Instituto Infnet

Um modelo de redes neurais profundas (DNN) para classificação de estabelecimentos nos segmentos Autosserviço, Mercado Quente e Mercado Frio com diferentes funções de ativação

Projeto da disciplina de Deep Learning com TensorFlow

Winicius Botelho Faquieri

AGENDA

1. Resumo dos Resultados

2. Método

    Análise de Negócio, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment

3. Resultados

4. Conclusão

Resumo dos Resultados

- Neste notebook, um modelo de rede neural profunda é desenvolvido para classificar empresas do varejo de bebidas em diferentes canais de distribuição. Muitos experimentos foram feitos para o modelo com diferentes funções de ativação e para diferentes números de época.

O problema de negócio

- **Canais de distribuição - entendendo o problema**
- Objetivo: classificar empresas do varejo relativo ao mercado de bebidas frias em 03 segmentos ou canais de distribuição, a saber:

Segmento I
Supermercados com, no mínimo 5 *checkouts* e distribuidores multimarcas

Segmento II
Supermercados com, no máximo 4 *checkouts* e distribuidores multimarcas

Segmento III
Bares, lanchonetes, restaurantes, churrascarias, pizzarias, padarias, confeitarias e lojas de

A base de dados utilizadas e seus atributos

- **NUM_DOC_DEST:** identificador dos estabelecimentos
- **N_TRANS:** número de transações realizadas por estabelecimento ou cupom fiscal
- **VAL_UNIT_MEDIO:** valor médio de compra
- **VOLUME_COMPRA_MEDIO:** quantidade média de compras realizadas
- **CESTA_PROD_DIFER:** diversidade de produtos comprados
- **GRUPO_PROD_DIFER:** diversidade de grupos de produtos diferentes

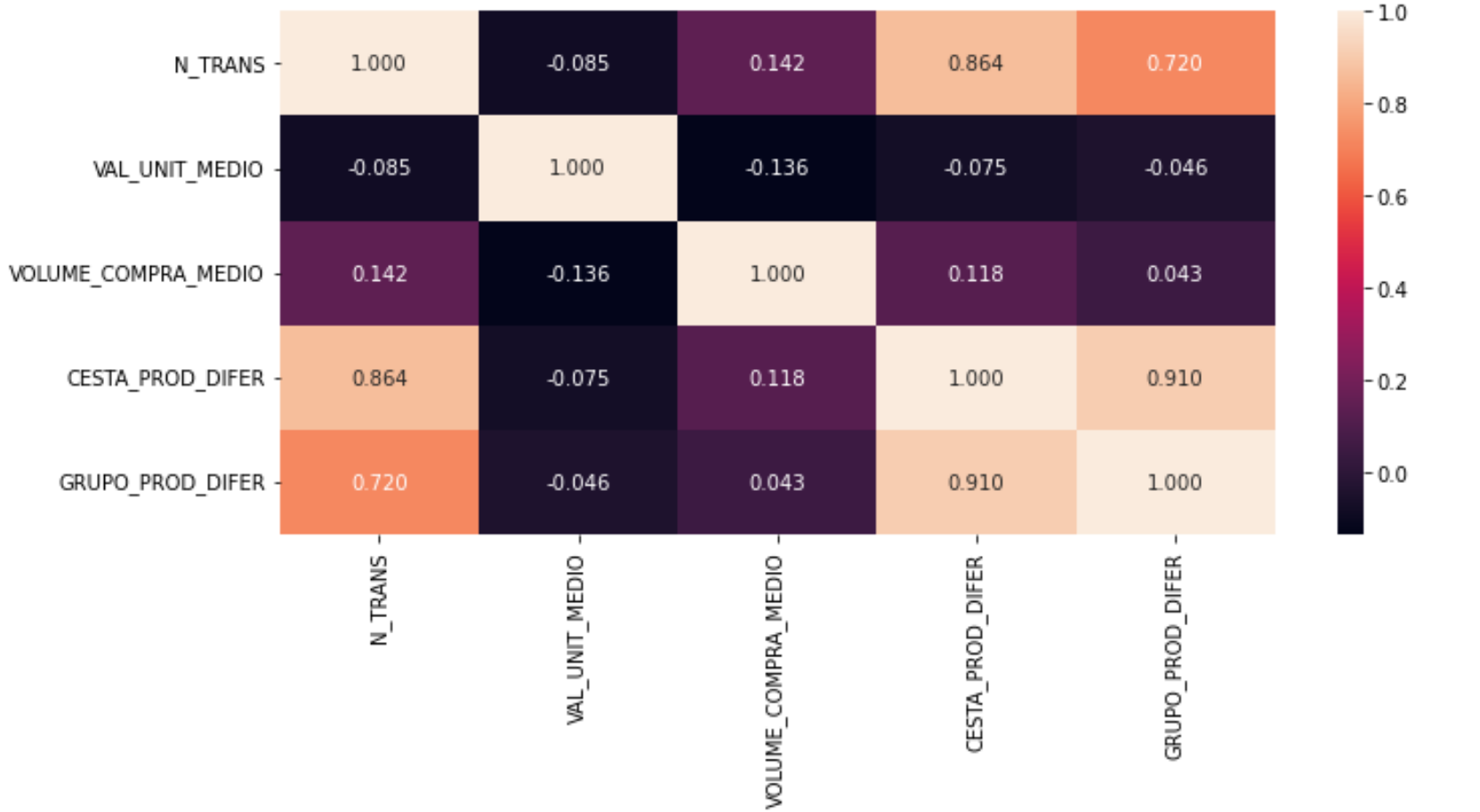
Diferentes features foram utilizadas com o intuito de traçar um perfil de compra dos estabelecimentos.

```
df.head()
```

	NUM_DOC_DEST	CNAE_DEST	N_TRANS	VAL_UNIT_MEDIO	VOLUME_COMPRA_MEDIO	CESTA_PROD_DIFER	GRUPO_PROD_DIFER	SEGMENTO	target
0	12058181000196	4712100	668	5.460947248502994	146.9311377245509	62	17	2	II
1	93209765032582	4711301	3847	21.188510626992255	3171.2399272160123	266	36	1	I
2	13004510017235	4711302	8293	20.919979153356135	295.0801881104546	244	37	1	I
3	01031452000101	5611201	170	27.170852152941176	167.41176470588235	28	12	3	III
4	13964957000108	5611201	128	20.630893885416405	66.484375	36	18	3	III

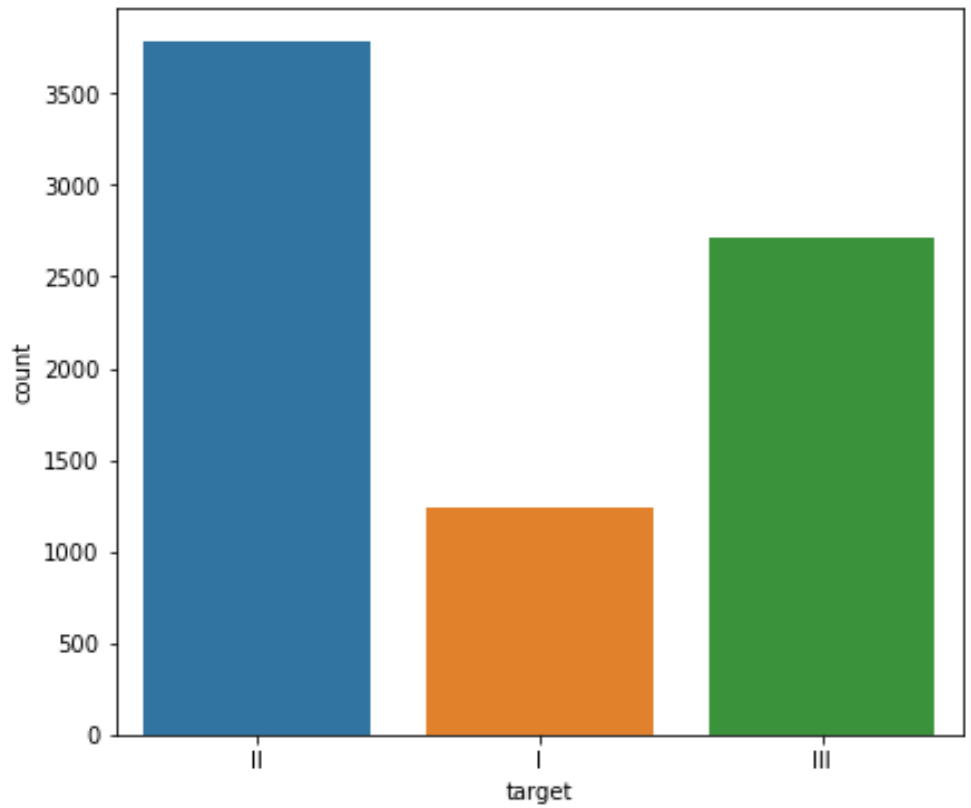
Matriz de Correlação

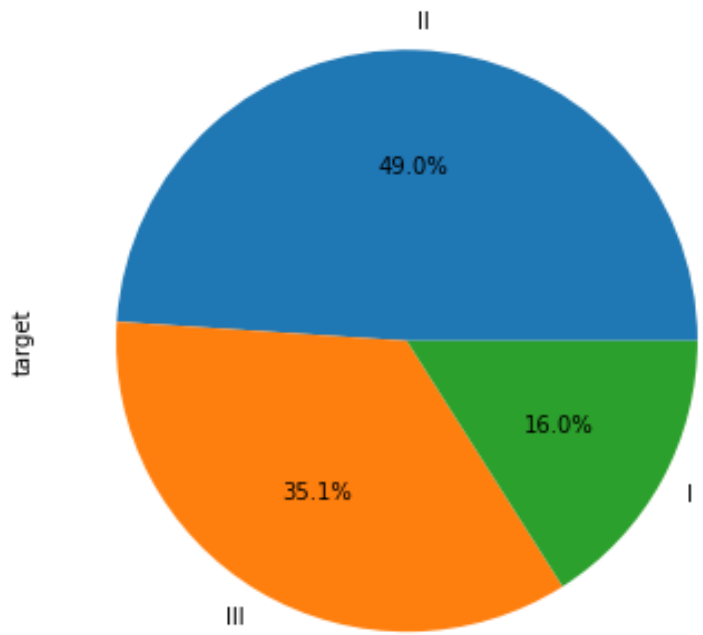
```
sns.heatmap(corr_matrix, annot=True, fmt='.3f');
```



Distribuição da variável target

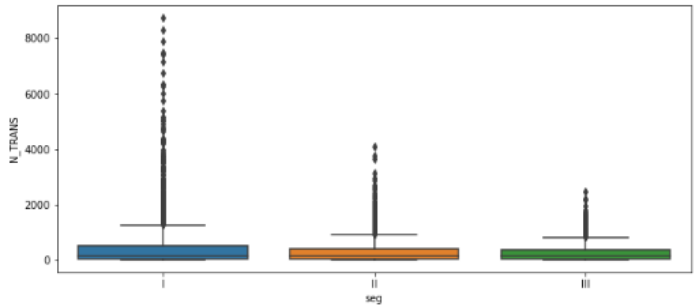
```
features = df.copy()
fig, ax=plt.subplots(1,2,figsize=(15,6))
_ = sns.countplot(x='target', data=features, ax=ax[0])
_ = features['target'].value_counts().plot.pie(autopct="%1.1f%%", ax=ax[1])
```





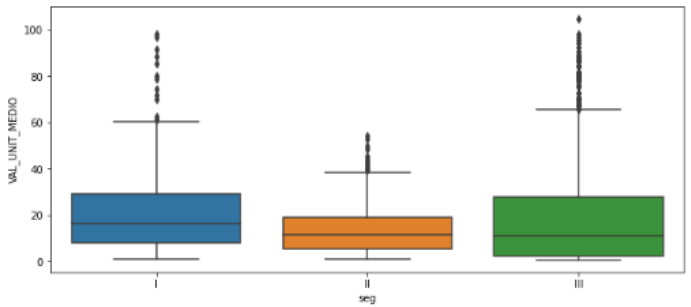
DATA PREPARATION

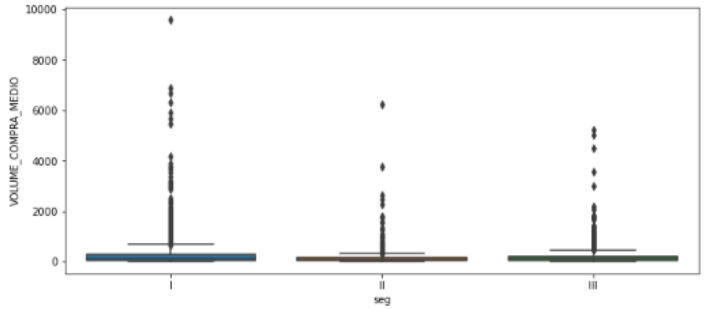
- Rescaling: reescala para o intervalo 0 e 1 (distribuições não Gaussianas):
- - Min-Max Scaler: utiliza o range
- - Robust Scaler (RS): utiliza o IQR
- **Utilizou-se o método RS devido a presença de valores atípicos.**




```
sns.boxplot( x='seg', y='VAL_UNIT_MEDIO', data=features )
```

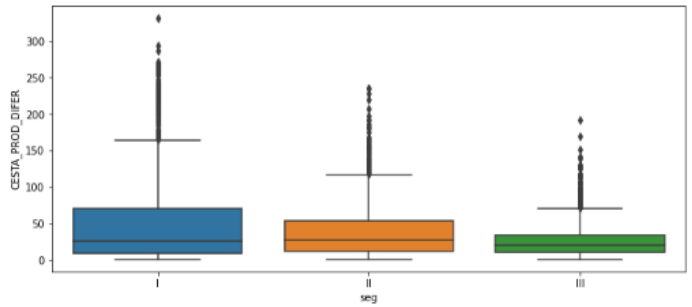
```
<AxesSubplot:xlabel='seg', ylabel='VAL_UNIT_MEDIO'>
```






```
sns.boxplot( x='seg', y='CESTA_PROD_DIFER', data=features )
```

```
<AxesSubplot:xlabel='seg', ylabel='CESTA_PROD_DIFER'>
```



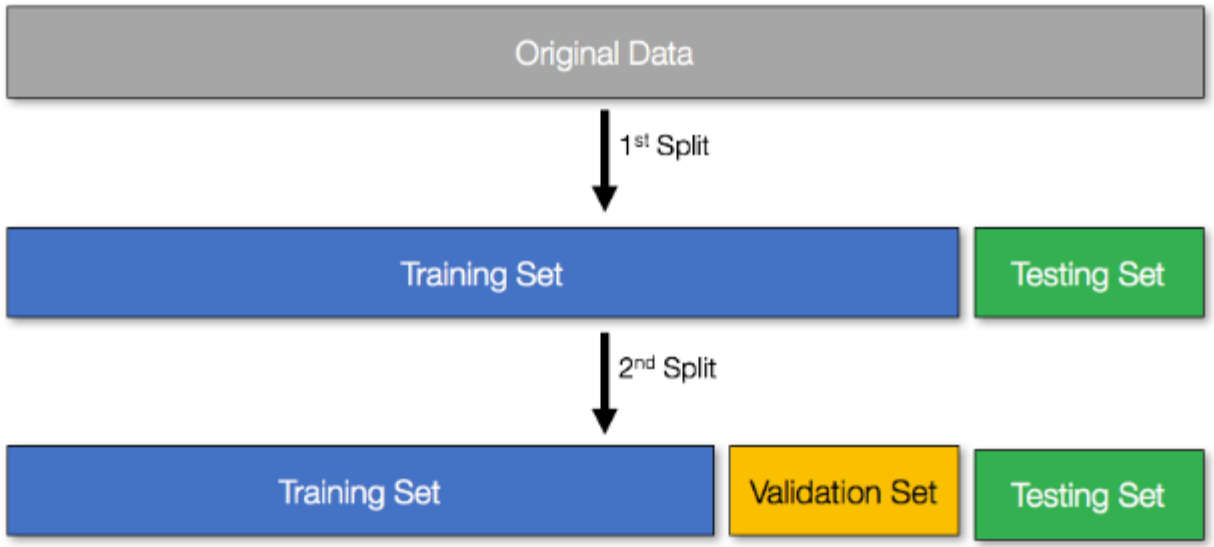
Dividindo os dados em conjuntos de treinamento, teste e validação

Setup:

```
- Full data 100%
- test      20%  -> Generalization!
- training  80%, sendo 80% train + 20% validation -> Aprendizado!
```

O conjunto de treinamento possui 6.180 linhas e 5 colunas, enquanto o conjunto de teste possui 1546 linhas e 4 colunas.

Como regra geral, devemos dividir os dados originais em 80% de treinamento e 20% de teste e, em seguida, dividir os dados de treinamento em 80% de treinamento e 20% de validação novamente.



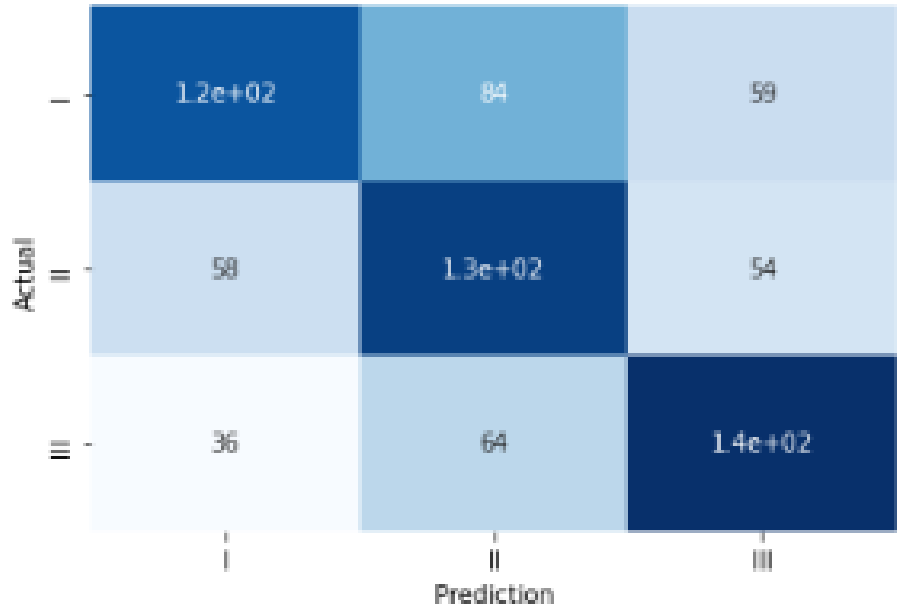
Qual topologia de Rede Neural foi escolhida e porque

- Analisamos a arquitetura do MLP que usamos, que consiste em 2 camadas ocultas, com 50 nós na primeira camada oculta e 20 nós na segunda camada oculta.
- Treinamos nosso MLP usando o conjunto de treinamento, utilizando algoritmo otimizador **Adam** para modificar os pesos e vieses na rede neural em mais de **400** iterações, melhorando gradualmente a precisão do modelo.

```
# TOP 10
results = pd.read_excel('../Tabela_Resultados.xlsx', sheet_name="4")
results.head(10)
```

	1st_hidden_layer	2nd_hidden_layer	epoch_number	accuracy_rate
0	Relu	Tanh	300	0.7000
1	Tanh	Tanh	400	0.6929
2	Tanh	Tanh	500	0.6700
3	Relu	Tanh	500	0.6700
4	Relu	Tanh	400	0.6600
5	Tanh	Tanh	300	0.6500
6	Relu	Tanh	200	0.6500
7	Relu	Relu	400	0.6500
8	Relu	Relu	100	0.6500
9	Relu	Relu	200	0.6500

Como será validado o resultado estatístico da rede/modelo



- Por fim, avaliamos nosso modelo usando métricas como **acurácia** e **matriz de confusão**. A melhor precisão alcançada foi de apenas 68,9%.
- O que significa que, dadas as quatro features de um novo estabelecimento, nosso modelo é capaz de prever com precisão de ~70% se essa empresa pertence ao segmento Autosserviço, Mercado Quente ou Mercado Frio.

Conclusão

- Em geral, qualquer limitação no desempenho geralmente se deve à falta de recursos fortes no conjunto de dados, e não à complexidade da rede neural usada.
- O conjunto de dados consiste apenas em quatro features, e pode-se argumentar que esses recursos sozinhos são insuficientes para realmente dizer se uma empresa pertence ao segmento de autosserviço, mercado quente ou mercado frio.

Muito Obrigado

- Perguntas
- Membros do Grupo