelos de classificação
- Otimização de hiperparâmetros (*tuning*)
- K-Fold Cross Validation
- Comparação da acurácia dos modelos

ICMC USP SÃO CARLOS

Dataset *MoCap*

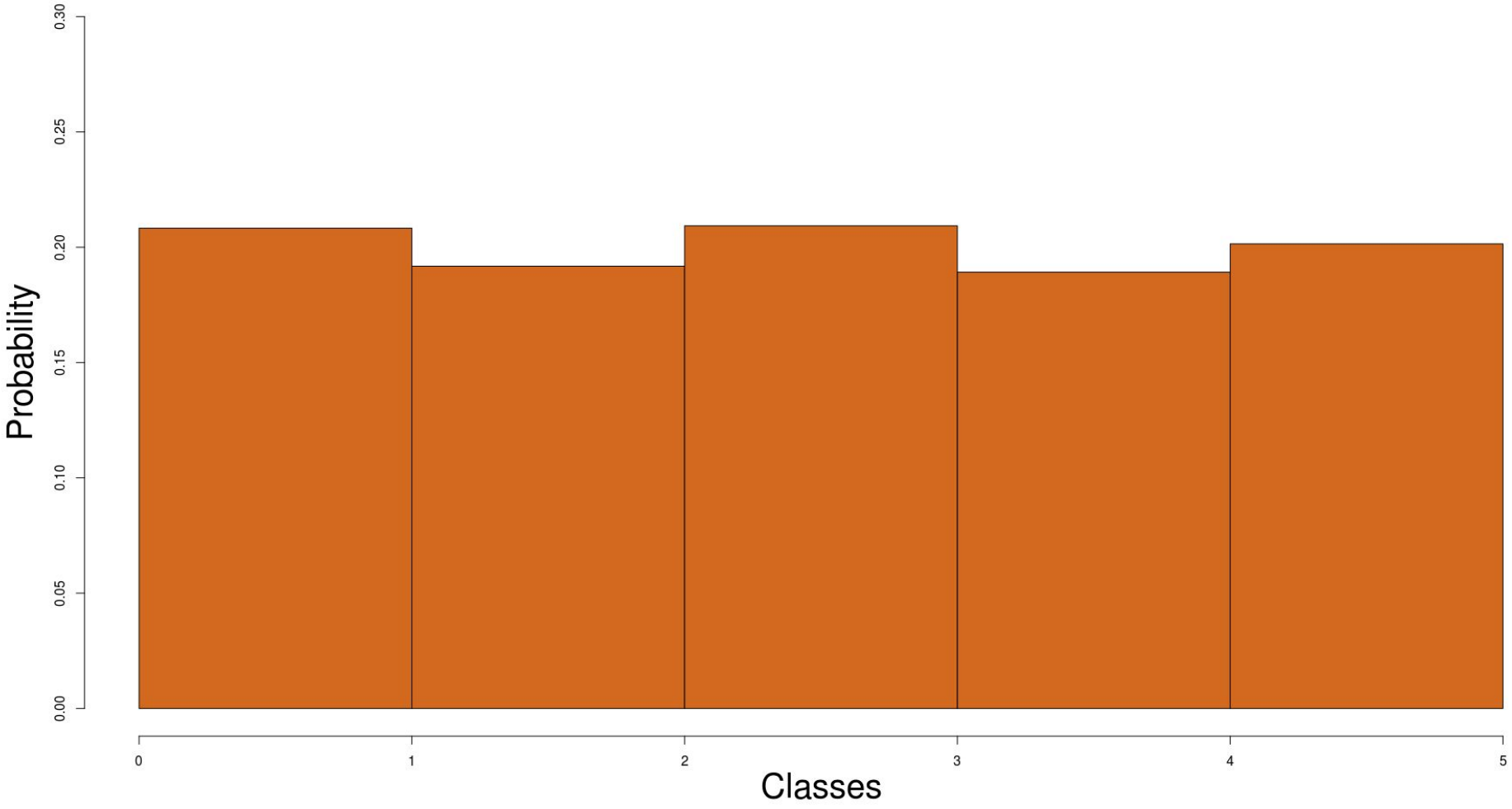
- Obtido por meio do *UCI - Machine Learning Repository*
- Classificação entre 5 posições de mão analisando sensores de movimento em uma luva de captura de movimento
- 78.095 instâncias com 36 atributos (intervalo dos reais)
- Dados reais, com diversos valores nulos e não normalizados

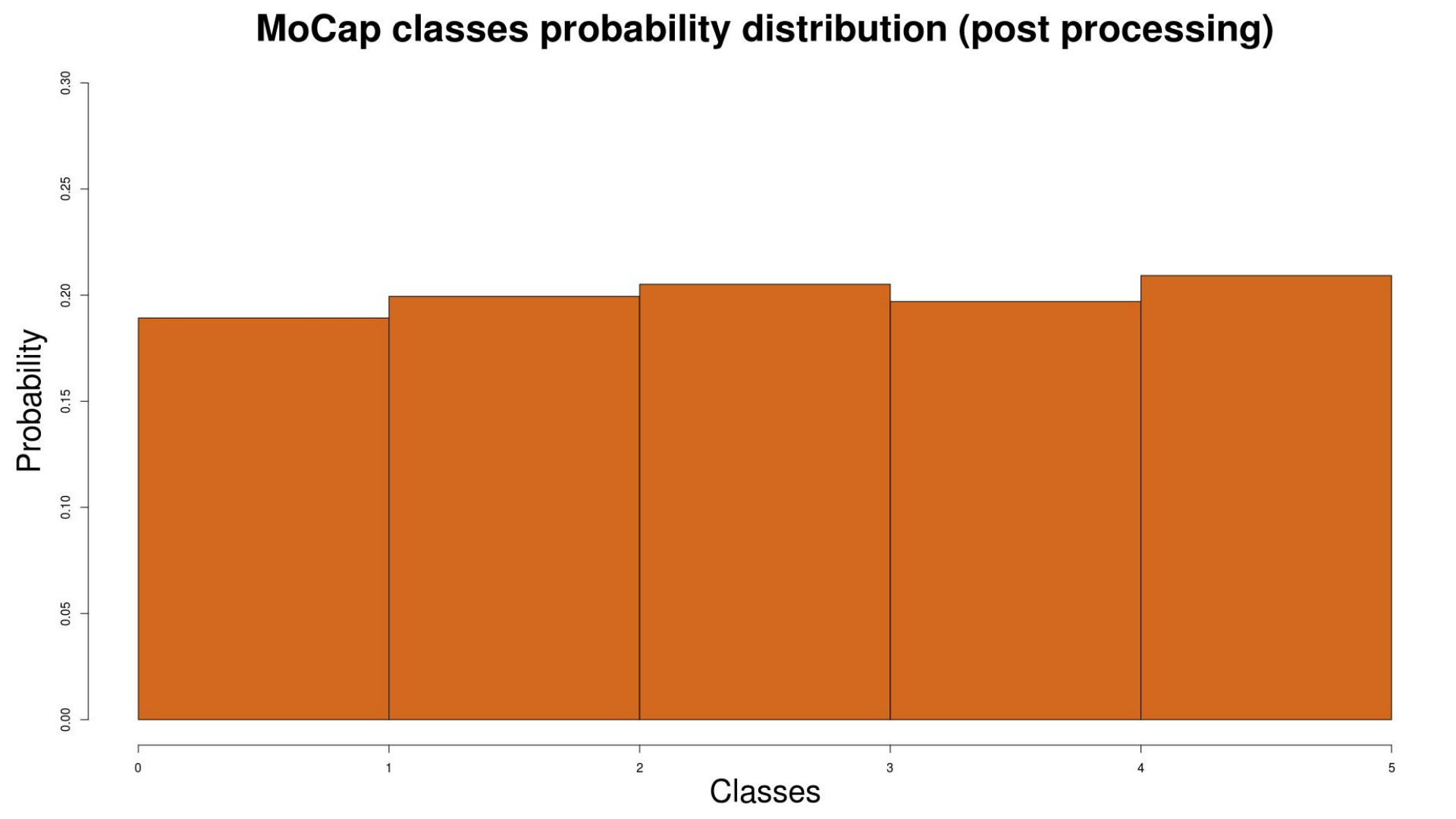
ICMC USP SÃO CARLOS

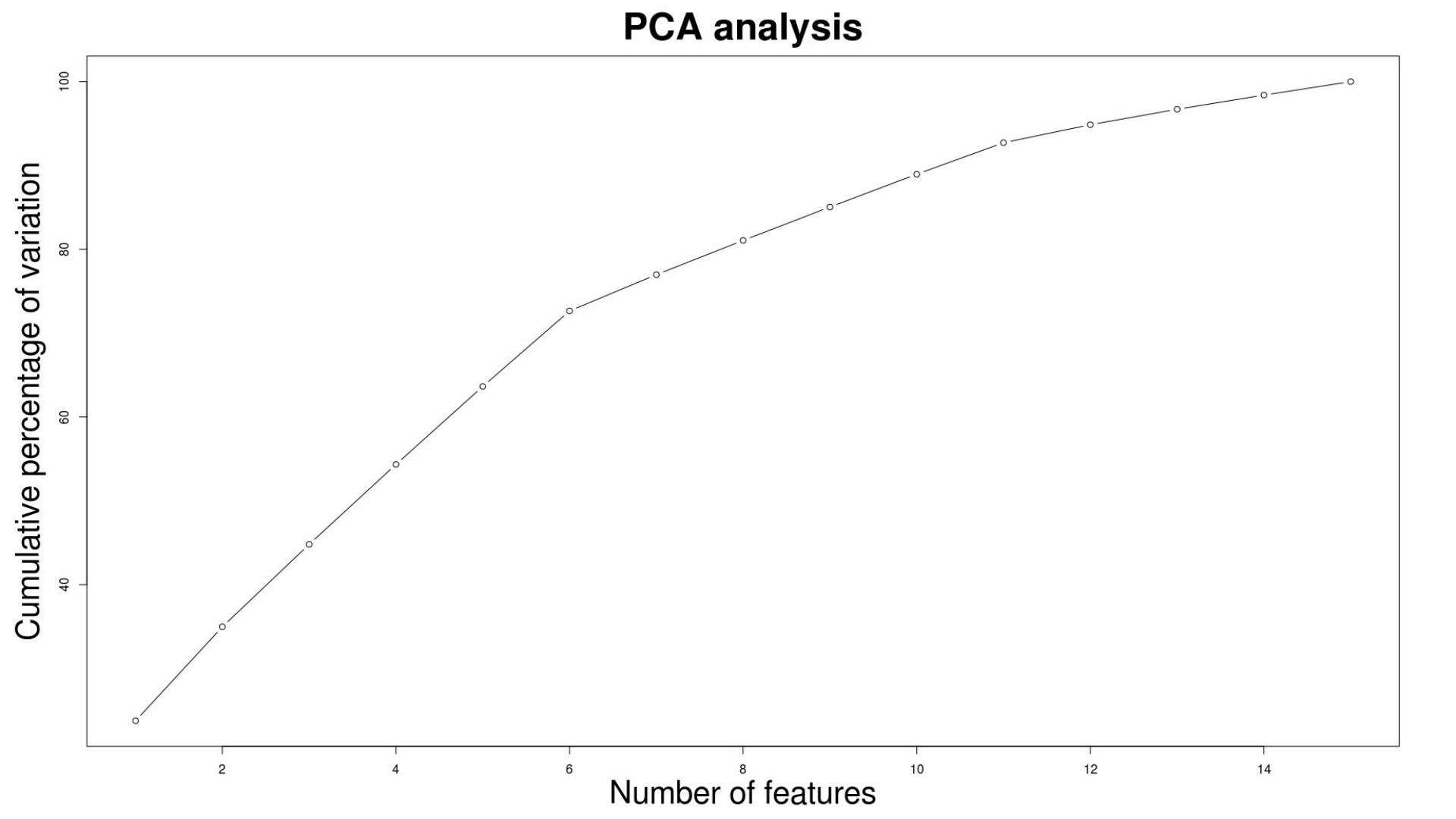
Pré-processamento dos dados

- Redução de 36 atributos (12 sensores) para 15 atributos (5 sensores) devido ao excesso de valores nulos
- Eliminou-se instâncias que ainda possuíam valores nulos
- Restaram 74.976 instâncias, as quais preservaram a distribuição de probabilidade das classes semelhante aos dados iniciais
- Realizou-se uma análise dos principais componentes (PCA)
- Normalizou-se os dados

ICMC USP SÃO CARLOS


MoCap classes probability distribution
Probability
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
1
2
3
4
5
Classes


MoCap classes probability distribution (post processing)
Probability
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0
1
2
3
4
5
Classes


PCA analysis
Cumulative percentage of variation
Number of features
100
80
60
40
2
4
6
8
10
12
14

Definição dos modelos de classificação

- K-nearest neighbors (KNN)
- Naive Bayes
- Support-Vector Machine (SVM)
- Multilayer Perceptron (MLP)
- Random Forest

Otimização dos hiperparâmetros

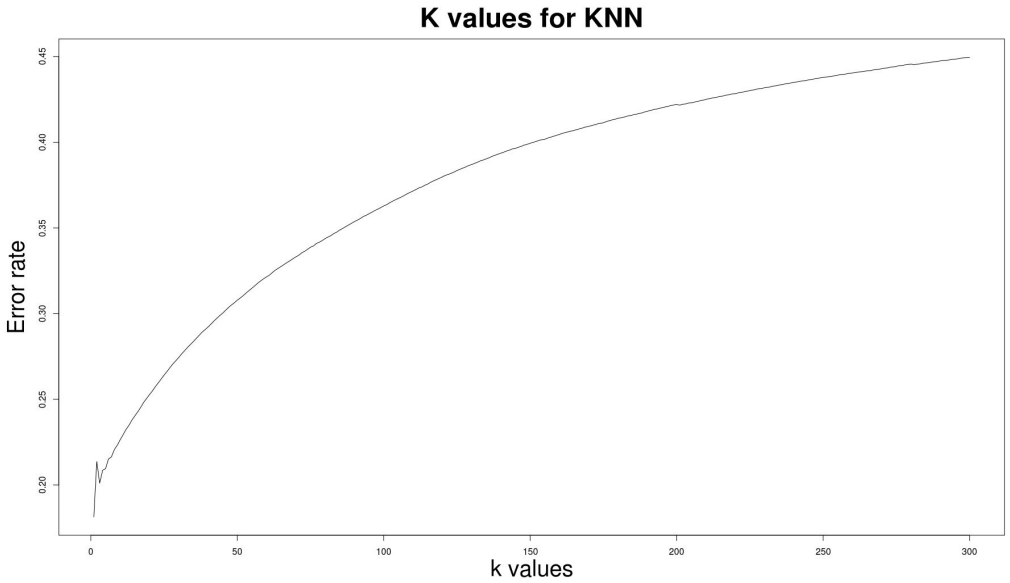
- Extrair os valores dos parâmetros de cada método, visando a melhor performance possível
- Cálculo do erro médio variando os parâmetros, por meio do *k-fold cross validation*
- Valor de K no modelo KNN
- *Kernel function* no modelo SVM
- Tamanho da camada escondida e eta (decaimento dos pesos) no modelo MLP
- Número de árvores no modelo Random Forest

Support-Vector Machine

Kernels	*Radial*	*Linear*	*Polynomial*	*Sigmoid*
Mean Error	1,605	3,713	2,417	5,404

K-Nearest Neighbors

- Escolher o melhor K a partir do erro empírico médio obtido aplicando o KNN com K-fold cross validation para vários valores de K



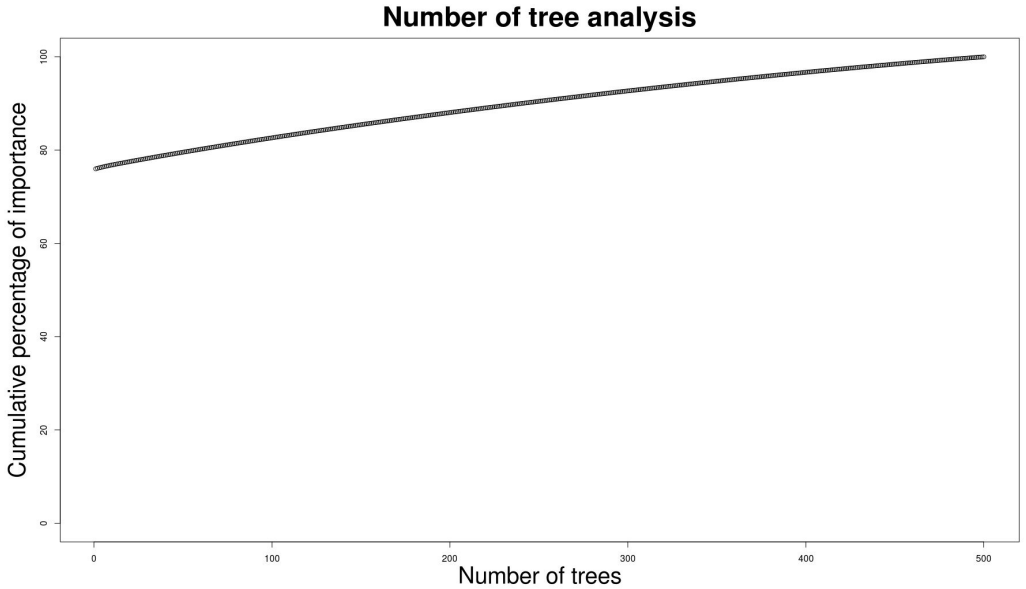
Multilayer Perceptron

- Escolher o número de neurônios na camada escondida e o fator de decaimento a partir do teste empírico, visando reduzir o erro médio

Hidden Layer Size	Decay	Error
15	0.005	0.2694
20	0.005	0.2163
30	0.005	0.2082
30	1	0.1712

Random Forest

- Realizou-se uma análise da importância de cada árvore em um conjunto de 500 árvores



Stratified K-fold cross validation

Modelo	MLP	SVM	KNN	Naive Bayes	Random Forest
Resultados	0.1714 ± 0.0049	0.1601 ± 0.0014	0.1981 ± 0.0028	0.3457 ± 0.0025	0.0829 ± 0.0021

Comparação da acurácia

- O método com melhor eficiência, para o caso estudado, foi o random forest com 300 árvores
- Este método obteve 91,7% de acurácia
- Os testes foram realizados utilizando a versão *stratified* do *k-fold*, a qual mantém a distribuição das classes nos diferentes *folds*.

Conclusão

- O melhor método encontrado obteve uma acurácia boa para um conjunto de dados reais e complexos
- Porém, notou-se o quão dependente são os resultados dos parâmetros determinados para cada método
- Em diferentes *datasets*, os resultados serão diferentes, pois o treinamento depende dos dados
- O método de comparação final desenvolvido foi generalizado para utilizar diferentes *datasets*.