


Aprendizagem Automática

– Projeto Final –

Letícia Lucas A48584 Daniela Soares A48590

– Introdução –

Este trabalho tem por objetivo lidar com textos de críticas de cinema do IMDb. O objetivo é tratar um problema de **classificação** (método supervisionado) treinando uma série de classificadores que devem ser capazes de classificar as críticas de **duas formas distintas**.

BINÁRIA

A

Cada crítica é classificada como **positiva** ou **negativa**

B

MULTI-CLASSE

Cada crítica é classificada no intervalo **1-4** ou **7-10** .

A presente apresentação procura resumir o processo e a metodologia seguidos, relembrando e resumindo os pontos mais importantes já descritos no relatório do trabalho entregue sob a forma de **Jupyter Notebook**.

Enquadramento – Problema

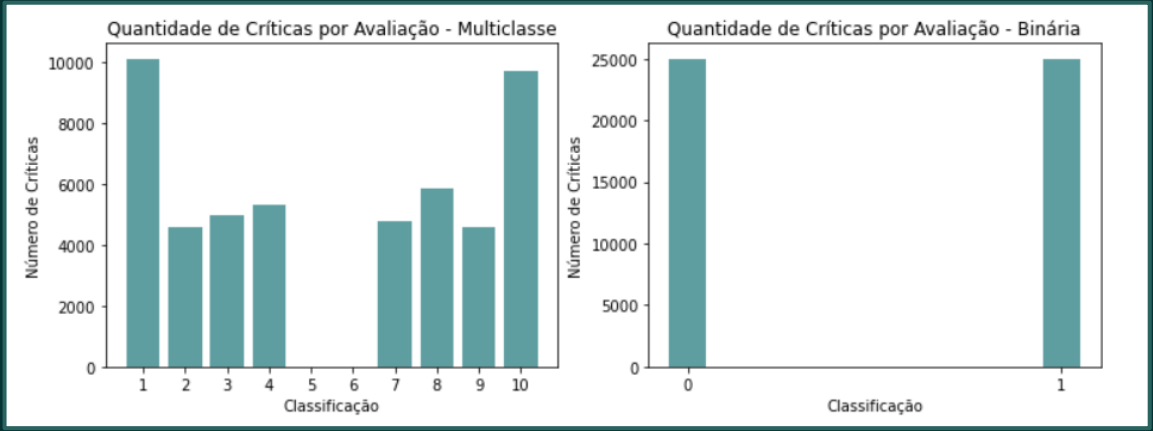
Que tipo de problema de trata?

Um problema de **classificação** refere aquele em que a variável de output é uma categoria e não um valor real, como na regressão. É um método **supervisionado**.

... e o que significa ser um método supervisionado?

É fornecida à máquina uma série de informações já divididas por classe, sendo depois fornecido um novo conjunto de dados a classificar da mesma forma.

Enquadramento – Dados

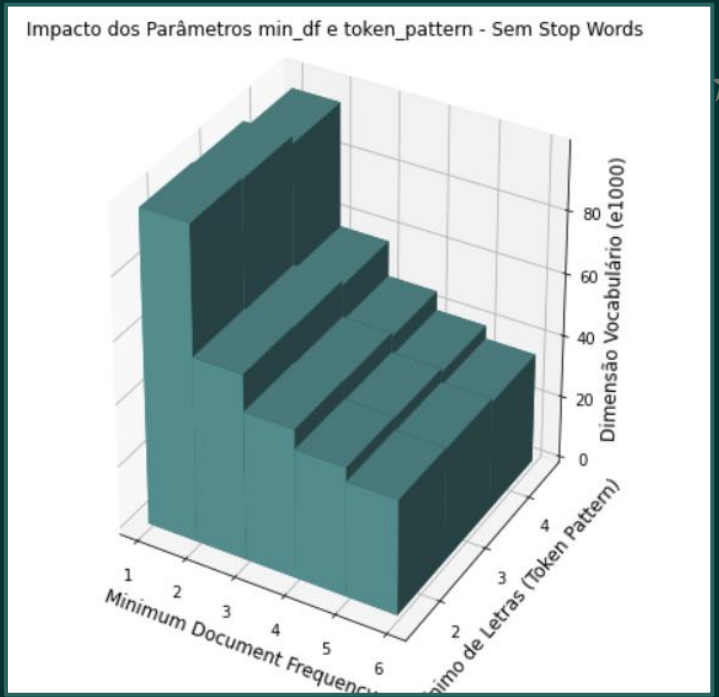


1. Existe uma divisão **equilibrada** de críticas positivas e negativas.
2. Existe uma maior incidência em críticas extremas (avaliação 1 e 10). De um ponto de vista de classificação, isto pode ser tanto benéfico como maléfico no processo – **pode causar maior incidência nessas duas classes.**

Enquadramento – TF IDF

O ***corpus*** (documentos) será transformado via **TFIDF Vectorizer** – uma forma que permite medir e quantificar a importância de dada palavra no mesmo.
TF (Term Frequency) trata a frequência de cada termo em estudo com cada documento do seu geral. Já **IDF (Inverse Document Frequency)** tem em conta o quão comum uma palavra é em todo o corpus.

Um dos objetivos do trabalho passa pela procura dos melhores valores para **min_df** (frequência mínima), **token_pattern** (estipula um "padrão" de palavra) e ainda gama de **n gramas** (tamanhos de janela de visualização de palavras).



Implementação – Metodologia TF-IDF

A procura inicial desses melhores parâmetros baseou-se em aliar vetorizadores com diferentes inicializações à classificação por **Logistic Regression**, igualmente inicializada com diferentes pares penalty/regularização, procurando os pares que obtinham melhores resultados de...

Cross Val.

O melhor par é o que tem **melhor capacidade de generalização.**

Nº de Erros

O melhor par é o que tem **menos erros** nos **dados de validação**

Precisão

O melhor par é o que tem melhor **percentagem de True Positives**

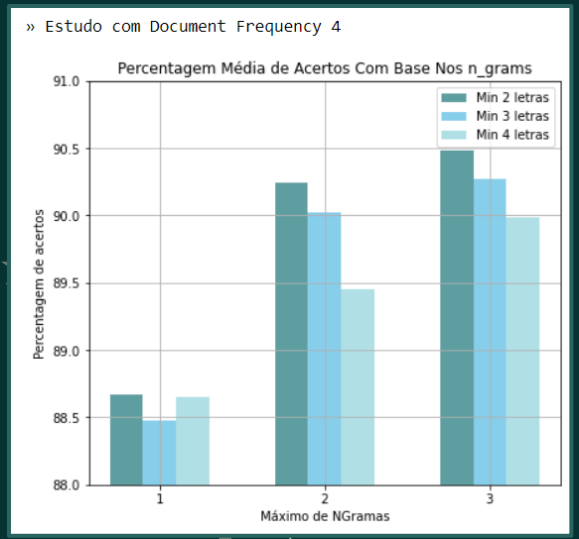
Score

O melhor par consegue **melhor *score*** nos dados de validação

Guardam-se as **médias dos *Scores* de classificação do conjunto de validação** obtidos por cada par, por cada combinação de parâmetros do vetorizador, a fim de os **comparar**. Para além de bons resultados de *Score* procura-se ainda manter um vocabulário não demasiado grande, para minimizar problemas de *performance*.

Implementação – Escolha TF IDF

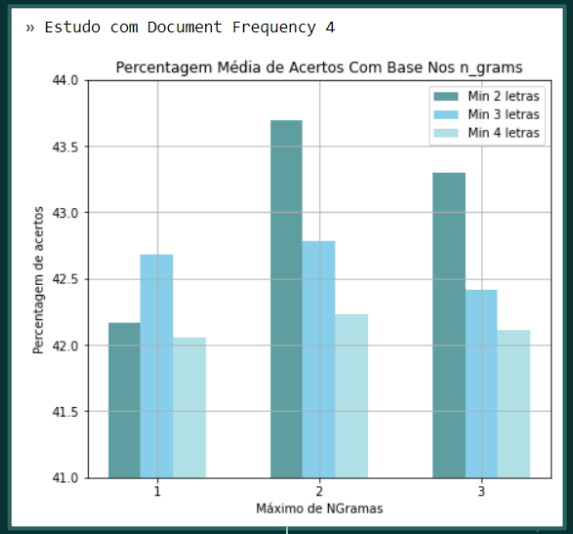
Binário



Os dois casos resultam nos mesmos melhores parâmetros:
min_df: 4
pattern: min. 2 letras
n-gramas: 1 e 2

(Ver até **3 n gramas** dá os melhores resultados, **mas vocabulário demasiado grande**)

Multi-Classe



Implementação – Classificadores

Logistic Regression — Classificador de implementação **obrigatória**

Linear SVC — Testes às capacidades de **classificação vetorial**

K-Neighbors — Um teste a classificadores "**preguiçosos**"

Naive Bayes — **Compensação** pelos resultados anteriores

Sobre Logistic Regression

É um **método de classificação baseado em observações anteriores de um conjunto de dados.** Prevê com base na análise da relação entre uma ou mais variáveis independentes.

Bons resultados quando o conjunto de dados é **linearmente separável**, menos propenso a sobre aprendizagem. ✓

Fácil de **implementar** e **interpretar** e muito eficiente no treino. ✓

No mundo real **os dados raramente são linearmente separáveis**. É também limitado a problemas cujo conjunto de dados é composto por números discretos. ✗

Sobre Linear SVC

Um classificador **SVM** é um tipo de algoritmo usado na análise e classificação de dados. Pode ser usado tanto na regressão como na classificação. **Funciona com base na procura do hiperplano que maximiza a margem entre duas classes**. É uma ferramenta poderosa e muitas vezes usada em reconhecimento de dígitos, expressões faciais e **classificação de texto**.

Tem bons resultados em problemas de classificação, entre eles **problemas com dados de alta dimensão**.

Muito **sensível** ao escalamento dos dados.

Sobre K-Neighbours Classifier

Classifica com base na proximidade de um ponto aos seus vizinhos, **pressupondo que pontos semelhantes se encontram perto uns dos outros**. É considerado "**preguiçoso**" pois não realiza, realmente, nenhum treino.

Fácil implementação e adaptabilidade – à medida que os dados de treino vão sendo adicionados, o algoritmo ajusta-se para considerar quaisquer novos dados. ✓

Não funciona bem com **elevados conjuntos de dados**, uma vez que requer um **alto armazenamento** de memória, nem com dados de dimensões elevadas. ✗

Necessário determinar o valor de K de acordo com o problema. Para além disso, o processo de estimação é **lento** se o número de vizinhos for elevado. ✗

Sobre Naive Bayes

Baseado no **teorema de Bayes** (que calcula a probabilidade de um evento ocorrer com base no conhecimento prévio das condições relacionadas a um evento), **prevê a "etiqueta" de um texto**.
O **Multinomial Naive Bayes** calcula a probabilidade de cada "etiqueta" para um determinado conjunto de dados e, em seguida, fornece a "etiqueta" com a maior probabilidade como saída.

Fácil de implementar, é apenas necessário calcular a probabilidade, **estima em tempo real.** É ainda escalável, **podendo lidar com elevados conjuntos de dados**.

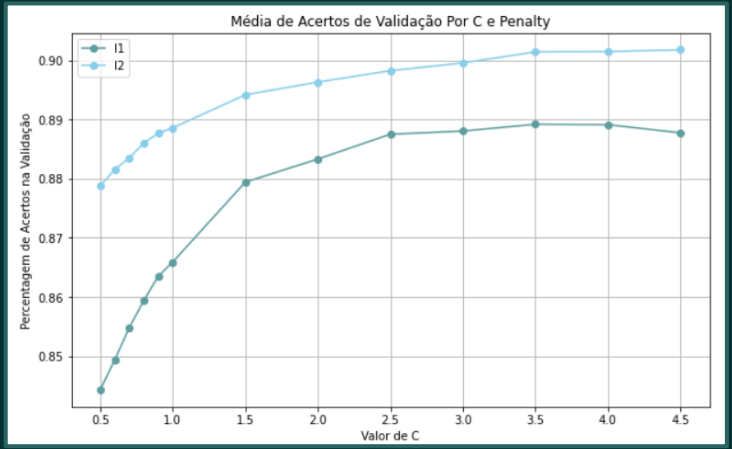
Precisão da previsão ser **menor comparativamente a outros algoritmos de probabilidade**.

Implementação – O que Calibrar?

Logistic Regression	Linear SVC	K-Neighbours	Naive Bayes
É necessário encontrar o melhor valor de **Penalty** e **Regularização**	É necessário encontrar o melhor valor de **Penalty** e **Regularização**	É necessário encontrar o melhor número de **Neighbors** e **Weight**	Basta encontrar o valor de **alpha** (parâmetro de suavização)
• O melhor par tem o melhor **Score** de validação	• O melhor par tem o melhor **Cross Val. Score**	• O melhor par tem o melhor **Score** de validação	• O melhor tem o melhor **Score** de validação

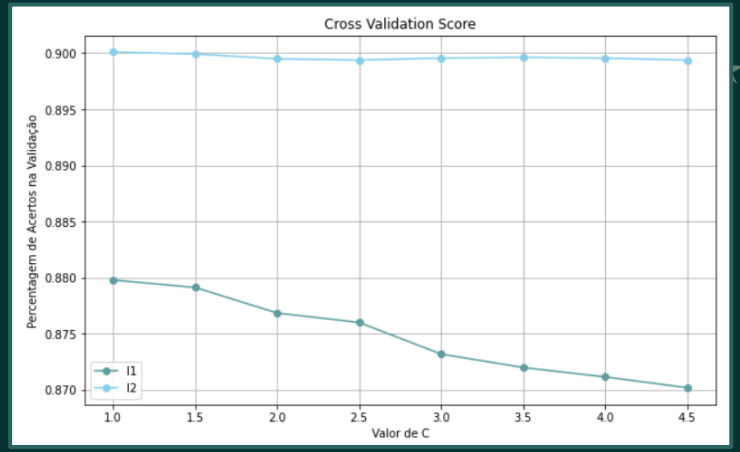
Implementação – Parâmetros Binários

Logistic Regression



- **Penalty:** l2 é mais benéfico (embora fique com mais vocabulário)
- **C:** Entre 3.5 e 4.5 há pouca diferença – opta-se pelo menor valor: 3.5.

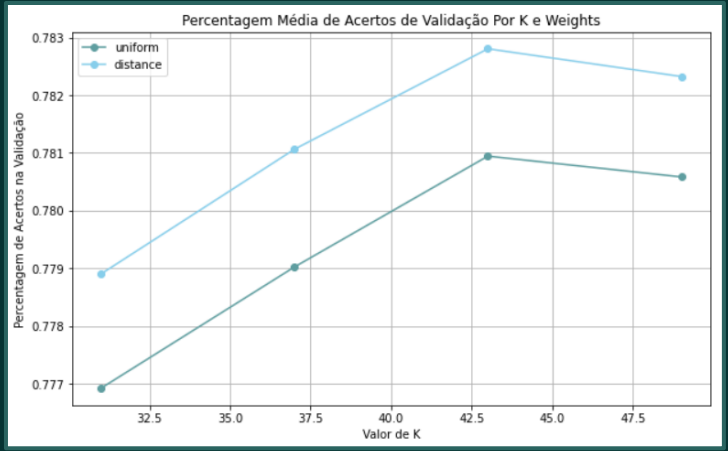
Linear SVC



- **Penalty:** l2 é mais benéfico.
- **C:** Escolhe-se C = 1, onde se vê início de convergência.

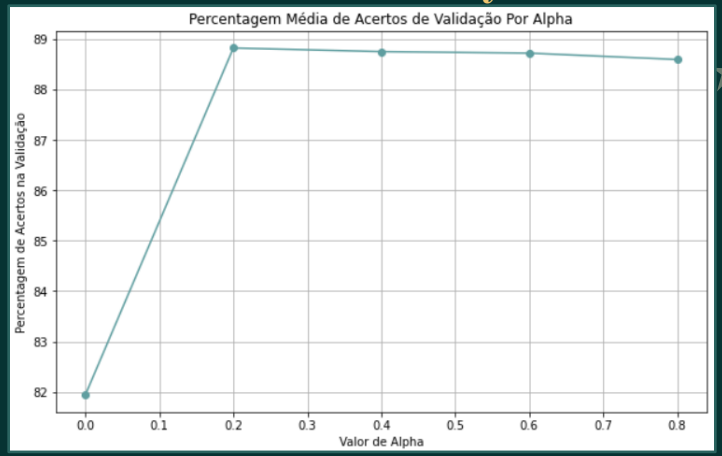
Implementação – Parâmetros Binários

K-Neighbours Classifier



- **Weight:** Distance, onde os pontos mais próximos têm mais peso.
- **K:** 43, um número que já se considera demasiado elevado.

Multinomial Naive Bayes



- **Alpha**: O melhor ponto fica a 0.2

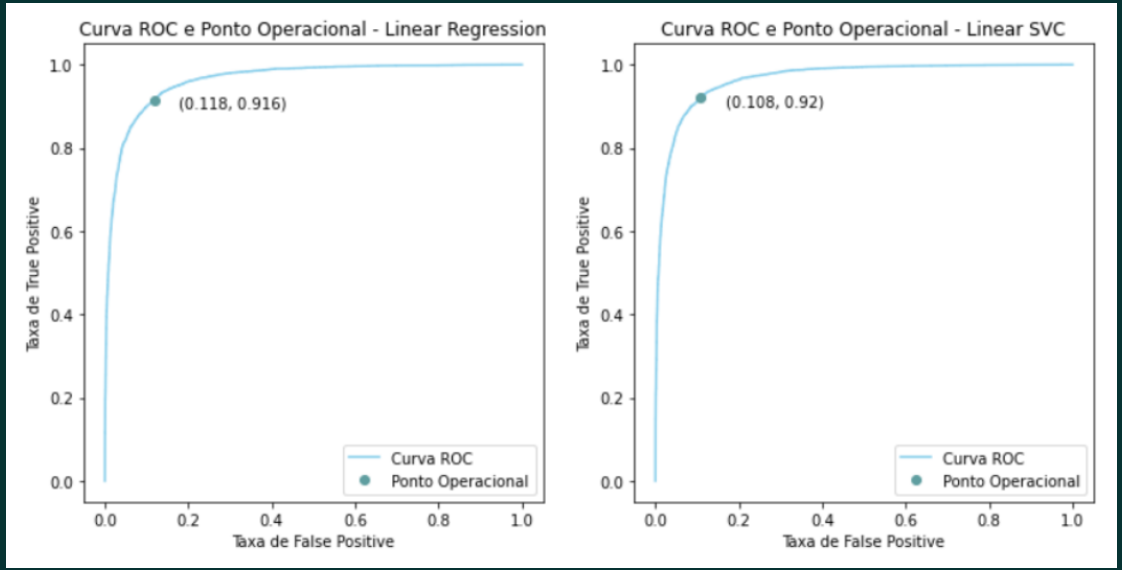
Comparação – Caso Binário

Metodologia

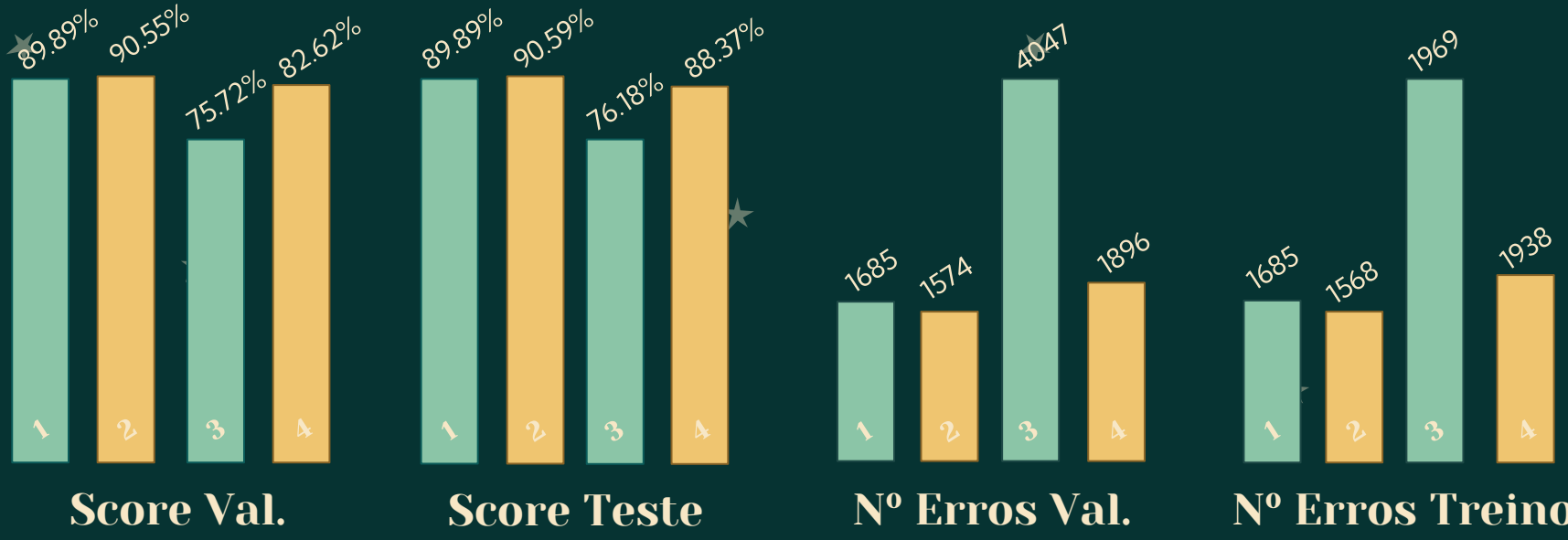
Dividem-se os dados em **Treino, Validação e Teste.** Vetoriza-se e classifica-se com os melhores parâmetros.

Análise

Compara-se o **Score de Teste e Validação** e o seu número de erros. Veem-se, ainda, as curvas ROC dos dois primeiros classificadores.



Comparação – Caso Binário



1 – Logistic Regression 2 – Linear SVC 3 – K-Neighbour Classifier 4 – Naïve Bayes

Comparação – Caso Binário

K-Neighbours

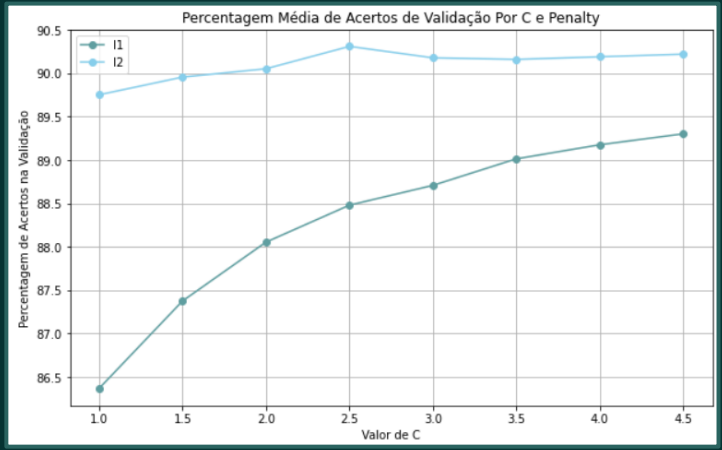
Sobressai, pela negativa, **o desempenho do classificador K-Neighbors**. Lembra-se que é um método "**preguiçoso**" que na verdade não aprende - apenas mantém dados em memória para comparação. Isto reflete-se nos resultados obtidos, com mais do dobro dos erros dos restantes classificadores.

Comparação Restante

Os restantes classificadores são bastante comparáveis - aplicando métodos mais robustos, conseguem garantir todos cerca de 90% de acertos. Na necessidade de distinção, o **Linear SVC** consegue sair com resultados muito ligeiramente melhores, embora esta diferença seja mínima, principalmente em comparação ao **Logistic Regression**.

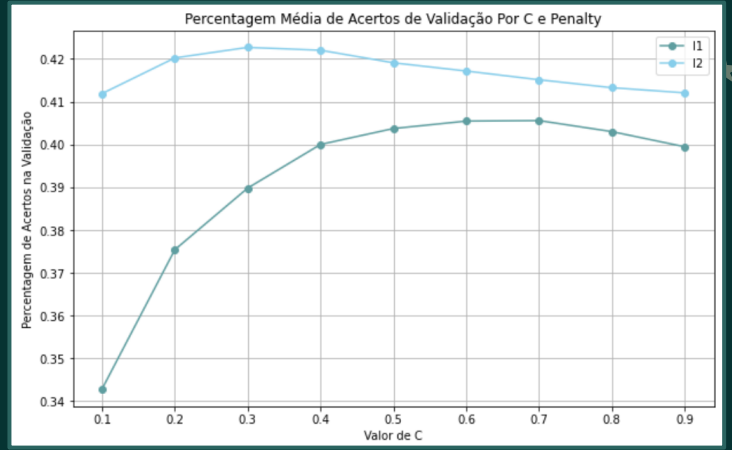
Implementação – Parâmetros Multi-Classe

Logistic Regression



- A escolha é bastante simples, notando-se um ponto de destaque **C = 2.5** com **Penalty l2**.

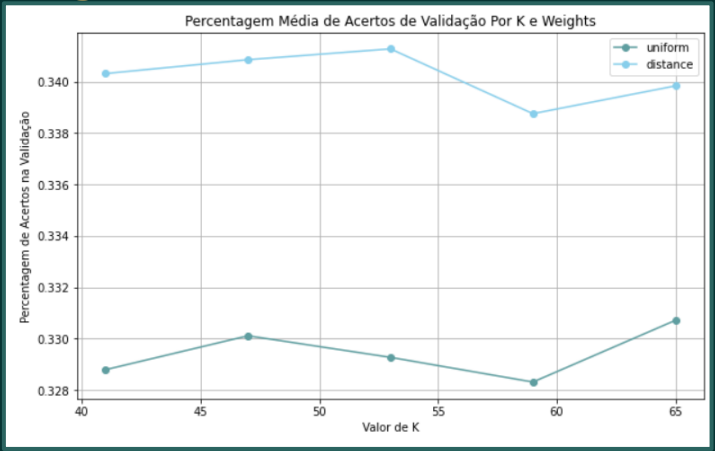
Linear SVC



- Escolhe-se o "pico" que parece aparecer na curva entre 0.1 e 0.5 – **C=0.3 com Penalty l2**.

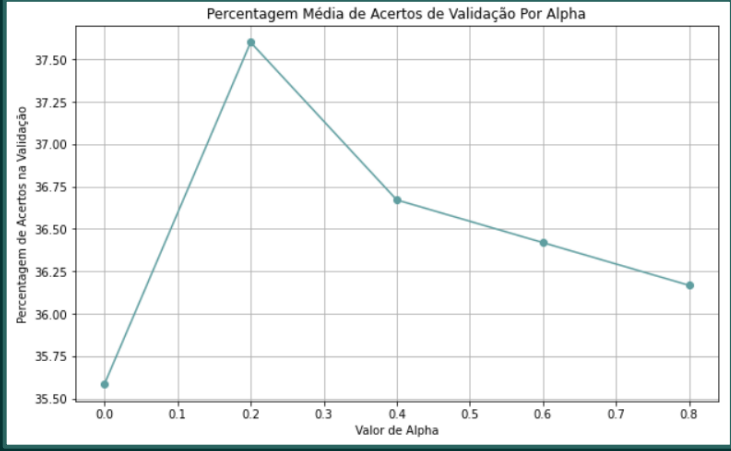
Implementação – Parâmetros Multi-Classe

K-Neighbours Classifier



- **Weight:** Distance
- **K:** 53, já é visível uma grande descida de capacidade face aos classificadores anteriores...

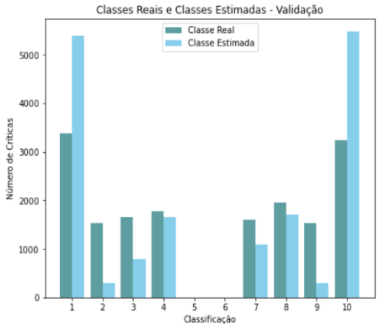
Multinomial Naive Bayes

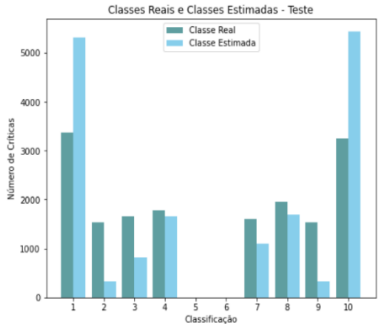


- **Alpha:** Não há dúvida de que o melhor é o valor 0.2, mas verifica-se que, também este classificador, tem agora dificuldade em acompanhar os dois primeiros...

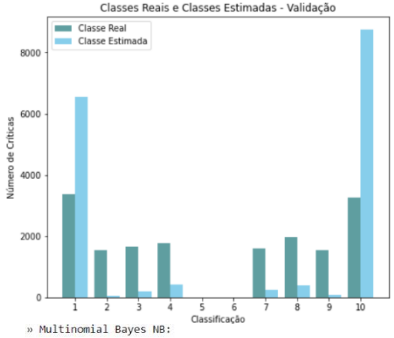
Comparação – Caso Multi-Classe

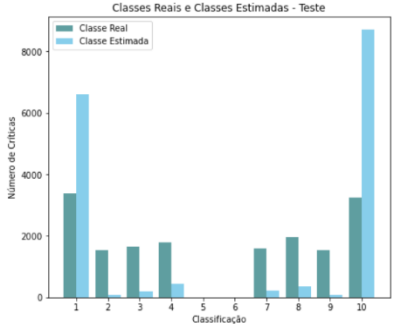
» Logistic Regression:



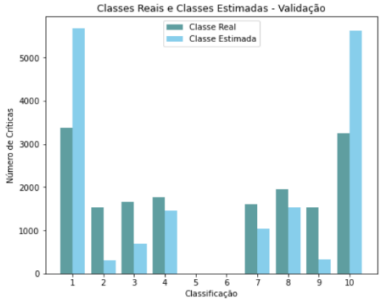


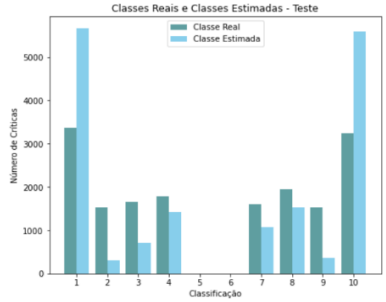
» k-Neighbors:



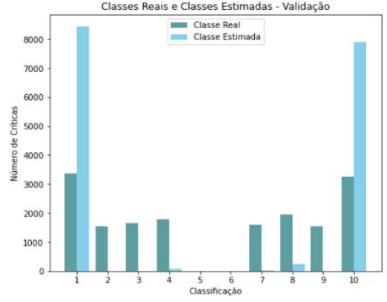


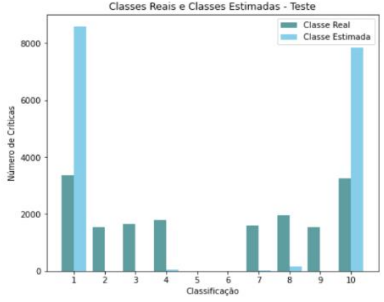
» Linear SVC:



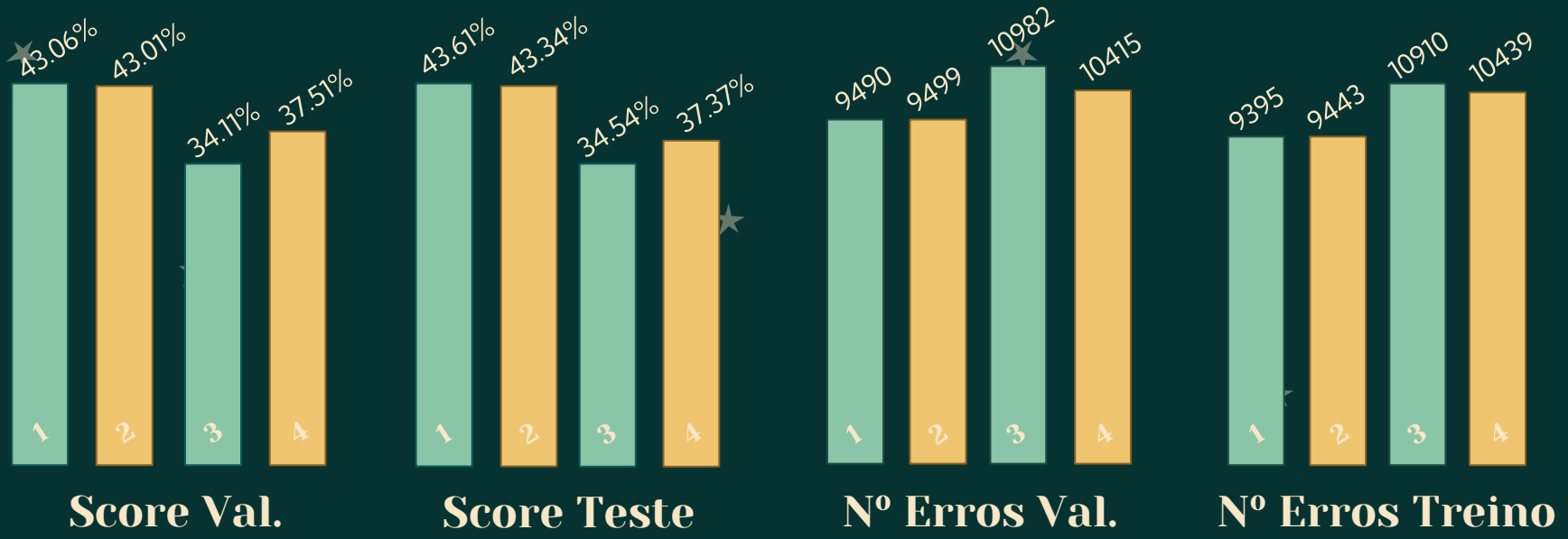


» Multinomial Bayes NB:





Comparação – Caso Multi-Classe



1 – Logistic Regression 2 – Linear SVC 3 – K-Neighbour Classifier 4 – Naïve Bayes

Comparação – Caso Multi-Classe

Porquê os maus resultados? (<45%)

Enquanto seres humanos, é fácil pensar "Quem escreveu esta crítica gostou do filme!", mas pode haver mais dificuldade em dizer "Quem escreveu esta crítica deu exatamente 7 valores!". O problema de **análise de sentimentos** assenta no facto de que duas pessoas que escrevam a mesma crítica podem, mesmo assim, dar classificações completamente diferentes – **depende da forma de pensar e sentir individual de cada uma, algo que não pode ser replicado com algoritmos deste tipo**.

Comparação Geral

Logistic Regression e Linear SVC destacam-se positivamente, exibindo resultados muito semelhantes um ao outro. O classificador **KNN** mantém-se como sendo o pior de todos os implementados e, neste caso, o **Multinomial Bayes** já não consegue exibir resultados tão benéficos como na classificação binária. Mais uma vez, na necessidade de escolha do melhor classificador, seleciona-se, por uma margem de centésimas em termos de percentagem de acertos, o **Logistic Regression**.

O ranking é refletivo daqueles implementados neste trabalho, não é, necessariamente, geral!

Comparação – Ranking

De um modo geral, obtiveram-se resultados bastante satisfatórios – **a análise binária é bastante mais simplificada, cada palavra poderá apenas ter uma conotação mais positiva ou negativa**, o que justifica os elevados resultados.

Caso Binário

1. Linear SVC
2. Logistic Regression
3. Naive Bayes
4. K-Neighbours

Caso Multi-Classe

1. Logistic Regression
2. Linear SVC
3. Naive Bayes
4. K-Neighbours

Como foi explicado em relação à análise de sentimentos, a classificação multi-classe demostra mais problemas – para piorar, **devido à elevada quantidade de críticas extremas, também as classificações têm mais tendência em aí cair**.

Estudo Extra – PCA

De que trata?

O **PCA** tem como objetivo **projetar dados nas direções de maior variância (as componentes principais), sendo os dados projetados descorrelacionados (a matriz de covariância é diagonal)**. Pressupõe-se, assim, que direções onde os dados variam mais contêm mais informação.

Enquadramento no Trabalho

Testa-se o ***Score*** obtido via **Logistic Regression** após tratamento dos dados com diferente número de componentes. Com o aumento do número de componentes, também os resultados melhoravam, mas, por outro lado, também aumentava a carga computacional. No desenvolvimento deste trabalho, muitos processos requeriam algum tempo de processamento, mas incomparáveis ao tempo que aqui foi necessário.

A melhor gama dentro do que pôde ser visto aparentou ser [3000-4000], mas, devido ao peso computacional, apenas se poderão usar 500 componentes principais nas comparações.

Estudo Extra – Com PCA vs Sem PCA

(500 componentes + Logistic Regression)

Estudo Extra – PCA

O que é que aconteceu?

Com 500 componentes – **um valor elevado e que já implica algum tempo a executar** – ainda não se consegue atingir os resultados obtidos sem **PCA**. Porquê?

O processo **PCA** é baseado na extração dos eixos nos quais os dados expressam mais variabilidade. Embora "espalhe" melhor os dados e possa ser uma boa ajuda em problemas de regressão, **não há garantia que os novos eixos sejam consistentes com as *features* discriminatórias num problema de classificação**.

Em suma, transformações **PCA** permitem **sumarizar a informação de um número elevado de *features*** para um número mais limitado de componentes - mas sendo as componentes principais, muitas vezes, difíceis de interpretar (ou não as mais intuitivas), levam a piores resultados nas *performances* de classificação.

A única forma de melhorar resultados seria aumentar o número de componentes, mas estar-se-ia a colocar em causa o tempo de execução – torna-se demasiado pesado.

Estudo Extra – *Clustering*

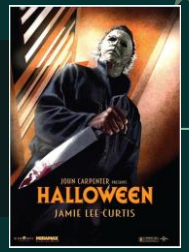
De que trata?

Os métodos de agrupamento (***Clusetring***) são técnicas de **aprendizagem não supervisionada** onde se procura encontrar representações mais compactas dos dados sem mais nenhuma informação sobre os mesmos. O objetivo é dividir os dados em grupos, de modo a que elementos do mesmo grupo sejam o mais semelhantes entre si possível. Desta forma, pode-se categorizar os dados e descobrir as "classes" que neles estão subjacentes.

Enquadramento no Trabalho – Quantos *Clusters*?

Um método popular de escolha de melhor número é o chamado "método do cotovelo" - o método consiste em traçar o SSE em função do número de *clusters* e escolher "o cotovelo" da curva como o número de *clusters* a utilizar. O *Sum Squared Error* refere uma forma de avaliação da qualidade dos *clusters* criados. No entanto, este método não é infalível - muitas vezes, com elevadas quantidades de dados desorganizados, pode dificultar a convergência, impossibilitando a escolha de um cotovelo exato. Tal aconteceu no conjunto de dados em mão. Ainda assim, fazendo variar este valor, encontram-se *clusters* interessantes.

Estudo Extra – Temas nos *Clusters*



“film, chilling, halloween, spooky, mask, meyers, michael, horror, carpenter”

Referência a Michael **Myers** – uma personagem fictícia da série de filmes de **terror Halloween**. Aparece pela primeira vez em 1978, no filme realizado por John **Carpenter**. A personagem usa uma **máscara**.

“spoof, object, truck, bride, comedy, dolls, play, child, horror, chucky”

Referência aos filmes ***Child's Play***. Com o progresso dos filmes, estes começaram a tornar-se mais satíricos e exagerados, tendo-se tornado mais um misto de **comédia** e **horror** a partir do filme **Bride of Chucky**.

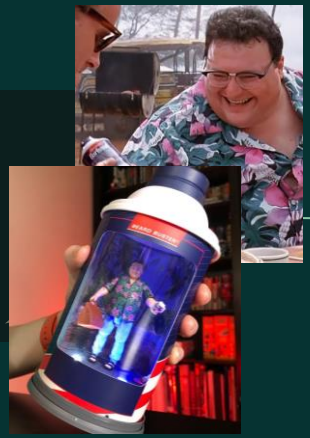
fbi, newspaper, freeman, prolific, saboteur, spies , hitchcock, kane

O filme "**Saboteur**", de **Hitchcock**, trata um trabalhador chamado Barry **Kane**. É uma marca deste realizador aparecer nos seus filmes. Neste caso, ele aparece num quiosque de **jornais**. As marcas policiais referem a carreira do mesmo.

“record, singer, cool, sequel , christina, travolta , gives , industry , performance”

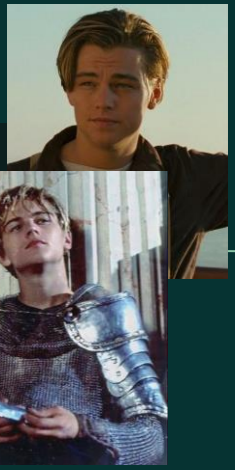
O filme **Be Cool** segue Chili Palmer (John **Travolta**) que, tendo feito o *debut* para produtor de filmes, decide focar-se na **indústria musical**. Este filme conta ainda com a participação e produção musical de **Christina** Millan.

Estudo Extra – Temas nos *Clusters*



“cream, dinosaurs, sharing, species, inconsistencies , goers , dino, movie, jurassic, short”

Referências variadas a filmes da época **jurássica**, referindo **dinossauros** e **espécies**. Nos filmes *Jurassic Park* existe um ponto na história onde uma lata modificada de **creme** de barbear é usada, podendo vir daí a inclusão.



“really , user, let, romeo, frozen, juliet, came, jack, hard, titanic”

Referência a **Titanic** e **Romeu e Julieta**. **Jack** e **Romeu**, protagonistas respetivos, são ambos representados por Leonardo diCaprio, daí surgir o agrupamento. **Frozen** refere *The Revenant* onde uma personagem interpretada pelo mesmo tem de sobreviver a um ambiente gélido.