

Trabalho Prático do Módulo 2

Iniciado: 12 set em 12:35

Instruções do teste

O Trabalho Prático do Módulo 2 está disponível!

1. Instruções para realizar o trabalho prático

Consulte a data de entrega no teste e em seu calendário.

Reserve um tempo para realizar a atividade, leia as orientações e enunciados com atenção. Em caso de dúvidas utilize o "Fórum de dúvidas do Trabalho Prático do Módulo 2".

Para iniciá-lo clique em "Fazer teste". Você tem somente **uma** tentativa e não há limite de tempo definido para realizá-lo. Caso precise interromper a atividade, apenas deixe a página e, ao retornar, clique em "Retomar teste".

Clique em "Enviar teste" **somente** quando você concluí-lo. Antes de enviar confira todas as questões.

O gabarito será disponibilizado a partir de sexta-feira, **16/09/2022**, às 23h59.

Bons estudos!

2. O arquivo abaixo contém o enunciado do trabalho prático:

Enunciado do Trabalho Prático – Módulo 2 – Analista de Machine Learning.pdf

3. O arquivo abaixo contém o notebook e o dataset para resolução do trabalho prático:

Notebook do Trabalho Prático - Módulo 2 - Analista de Machine Learning.ipynb

corpus.csv [_ \(https://raw.githubusercontent.com/Gunjibedi/Text-Classification/master/corpus.csv\)](https://raw.githubusercontent.com/Gunjibedi/Text-Classification/master/corpus.csv)



Pergunta 1

1,78 pts

O bloco de código responsável por fazer o download dos módulos necessários do NLTK para a correta execução do programa consiste em:

- ☐ nltk.download('punkt') nltk.download('confusion_matrix')
nltk.download('averaged_perceptron_tagger') nltk.download('stopwords')
- ☐ nltk.download('pos_tag') nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger') nltk.download('stopwords')
- ☒ nltk.download('punkt') nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger') nltk.download('stopwords')

- ☐ nltk.download('punkt') nltk.download('wordNetLemmatizer')
nltk.download('averaged_perceptron_tagger') nltk.download('accuracy_store')



Pergunta 2

1,78 pts

A definição de sementes aleatórias pode ser definida pelo seguinte código:

- ☐ random.seed(500)
- ☐ np.randrange.seed(500)
- ☒ np.random.seed(500)
- ☐ np.seed(500)



Pergunta 3

1,78 pts

Qual o tipo de dados da variável Corpus criada?

- ☐ Pandas Array
- ☐ String
- ☒ Pandas Data Frame
- ☐ Numpy Array



Pergunta 4

1,78 pts

Para remover linhas em branco, se houver, devemos utilizar qual trecho de código?

- ☐ Corpus['text'].droptext(inplace=True)
- ☐ Corpus['text'].drop(inplace=True)

- ☒ `Corpus['text'].dropna(inplace=True)`
- ☐ `Corpus['text'].droptext(inplace=False)`



Pergunta 5

1,78 pts

Para passar todo o texto para letras minúsculas, devemos utilizar qual trecho de código?

- ☒ `Corpus['text'] = [entry.lower() for entry in Corpus['text']]`
- ☐ `Corpus['text'] = [entry.short() for entry in Corpus['text']]`
- ☐ `Text['corpus'] = [entry.split() for entry in Text['corpus']]`
- ☐ `Text['corpus'] = [entry.lower() for entry in Text['corpus']]`



Pergunta 6

1,78 pts

Para quebrar o corpus em um conjunto de palavras, devemos utilizar qual trecho de código?

- ☐ `Text['corpus'] = [word_stemmatize(entry) for entry in Text['corpus']]`
- ☒ `Corpus['text'] = [word_tokenize(entry) for entry in Corpus['text']]`
- ☐ `Corpus['text'] = [word_lemmatize(entry) for entry in Corpus['text']]`
- ☐ `Corpus['text'] = [word_token (entry) for entry in Corpus['text']]`



Pergunta 7

1,78 pts

Para fazer o mapa de taggeamento das palavras em Adjetivo, Verbo e Advérbio, devemos utilizar qual trecho de código?

- ☐ `tag_map = defaultarray(lambda : wn.NOUN) tag_map['J'] = wn.ADJ tag_map['V'] =`

wn.VERB tag_map['R'] = wn.ADV

- ☐ tag_map = defaultlist(lambda : wn.ADV) tag_map['J'] = wn.NOUM tag_map['V'] = wn.VERB tag_map['R'] = wn.ADV
- ☐ tag_map = defaultdict(df : wn.NOUN) tag_map['J'] = wn.ADJ tag_map['V'] = wn.VERB tag_map['R'] = wn.ADV
- ☒ tag_map = defaultdict(lambda : wn.NOUN) tag_map['J'] = wn.ADJ tag_map['V'] = wn.VERB tag_map['R'] = wn.ADV



Pergunta 8

1,78 pts

A lematização é o processo de converter uma palavra em sua forma básica, e o wordnet é um grande banco de dados lexical disponível gratuitamente e publicamente para a língua inglesa, com o objetivo de estabelecer relações semânticas estruturadas entre palavras. Ele também oferece recursos de lematização e é um dos primeiros e mais usados lematizadores. Para iniciar o WordNet Lemmatizer, devemos utilizar qual trecho de código?

- ☐ word_Lemmatized = word_Lemmatized()
- ☒ word_Lemmatized = WordNetLemmatizer()
- ☐ word_Lemmatized = word.lemmatize ()
- ☐ word_Lemmatized = Lemmatizer()



Pergunta 9

1,78 pts

Para separar o conjunto entre treino e teste, com 70% para treino e 30% para teste, devemos utilizar qual trecho de código?

- ☐ Train_X, Test_X, Train_Y, Test_Y = model_selection.train_test_split(Corpus['text_final'],Corpus['label'],test_size=0.2)
- ☒ Train_X, Test_X, Train_Y, Test_Y = model_selection.train_test_split(Corpus['text_final'],Corpus['label'],test_size=0.3)
- ☐ Train_X, Test_X, Train_Y, Test_Y = model_selection.train_test_split(Corpus['text_final'],Corpus['label'],test_size=0.7)

- ☐ Train_X, Test_X, Train_Y, Test_Y =
model_selection.train_test_split(Corpus['text_final'],Corpus['label'],test_size=0.8)



Pergunta 10

1,78 pts

Label encoder pode ser usado para normalizar rótulos e também para transformar rótulos não numéricos (desde que sejam hashable e comparáveis) em rótulos numéricos. Para iniciar a transformação dos dados categóricos do tipo string no conjunto de dados em valores numéricos que o modelo possa entender, devemos utilizar qual trecho de código?

- ☐ Encoder = Encoder()
- ☒ Encoder = LabelEncoder()
- ☐ Encoder = Label ()
- ☐ Encoder = EncoderLabel()



Pergunta 11

1,78 pts

Ao utilizar o TF-IDF, com o tamanho máximo do vocabulário definido em 5000, devemos utilizar qual trecho de código?

- ☐ Tfidf_vect = TfidfVectorizer(max_features=5000) Tfidf_vect.clf(Corpus['corpus'])
Train_X_Tfidf = Tfidf_vect.transform(Train_X) Test_X_Tfidf =
Tfidf_vect.transform(Test_X)
- ☐ Tfidf_vect = TfidfVector(max_features=5000) Tfidf_vect.fit(Corpus['text_final'])
Train_Y_Tfidf = Tfidf_vect.transform(Train_Y) Test_Y_Tfidf =
Tfidf_vect.transform(Test_Y)
- ☐ Tfidf_vect = TfidfVectorizer(max_features=8000) Tfidf_vect.fit(Corpus['text_final'])
Train_X_Tfidf = Tfidf_vect.transform(Train_X) Test_X_Tfidf =
Tfidf_vect.transform(Test_X)
- ☒ Tfidf_vect = TfidfVectorizer(max_features=5000) Tfidf_vect.fit(Corpus['text_final'])
Train_X_Tfidf = Tfidf_vect.transform(Train_X) Test_X_Tfidf =
Tfidf_vect.transform(Test_X)



Pergunta 12

1,78 pts

Considerando os valores de (`n_estimators = 10`, `random_state = 0`) e o conjunto de treino e teste como 70/30, o Random Forest teve a sua acurácia prevista na faixa de qual porcentagem?

☐ [95-100]

☒ [75-80]

☐ [87-90]

☐ [81-86]



Pergunta 13

1,78 pts

Considerando os valores de (`n_estimators = 100`, `random_state = 0`) e o conjunto de treino e teste como 80/20 em relação ao Random Forest, marque a alternativa CORRETA:

☒ O RF possui uma boa acurácia, mas pode ser melhorado com técnicas de otimização.

☐ O RF possui uma acurácia alta, indicando uma taxa de erro alta.

☐ O RF possui uma acurácia baixa, indicando uma baixa taxa de erro.

☐ O RF possui uma acurácia baixa devido à impossibilidade de ajustar seus parâmetros.



Pergunta 14

1,86 pts

Pensando na perspectiva de melhoria dos modelos de Machine Learning, podemos avaliar o ajuste de hiperparâmetros, considerando as seguintes técnicas:

☒ Random Search, Grid Search, Cross Validation.

☐ Cross Validation, Random Search e Aleatory Validation.

☐ Random Search, Grid Search e Majority Voting.

☐ Cross Validation, Aleatory Validation, Random Cross.

Não salvo

Enviar teste