

# Modelos de Classificação Aplicados à Predição do Nível de QoE de Usuários em uma Rede de Telefonia Móvel

Ingrid S. M. Furtado, Ezequias M. S. de Santana Jr., Yuri S. Ribeiro

Engenharia de Telecomunicações

Departamento de Engenharia de Teleinformática (DETI)

Universidade Federal do Ceará, Fortaleza, Brasil.

**Resumo**—Este artigo aborda a aplicação de modelos de classificação para a predição dos níveis de qualidade de experiência (*quality of experience*, QoE) de usuários de uma rede sem fio. O processo consiste na análise exploratória de dados (*exploratory data analysis*, EDA) dos valores de alguns indicadores-chave de desempenho (*key performance indicators*, KPI) para determinar possíveis relações e a relevância de cada preditor, permitindo uma escolha adequada dos parâmetros para o modelo de regressão e, em seguida, a aplicação dos modelos estudados em sala de aula sobre dados a fim de traçar um comparativo entre as estratégias de aprendizado.

**Palavras-Chave**—modelos de classificação, redes neurais, qualidade de experiência, *mean opinion score*, redes móveis.

## I. INTRODUÇÃO

Atualmente, a demanda por bons serviços de transmissão de dados móvel é notória, visto que a aplicabilidade de sistemas sem-fio já se estende em diversos campos da indústria principalmente no contexto da quinta geração (*fifth generation*, 5G) [1]. Nesse cenário, o monitoramento do desempenho desses sistemas e da qualidade que os usuários experienciam é de fundamental importância para a otimização e manutenção dos serviços (eg. voz e dados) de acordo com os novos padrões de sistema como maiores taxas e menor latência.

Para verificar a qualidade de uma rede móvel, mais especificamente a condição de um enlace, existem parâmetros de desempenho que podem ser medidos e informados à rede. Alguns deles são medidas quantitativas e outras qualitativas, como potência do sinal recebido por um usuário e a pontuação de opinião média (*mean opinion score*, MOS) [2], respectivamente.

Neste trabalho, com a posse de dados referentes à diversas medições de KPI serão aplicados modelos de classificação sobre essas medidas com o intuito de observar o desempenho dessas estratégias de aprendizado estatístico a fim de observar como esses indicadores se relacionam com a qualidade do serviço experienciado pelos usuários, visando a construção de um modelo para a estimação onde a saída é o índice de MOS dos usuários da rede em função desses indicadores.

O restante deste trabalho está organizado da seguinte forma. Na Seção II, será comentada a metodologia desenvolvida. Na Seção III, serão apresentados e discutidos os resultados dos modelos aplicados e, por fim, na Seção IV serão apresentadas as conclusões.

## II. METODOLOGIA

### A. Conjunto de dados

O conjunto de dados selecionado foi obtido na plataforma Kaggle no endereço indicado em [3]. Esse conjunto consiste em várias observações referentes à KPIs, que foram obtidas em campanhas de testes de chamadas de voz realizadas com *smartphones* em uma rede sem-fio. O objetivo com o uso desse conjunto de dados é prever o nível de QoE dos usuários, no caso o MOS, com base em outros indicadores-chave de desempenho.

O conjunto de dados consistia em 105828 amostras com 8 preditores, correspondendo aos indicadores: data do teste, força do sinal em *dBm*, velocidade do móvel em *m/s*, distância da estação base em *m*, duração da chamada de teste em *s*, resultado do teste, tecnologia empregada e tempo para estabelecimento da chamada de teste em *s* juntamente com o respectivo valor de MOS, representando as 5 classes (L).

Durante a etapa de pré-processamento dos dados, os conjunto de dados passou a ter 68025 amostras com 6 preditores, pois Call Test Result e Call Test Technology foram retirados.

### B. LDA E QDA

Como mostrado nos trabalhos anteriores, ao se desejar uma redução da dimensionalidade dos dados, pode-se utilizar o PCA (Análise de Componentes Principais), porém a direção apontada pelos autovetores não necessariamente indicam a melhor direção com a finalidade de classificação. Assim, nesse trabalho foi proposto um critério de maximização de separação entre classes e que ao mesmo busca minimizar o espalhamento dentro delas.

Primeiramente, foi utilizado o modelo de classificação linear através da análise de discriminantes lineares (LDA), o qual também é conhecido como discriminantes lineares de Fisher, trata-se de um método que utiliza informações das categorias associadas a cada padrão para extrair as características mais discriminantes. Em LDA, a separação inter-classes é enfatizada através da substituição da matriz de covariância total do PCA por uma medida de separabilidade, o critério Fisher.

Assim, calcula-se as matrizes de espalhamento inter-classes e intra-classes com objetivo de discriminar os grupos de amostras.

Abaixo o calculo do de interclasse e intra classe.

$$S_b = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^t \quad (1)$$

$$S_w = \sum_{i=1}^c \sum_{x_k \in X_i} (x_i - \mu)(x_i - \mu)^t \quad (2)$$

Sendo  $\mu_i$  a média total,  $X_i$  e  $N_i$  o número de amostras em uma classe.

Para isso deve-se maximizar o critério de Fisher, o qual segue abaixo.

$$P_{LDA} = \frac{\det |S_b|}{\det |S_w|} \quad (3)$$

em que  $i$  é o conjunto de autovetores generalizados de  $S_b$  e  $S_w$  correspondentes aos maiores autovalores.

$$W_{op} = \operatorname{argmax}_w \frac{\det |W^T S_b W|}{\det |W^T S_w W|} = [w_1 w_2 \dots w_m] \quad (4)$$

Acima podemos ver a interpretação do procedimento, a seguir iremos detalhar a interpretação probabilística do modelo. Na visão probabilística o LDA usa a regra de Bayes para determinar a probabilidade posterior de que uma observação pertence à uma classe. Devido à suposição normal de LDA, o posterior é definido por um gaussiano multivariado cuja matriz de covariância é considerada idêntica para todas as classes. Novos pontos são classificados computando a função discriminante  $\gamma_k$  (o numerador da probabilidade posterior) e retornando a classe com máximo  $\gamma$ . As variáveis discriminantes podem ser obtidas através de decomposições próprias da variância dentro da classe e entre classes.

Sendo a regra de Bayes a probabilidade posterior :

$$P(y = k/X) = \frac{P(X/y = k) * P(y = k)}{P(X)}$$

Mais especificamente,  $P(X/y=k)$  é modelado como uma distribuição gaussiana multivariada. Para usar esse modelo como classificador, precisamos apenas estimar, a partir dos dados de treinamento  $P(y=k)$ , a média das classes  $\mu_i$ , por médias empíricas do conjunto de amostras e pelas matrizes de covariância.

No caso da LDA, supõe-se que os gaussianos de cada classe compartilhem a mesma matriz de covariância. Isso leva a superfícies de decisão lineares

Após isso, foi utilizado o modelo de classificação não linear, com a análise de discriminates quadrados (QDA). Tanto o LDA quanto o QDA são derivados de modelos probabilísticos que podem ser obtidos utilizando a regra de Bayes. Esses modelos probabilísticos nos mostram a distribuição condicional de classe dos dados para cada classes  $k$ .

A análise discriminante quadrática (QDA) está intimamente relacionada à análise discriminante linear (LDA), onde se assume que as medidas de cada classe são normalmente distribuídas . Porém, diferentemente da LDA, na QDA, não há suposição de que a covariância de cada uma das classes seja idêntica.

Na prática ao tentar se classificar as observações, o LDA fornece limites de decisão lineares baseados na suposição de que as observações variam consistentemente em todas as classes. No entanto, muitas vezes ao analisar os dados, torna-se aparente que a variabilidade das observações dentro de cada classe é diferente. E portanto, deve-se utilizar o QDA, pois o esse método é capaz de capturar as diferentes covariâncias e fornecer limites de decisão de classificação não lineares mais precisos.

### C. K-Nearest Neighbors

Após isso, foi testada a previsão utilizado o KNN (K-Nearest Neighbors), o qual é um classificador, utilizado na aprendizagem supervisionada, onde o aprendizado é baseado no quão similar é um dado do outro. É uma técnica não paramétrica, o que significa que não faz suposições sobre a distribuição de dados subjacente, ou seja, a estrutura do modelo é determinada a partir dos dados. Além disso é classificado como um algoritmo lento, o que isso significa que ele não usa os pontos de dados de treinamento para fazer qualquer generalização . Em outras palavras, não há fase de treinamento explícita. A falta de generalização faz com que todos (ou a maioria) dos dados de treinamento são necessários durante a fase de teste. O algoritmo KNN baseia-se na semelhança de recursos e pode ser usado tanto para regressão quanto para a classificação.

As etapas de um algoritmo KNN são: receber um dado não classificado, especificar um número inteiro positivo  $k$ , selecionar as  $k$  entradas no banco de dados mais próximas da nova amostra, através da medida da distância Euclidiana entre os dados, após isso é encontrada a classificação mais comum dessas entradas e finalmente é essa a classificação que atribuímos à nova amostra. Um dos pontos cruciais para que o algoritmo funcione de forma ótima é a definição correta do número  $k$ , assim, foi plotado um gráfico para que fosse observado a variação das previsão em função da mudança desse parâmetro, para que o valor fosse escolhido aprimorasse a estimativa.

### D. Rede Neural

Logo, posteriormente outra técnica foi testada, foram feitas previsões com base em um Rede Neural Artificial, executada a partir da função MLPClassifier().

Redes Neurais Artificiais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos vivos inteligentes e que adquirem conhecimento através da experiência. Uma rede neural artificial é composta por várias unidades de processamento, cujo funcionamento é bastante simples. Os sinais da entrada no neurônio são representados por um vetor, cada sinal é multiplicado por um número, ou peso, que indica a sua influência na saída da unidade. É feita a soma ponderada dos sinais que produz um nível de atividade. Após isso, é adicionado um grau de liberdade a mais, que não é afetado pela entrada nessa expressão, correspondendo tipicamente ao “bias” (viés).

$$z = \sum_{i=1}^N X_i w_i + b \quad (5)$$

Essa equação passa então por uma função matemática de ativação, com a característica de ser não linear, responsável por limitar tal valor a um certo intervalo, além de cada função ter características que devem ser levadas em consideração depende da saída que se deseja obter. Se este nível de atividade exceder um certo limite (threshold) a unidade produz uma determinada resposta de saída.

Existe uma rede neural denominada Perceptron Multicamadas (MLP), onde o vetor de entrada passa pela camada inicial e após isso vai para as camadas ocultas, os valores da camada inicial são ligados às entradas da camada seguinte e assim até chegar a camada de saída que fornece o resultado da rede.

O treinamento de uma rede MLP tem como classificação o aprendizado de máquina supervisionado, em que cada amostra de dados utilizada apresenta um rótulo que nos diz a qual classificação ela pertence. Assim, esse tipo de rede tem por finalidade aprender padrões referentes a cada tipo de classe e quando uma nova amostra não classificada for analisada, a rede preverá a qual classe o dado pertence. O treinamento é feito através do processo de otimização no qual minimizamos uma função custo, a qual pode ser pensada como uma função que mede a diferença entre o que a rede neural prevê e o que de fato foi observado.

O algoritmo mais utilizado para o aprendizado de tais redes é o backpropagation. No backpropagation tendo como base o erro encontrado na camada de saída, os pesos da rede neural são recalculados começando da última camada, de trás para frente. Desse modo, todos os pesos são atualizados realizando a retropropagação do erro obtido pela rede. Funciona da seguinte maneira: os pesos são inicializados com valores pequenos e aleatórios para fornecer à rede dados de entrada e calcular a função de erro, para tentar minimizar o valor da função de erro, pois esses pesos estarão associados a uma região elevada na superfície de custo. Logo após, calcula-se os valores do gradiente para avaliar o vetor gradiente que fornece a direção de maior crescimento da função, como queremos o de maior decréscimo da função de erro, tomamos o sentido contrário do gradiente. Após, isso o peso é atualizado de forma iterativa, até chegar a um limiar estabelecido ou o número de iterações atingir um valor máximo.

No trabalho foram utilizadas 3 camadas, a camada de entrada, uma camada oculta com 100 neurônios e a camada de saída. Ademais, foi usado o otimizador Adam, excelente algoritmo de primeira ordem para otimização baseada em gradiente de funções objetivas estocásticas.

Além disso, foi utilizada a função de ativação ReLU. Caracterizada por  $f(x) = \max(0, x)$ , tendo sua derivada igual a zero para valores negativos e igual a 1 para valores positivos. Redes com a função ReLU são fáceis de otimizar, já que a ReLU é extremamente parecida com a função identidade. A única diferença é que a ReLU produz zero em metade do seu domínio. Como consequência, as derivadas se mantêm

grandes enquanto a unidade estiver ativa. ReLU é a função de ativação mais amplamente utilizada ao projetar redes neurais atualmente. Primeiramente, a função ReLU é não linear, o que significa que podemos facilmente copiar os erros para trás e ter várias camadas de neurônios ativados pela função ReLU.

### III. RESULTADOS

Os códigos comentados, os gráficos e tabelas geradas encontram-se disponíveis no repositório indicado em [4].

#### A. Pré-processamento

Após realizar o Pré-processamento o conjunto de dados foi bastante reduzido ao se retirar os possíveis outliers, pois a forma como as medições foram feitas não é claro, impossibilitando a obtenção de informação a partir dessas medidas. Também foram retiradas as amostras de tecnologias diferentes do UMTS (3G), visto que cada tecnologia utiliza ferramentas distintas que devem ser analisadas de forma separada. Tudo isso foi feito com o intuito de preparar melhor os dados para que o modelo de classificação possa ser mais preciso.

Além disso, foram separados os conjuntos de teste e treino a uma taxa de 9/10, ou seja, 90% dos dados foram utilizados para treinar o modelo e 10% para validar.

Como os preditores não possuem correlações consideráveis entre si todos foram usados para treinar os modelos a seguir.

#### B. Modelo de Classificação Linear

O modelo de classificação linear utilizado foi o LDA, devido a quantidade de classes.

Nesse caso, o LDA conseguiu classificar corretamente 58.52% do conjunto de teste e sua confusion matrix pode ser vista na figura 1. Foi realizada uma normalização devido a grande quantidade de amostras, tornando mais fácil observar os acertos e erros.

A confusion matrix é uma forma gráfica de mostrar como o modelo classificou as amostras. As colunas representam as previsões do modelo e as linhas a classificação correta, ou seja, na diagonal principal da matriz estão indicadas as previsões corretas e fora dela são classificações errôneas.

A partir da figura 1 é possível observar que esse modelo da uma grande preferência aos MOS 1 e 2. Nesses casos ele consegue prever com uma grande precisão, porém nos outros valores ele possui grande dificuldade nas previsões, acertando apenas 2% das amostras com MOS 3, 3% das amostras com MOS 4 e 7% das amostras com MOS 5.

Isso mostra uma grande limitação desse modelo, pois é notório uma grande tendência de previsão.

#### C. Modelos de Classificação não Linear

1) QDA: Nesse caso foi utilizado o QDA para tentar obter uma melhor precisão do modelo, já que no caso o conjunto de dados é bem grande, estimar a variância de cada classe pode trazer melhores resultados, tentando reduzir a tendência do LDA observada anteriormente.

O QDA por sua vez foi capaz de classificar corretamente 59.23% do conjunto de teste e sua confusion matrix pode ser

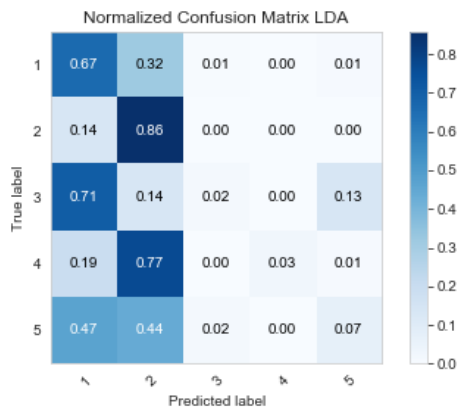


Figura 1. Confusion Matrix do LDA

vista na figura 2. Novamente a matriz foi normalizada pelo mesmo motivo.

Como é possível observar pelos valores de acurácia, o QDA aparentemente não foi muito diferente do LDA, porém ao observar a confusion matrix da figura 2 é possível observar que nos casos de MOS 4 e 5 conseguiu ter uma precisão consideravelmente maior, indicando uma redução da tendência observada no modelo anterior.

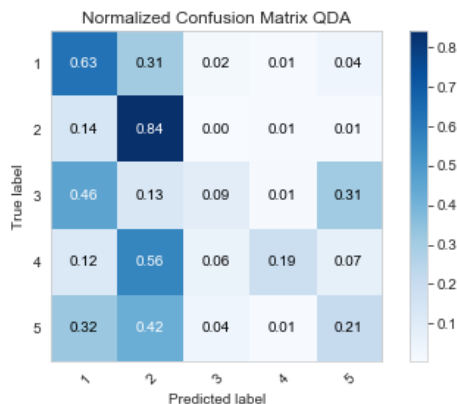


Figura 2. Confusion Matrix do QDA

2) *KNN*: Nesse caso, foi utilizado o KNN para tentar melhorar ainda mais o resultado obtido pelo QDA. Esse método é utilizado com o objetivo de evitar a necessidade de estimar diretamente as densidades de probabilidades, embora se tenha muitas amostras para fazer essa estimativa o objetivo é evitar fazê-la devido ao desconhecimento do processo de obtenção dessas amostras como já foi comentado anteriormente. Embora já tenha sido feito um tratamento no conjunto ainda não é possível saber se as amostras não descartadas estão corretas.

Devido a isso esse método foi utilizado já que ele estima as densidades de uma forma indireta.

Para estimar o valor de K, alguns valores foram testados para treinar o modelo e em seguida foram testados e suas precisões podem ser observadas no gráfico 3.

O K que melhor classificou os dados foi igual a 55 e conseguiu acertar 60.90% das amostras e sua confusion matrix pode ser vista na figura 4.

Novamente, pode-se observar um melhor desempenho comparado ao QDA e o LDA e as amostras com MOS 4 e 5 foram ainda mais precisos que no caso anterior.

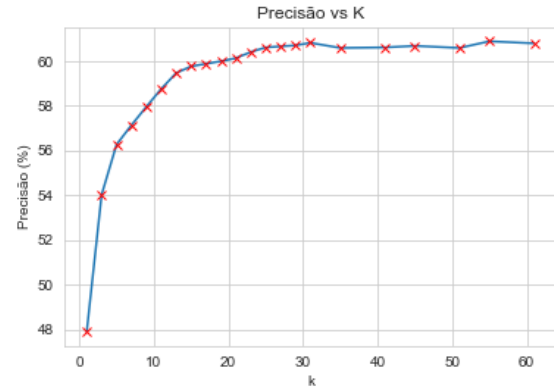


Figura 3. Precisão do KNN para alguns valores de K

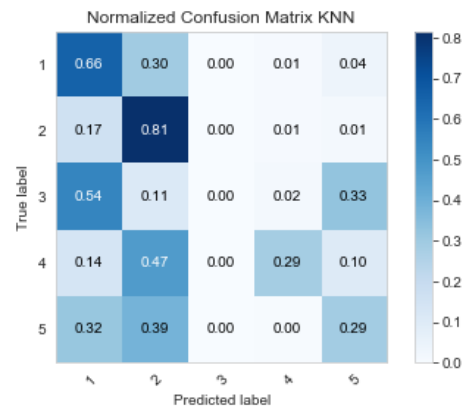


Figura 4. Confusion Matrix do KNN para  $K = 55$

3) *Rede Neural*: Nesse caso, foi utilizada uma rede neural com 3 camadas, ou seja, apenas uma hidden-layer com 100 neurônios e função de ativação ReLU. Novamente, deseja-se obter um modelo ainda melhor de classificação e as redes neurais são uma ótima escolha para isso devido a sua grande capacidade de adaptação, alterando a quantidade de camadas, utilizando funções de ativações não lineares distintas para tentar capturar melhor as variações das amostras.

A rede neural foi capaz de classificar corretamente 61.26% das amostras, mostrando o melhor resultado nesse critério. Sua confusion matrix pode ser vista na figura 5.

Embora ela tenha tido uma precisão maior, ainda é possível ver que sua confusion matrix é bem similar ao caso do KNN, porém demonstra uma maior dificuldade em prever amostras com MOS 3.

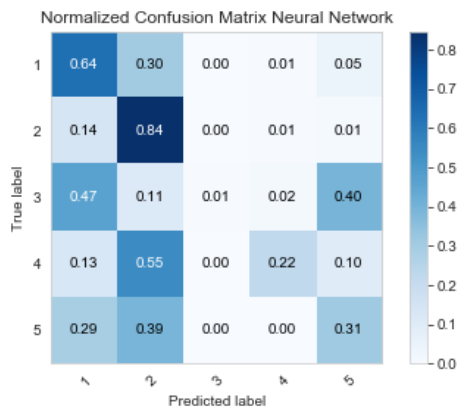


Figura 5. Confusion Matrix da rede neural

#### IV. CONCLUSÃO

Neste trabalho foram vistas diversas formas de classificação, uma linear e três não lineares e a comparação da precisão de cada uma pode ser visto no gráfico 6.

Como ficou evidente na seção anterior, todos os modelos tiveram uma grande tendência a classificar as amostras com MOS 1 e 2, principalmente o linear, e dificuldade nos demais. Isso pode ter ocorrido devido à distribuição das amostras que estão concentradas nesses valores de MOS.

Além disso, embora os modelos não lineares conseguiram classificar melhor esses outros MOS, ainda ficou muito aquém do esperado para ferramentas tão poderosas como a rede neural que atualmente está sendo amplamente utilizada devido sua grande capacidade adaptativa.

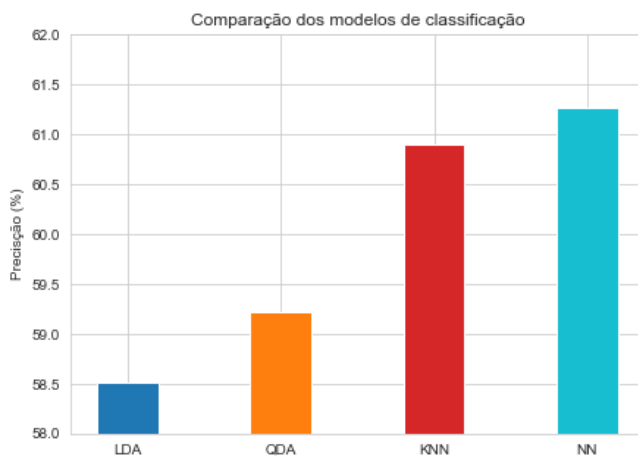


Figura 6. Comparação entre os modelos testados

#### REFERÊNCIAS

- [1] D. Soldani and A. Manzalini, "Horizon 2020 and beyond: On the 5G operating system for a true digital society," *IEEE Vehicular Technology Magazine*, vol. 10, no. 1, pp. 32–42, Mar. 2015.

- [2] *ITU-T Rec. P.10/G.100 (11/2017), Vocabulary for performance, quality of service and quality of experience.*
- [3] V. Luciani. (2018, jan) Call tests measurements for mos prediction. [Online]. Available: <https://www.kaggle.com/valeriol93/predict-qoe/>
- [4] E. M. S. de Santana Júnior. (2019, 08) ica2019.2. [Online]. Available: <https://github.com/ezequiasjunior/ica2019.2>