

《Python 程序设计高阶》

(2023-2024 学年第 1 学期)

课题设计报告

学号:2021329600006 姓名: 陈昊天 班级: 计算机科学与技术 21(4) 班
学号:2021329621213 姓名: 陈佳伟 班级: 计算机科学与技术 21(3) 班
学号:2021329621257 姓名: 冯佳钧 班级: 计算机科学与技术 21(4) 班

§1 引言

随着计算机技术的飞速发展,传统商业模式正经历着前所未有的变革。实体店铺,作为历史悠久的商业模式,曾是商业活动的主要场所。然而,随着电话的普及和电子商务的兴起,商业模式发生了翻天覆地的变化。特别是电子商务的兴起,极大地拓宽了商业的边界,使得交易活动不再受地理位置的限制。马云曾言:“让天下没有难做的生意”,这不仅预示了电子商务的兴盛,更标志着实体店模式的逐渐式微。

近年来,直播电商作为电子商务的新兴形态,进一步颠覆了传统商业和广告模式。通过直播平台,主播能够利用个人影响力,直接与广大粉丝互动,实现高效的市场传播。这种新型的商业模式在提升市场效率的同时,也对实体店造成了巨大的冲击。头部主播的出现,更是将这种冲击提升到了新的高度。他们通过高度分工的业务体系,有效提升了销售效率,但也导致了实体店的大量关闭和从业人员的减少。

本报告旨在深入探讨和分析这一现象,从四个关键方面——实体店与电商、电商与直播电商、直播电商与头部主播、头部主播与实体店——对直播电商的崛起及其对传统实体店的影响进行全面分析。通过对直播电商市场占比、头部主播在直播电商中的占比、电商与实体店的比例等关键数据的收集和预测,本报告旨在揭示头部主播的销售额与实体店之间的关系,并探讨对头部主播或直播电商行业进行定量限制的可能性与效果。本报告将使用 Python 作为主要的分析工具,结合丰富的数据资源,为读者呈现一个深入、全面的分析报告。

§2 研究方法

在数据的搜集和预处理过程中,我们发现很难直接找到头部主播的相关数据与实体店直接对比,故而采用自顶而下、逐步展开的方法研究课题。

处于最顶层的是实体店和电商。我们将从全国线下零售额与全国网上零售额的比较得出实体店和电商的比例关系。在这里,我们讨论的是广义的实体店和电商。然后,通过电商与直播电商的比较,得出直播电商的市场份额在电商行业中的占比。最后,研究直播电商和头部主播的数值关系。

得到以上的数据后,即可探讨头部主播的发展和实体店衰落之间的关系。

必须指出的是,我们的研究没有考虑很多外部因素,如疫情流行、国际经济形势的影响。尽管如此,对于头部主播和实体店间关系的研究仍然是有意义的。

§3 实证分析

§3.1 实体店与电商的比较分析

本节讨论电商对实体经济的影响,对未来一年的网上零售额与社会零售总额进行预测。

3.1.1 电商

电商，即电子商务。

电子商务就是利用现在先进的电子技术从事各种商业活动的方式，是在互联网 Internet 环境下，实现消费者的网上购物、商户之间的网上交易和在线电子支付的一种新型的商业运营模式。电子商务可提供网上交易和管理等全过程的服务，因此它具有广告宣传、咨询洽谈、网上订购，网上支付、电子帐户、服务传递、意见征询、交易管理等各项功能。

“十三五”时期，我国电子商务取得了显著成就：电子商务交易额从 2015 年的 21.8 万亿元增至 2020 年的 37.2 万亿元；全国网上零售额 2020 年达到 11.8 万亿元，我国已连续 8 年成为全球规模最大的网络零售市场；2020 年实物商品网上零售额占社会消费品零售总额的比重接近四分之一，电子商务已经成为居民消费的主渠道之一 [1]。

3.1.2 电子商务的特点

1. 交易便捷。借助互联网平台与第三方支付平台，客户与商家之间可以进行在线交易，实现实时交易和支付。

2. 成本低廉。相较于传统贸易方式需要建立实体店面、雇佣人员、租赁物流仓库，电子商务无需实体店面、只需少量员工即可正常运营。

3. 市场范围广。传统贸易中的实体店，市场只有所在地区的很小的范围，而电子商务的市场范围覆盖全国，甚至可以实现跨境贸易，拓展国际市场。

3.1.3 电子商务对实体经济的影响

国家统计局每个月都会公布上月或之前两月的社会零售总额，所以可从国家统计局官网获得历史网上零售额与社会零售总额的数据。由于国家统计局公布的数据从 2019 年开始，截至目前，已公布 2023 年 11 月的数据，2023 年 12 月的数据还未公布。

国家统计局不单独公布每年 1 月、2 月的社会零售总额数据，而是将 1 月、2 月发数据放在一起公布，为了便于处理，将 1 月、2 月的数据分别设置为 1、2 月总数据的一半。

借助所获得的数据，我们可以对未来一年的网上零售额与社会零售总额分别进行预测，预测结果如图 1 与图 2 所示。

```
1 import pandas as pd
2 from pmdarima import auto_arima
3 import matplotlib.pyplot as plt
4
5 plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签
6 plt.rcParams['axes.unicode_minus'] = False # 用来正常显示负号
7     #有中文出现的情况，需要u'内容'
8
9 # 从Excel文件读取数据
10 df1 = pd.read_excel('网上零售额.xlsx', header=None, names=['月份', '值'])
11
12 # 将月份转换为pandas的日期格式
13 df1['月份'] = pd.to_datetime(df1['月份'], format='%Y年%m月')
14
15 # 设置月份为索引
16 df1.set_index('月份', inplace=True)
```

```

16
17 # 使用auto_arima自动寻找最佳ARIMA模型
18 model = auto_arima(df1['值'], seasonal=True, m=12)
19 print(model.summary())
20
21 # 拟合模型
22 model.fit(df1['值'])
23
24 # 进行预测
25 forecast = model.predict(n_periods=13)
26
27 # 创建正确的预测日期索引（每个月的第一天）
28 forecast_index = pd.date_range(start=df1.index[-1] +
    pd.offsets.MonthBegin(1), periods=13, freq='MS')
29
30 # 创建预测数据的DataFrame
31 forecast_df1 = pd.DataFrame({'预测': forecast}, index=forecast_index)
32
33 # 打印forecast_df以进行调试
34 print(forecast_df1)
35
36 # 可视化历史数据和预测数据
37 plt.figure(figsize=(10, 5))
38 plt.plot(df1, label='历史数据')
39 plt.plot(forecast_df1, label='预测', color='red')
40 plt.title("2024年网上零售额预测")
41 plt.xlabel('日期')
42 plt.ylabel('单位（亿元）')
43 plt.legend()
44 plt.show()
45
46 # 将预测数据转换为与原始数据相同的格式
47 forecast_df1.reset_index(inplace=True)
48 forecast_df1.columns = ['月份', '预测值']
49 forecast_df1['月份'] = forecast_df1['月份'].dt.strftime('%Y年%m月')
50
51 # 合并原始数据和预测数据
52 combined_df1 = pd.concat([df1.reset_index(), forecast_df1],
    ignore_index=True)
53
54 # 写入到新的Excel文件
55 combined_df1.to_excel('2024年网上零售额.xlsx', index=False)
56
57 print("预测数据已成功写入 '2024年网上零售额.xlsx' 文件.")

```

```

58
59 df2 = pd.read_excel('零售总额.xlsx', header=None, names=['月份', '值'])
60
61 # 将月份转换为pandas的日期格式
62 df2['月份'] = pd.to_datetime(df2['月份'], format='%Y年%m月')
63
64 # 设置月份为索引
65 df2.set_index('月份', inplace=True)
66
67 # 使用auto_arima自动寻找最佳ARIMA模型
68 model = auto_arima(df2['值'], seasonal=True, m=12)
69
70 # 拟合模型
71 model.fit(df2['值'])
72
73 # 进行预测
74 forecast = model.predict(n_periods=13)
75
76 # 创建正确的预测日期索引（每个月的第一天）
77 forecast_index = pd.date_range(start=df2.index[-1] +
    pd.offsets.MonthBegin(1), periods=13, freq='MS')
78
79 # 创建预测数据的DataFrame
80 forecast_df2 = pd.DataFrame({'预测': forecast}, index=forecast_index)
81
82 # 打印forecast_df以进行调试
83 print(forecast_df2)
84
85 # 可视化历史数据和预测数据
86 plt.figure(figsize=(10, 5))
87 plt.plot(df2, label='历史数据')
88 plt.plot(forecast_df2, label='预测', color='red')
89 plt.title("2024年零售总额预测")
90 plt.xlabel('日期')
91 plt.ylabel('单位（亿元）')
92 plt.legend()
93 plt.show()
94
95 # 将预测数据转换为与原始数据相同的格式
96 forecast_df2.reset_index(inplace=True)
97 forecast_df2.columns = ['月份', '预测值']
98 forecast_df2['月份'] = forecast_df2['月份'].dt.strftime('%Y年%m月')
99
100 # 合并原始数据和预测数据

```

```

101 combined_df2 = pd.concat([df2.reset_index(), forecast_df2],
102                             ignore_index=True)
103 # 写入到新的Excel文件
104 combined_df2.to_excel('2024年零售总额.xlsx', index=False)
105
106 print("预测数据已成功写入 '2024年零售总额.xlsx' 文件.")

```

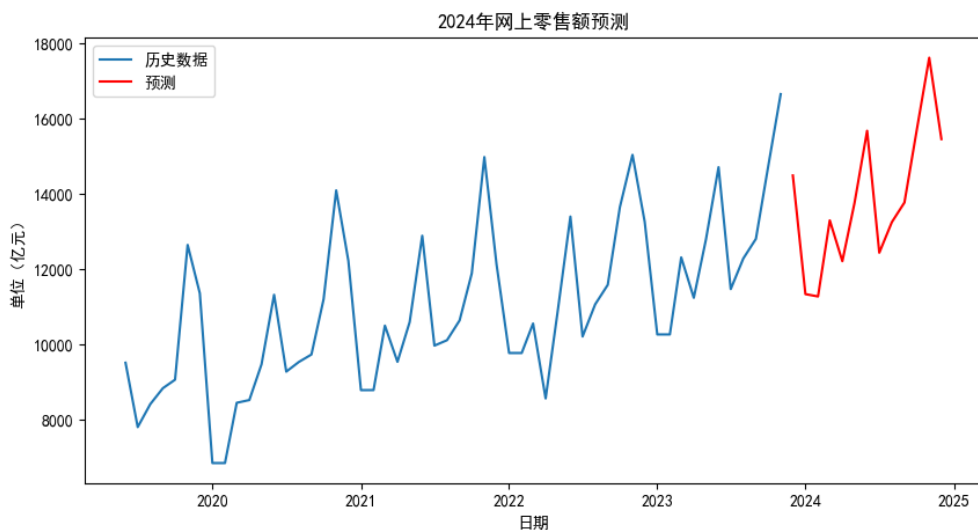


图 1: 网上零售额预测结果

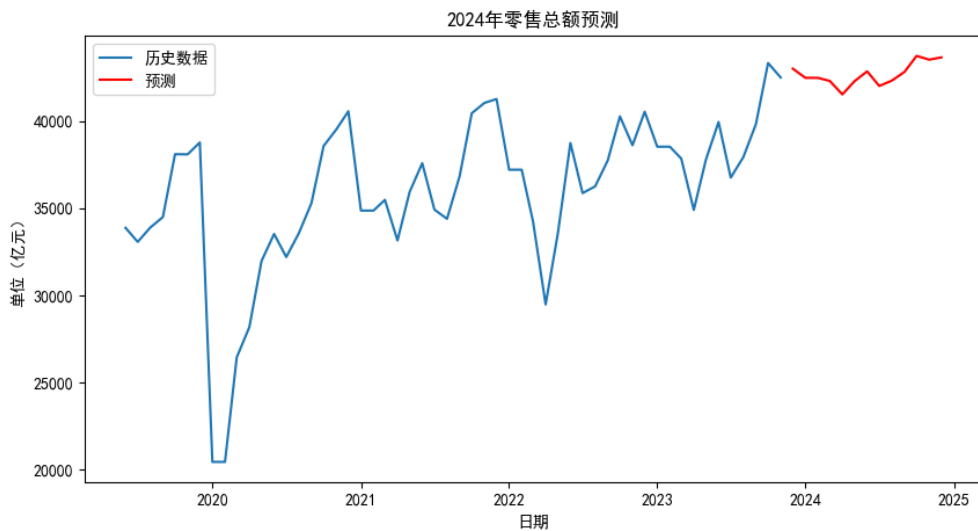


图 2: 社会零售总额预测结果

得到预测数据后，我们将历史数据与预测得到的数据放在一起，查看过去及未来一年中，网上零售额在社会零售总额中所占的比例

```

1 import pandas as pd
2 import matplotlib.pyplot as plt

```

```

3
4 plt.rcParams['font.sans-serif'] = ['SimHei'] # 用来正常显示中文标签
5 plt.rcParams['axes.unicode_minus'] = False # 用来正常显示负号
   #有中文出现的情况，需要u'内容'
6
7 # 读取数据
8 online_sales = pd.read_excel('历史+预测网上零售额.xlsx', header=None,
   names=['月份', '网上零售额'])
9 total_sales = pd.read_excel('历史+预测零售总额.xlsx', header=None,
   names=['月份', '零售总额'])
10 # print(online_sales)
11
12
13 # 合并数据
14 merged_data = pd.merge(online_sales, total_sales, on='月份', how='inner')
15
16 # print(merged_data)
17
18 # 计算占比
19 merged_data['Online_Percentage'] = merged_data['网上零售额'] /
   merged_data['零售总额'] * 100
20 merged_data['线下零售额'] = merged_data['零售总额'] -
   merged_data['网上零售额']
21 # merged_data.to_excel('合并后的数据.xlsx', index=False)
22 # 绘制图表
23 plt.figure(figsize=(10, 6))
24 plt.plot(merged_data['月份'], merged_data['Online_Percentage'], marker='o')
25 plt.xlabel('月份')
26 plt.ylabel('百分比 (%)')
27 plt.title('网上零售额占社会零售总额百分比')
28 plt.grid(True)
29 plt.show()

```

得到结果如图 3 所示。从图中可以看出，网上零售额在社会零售总额中所占的比例总体呈上升趋势，而从图 2 中可以得知，社会零售总额的变化并不明显。综合图 2 与图 3 进行分析可知：未来，电子商务会逐渐蚕食实体经济的市场，压缩实体经济的生存空间。

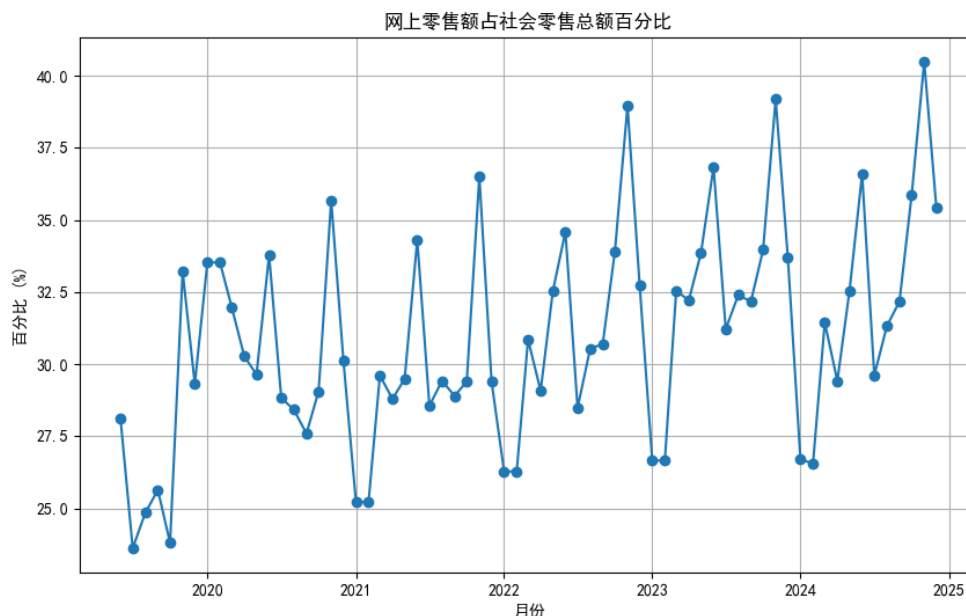


图 3: 网上零售额占零售总额百分比

§3.2 电商与直播电商的比较分析

本节讨论电商与直播电商之间的差异性，对未来五年的网上零售额与直播电商的市场份额进行预测，并根据预测数据和原有数据给出直播电商在电商中的占比。

3.2.1 直播电商的介绍

直播电商是一种结合了在线直播和电子商务的商业模式，通过实时的视频直播来进行商品展示、销售和互动。这种商业模式的兴起主要得益于互联网和移动互联网的发展，以及社交媒体的普及。

从销售方式看，传统电商主要通过网站、移动应用等平台，以文字、图片和一些视频展示商品信息，消费者通过浏览页面选择商品进行购买。直播电商通过实时视频直播展示商品，主播可以在直播中演示产品使用、回答观众提问，观众可以直接在直播中点击购买链接完成购物。

在营销手段方面，电商平台一般采用搜索引擎优化、广告投放等方式进行推广和营销。直播电商常常通过直播平台的推荐、明星主播效应、限时促销等形式来进行营销，营造紧迫感促进消费。

在购物决策速度上，用户在电商平台上购物一般需要经过比较、思考，购物决策相对较为谨慎。由于直播过程中的实时互动和限时促销等因素，用户在直播电商平台上的购物决策可能更为迅速。

3.2.2 网上零售额与直播电商的分析和预测

以下为模型预测未来 5 年的网上销售额：

数据输入部分：在 `y_data` 中储存官方公布的每月已确定的往期网上销售额。并利用 `np.array()` 将 `y_data` 内的数据转换为 `np` 数组形势，并将 `y_data` 的数据格式变更为浮点型

```

1 import tensorflow as tf
2 import numpy as np
3
4 x_data, y_data = [], []
5

```

```

6 x_data=[
7     2017.01,2017.02,2017.03,2017.04,2017.05,2017.06,
8     2017.07,2017.08,2017.09,2017.10,2017.11,2017.12,
9     2018.01,2018.02,2018.03,2018.04,2018.05,2018.06,
10    2018.07,2018.08,2018.09,2018.10,2018.11,2018.12,
11    2019.01,2019.02,2019.03,2019.04,2019.05,2019.06,
12    2019.07,2019.08,2019.09,2019.10,2019.11,2019.12,
13    2020.01,2020.02,2020.03,2020.04,2020.05,2020.06,
14    2020.07,2020.08,2020.09,2020.10,2020.11,2020.12,
15    2021.01,2021.02,2021.03,2021.04,2021.05,2021.06,
16    2021.07,2021.08,2021.09,2021.10,2021.11,2021.12,
17    2022.01,2022.02,2022.03,2022.04,2022.05,2022.06,
18    2022.07,2022.08,2022.09,2022.10,2022.11,2022.12,
19    2023.01,2023.02,2023.03,2023.04,2023.05,2023.06,
20    2023.07
21 ]
22 y_data=[
23     4234,4346,5465,5135,5483,6410,5544,5894,6276,6563,
24     8956,7445,6014,6257,7047,6474,6899,8119,7053,7332,
25     7590,7754,10150,9376,7683,6300,8396,8060,8202,9520,
26     7811,8421,8844,9070,12651,11366,6510,7202,8457,8529,
27     9478,11325,9284,9541,9739,11210,14099,12227,8651,8936,
28     10506,9545,10601,12894,9975,10119,10644,11894,14984,
29     12135,9543,10015,10562,8572,10912,13403,10217,11071,
30     11589,13658,15043,13268,10272,10272,12319,11245,12798,
31     14715,11476
32 ]
33
34 x_data,y_data=np.array(x_data),np.array(y_data)
35 y_data=y_data.astype('float32')

```

定义一个函数 `split_data` 用于将序列数据划分为输入和目标数据，并返回相应的 NumPy 数组：将过去数据设置为 `x`，将预测数据设置为 `y`，`wth` 作为窗口长度，代表选取数据的范围。

```

1 def split_data(seq,wth,fea):
2     x,y=[],[]
3     for i in range(len(seq)):
4         idx_end=i+wth
5         if idx_end>len(seq)-1:break
6         x_seq,y_seq=seq[i:idx_end],seq[idx_end]
7         x.append(x_seq),y.append(y_seq)
8         x_proc,y_proc=np.array(x),np.array(y)
9         x_proc=x_proc.reshape((x_proc.shape[0],x_proc.shape[1],fea))
10    return x_proc,y_proc
11
12 x_data_split,y_data_split=split_data(y_data,12,1)

```


继承 `tf.keras.Model` 类 `self.c1`: 卷积层, 包含 64 个卷积核, 每个卷积核的大小是 2, 使用 ReLU 激活函数, 步长为 1, padding 为 same。 `self.p1`: 最大池化层, 池化大小为 2, 步长为 None (自动使用池化大小), padding 为 same。 `self.flatten`: 扁平化层, 将卷积和池化层的输出拉直成一维数组。 `self.f1`: 全连接层, 包含 50 个神经元, 使用 ReLU 激活函数, 带有偏置, 权重使用 Glorot uniform 初始化, 偏置使用零初始化。 `self.f2`: 输出层, 包含 1 个神经元, 用于回归任务。

```
1 class CNN(tf.keras.Model):
2     def __init__(self):
3         super(CNN, self).__init__()
4         self.c1 = tf.keras.layers.Conv1D(
5             filters=64, kernel_size=2, activation="relu", strides=1,
6             padding="same"
7         )
8         self.p1 = tf.keras.layers.MaxPooling1D(
9             pool_size=2, strides=None, padding="same"
10        )
11        self.flatten = tf.keras.layers.Flatten()
12        self.f1 = tf.keras.layers.Dense(
13            units=50,
14            activation="relu",
15            use_bias=True,
16            kernel_initializer="glorot_uniform",
17            bias_initializer="zeros",
18        )
19        self.f2 = tf.keras.layers.Dense(units=1)
20
21    def call(self, x):
22        x = self.c1(x)
23        x = self.p1(x)
24        x = self.flatten(x)
25        x = self.f1(x)
26        y = self.f2(x)
27        return y
```

模型的创建与设置训练参数

```
1 model=CNN()
2 model.compile(optimizer='adam',loss='mse',metrics=['accuracy'])
3 model.fit(x_data_split,y_data_split,batch_size=32,epochs=10000)
```

设置希望预计未来多少次的数据, 将窗口前 `n-1` 位作为过去, 预测第 `n` 位的值, 并更新到原始数据中, 再往后迭代

```
1 num_pred=70
2
3 for i in range(num_pred):
```

```

4     if(i==0):x_pred=[y_data[-12:]]
5     else:x_pred=[np.append(x_pred[0][-11:],y_pred[0])]
6     x_pred_split=np.expand_dims(x_pred,axis=-1)
7     y_pred=model.predict(x_pred_split,verbose=0)
8     y_data=np.append(y_data,y_pred[0][0])
9
10 print(y_data)

```

打印图表:

```

1 import matplotlib.pyplot as plt
2
3 time=[]
4 for i in range(len(y_data)): time.append(i)
5
6 data=y_data.tolist()
7
8 plt.plot(time,data)
9 plt.show()

```

结果分析:

1. 对网上销售额的预测

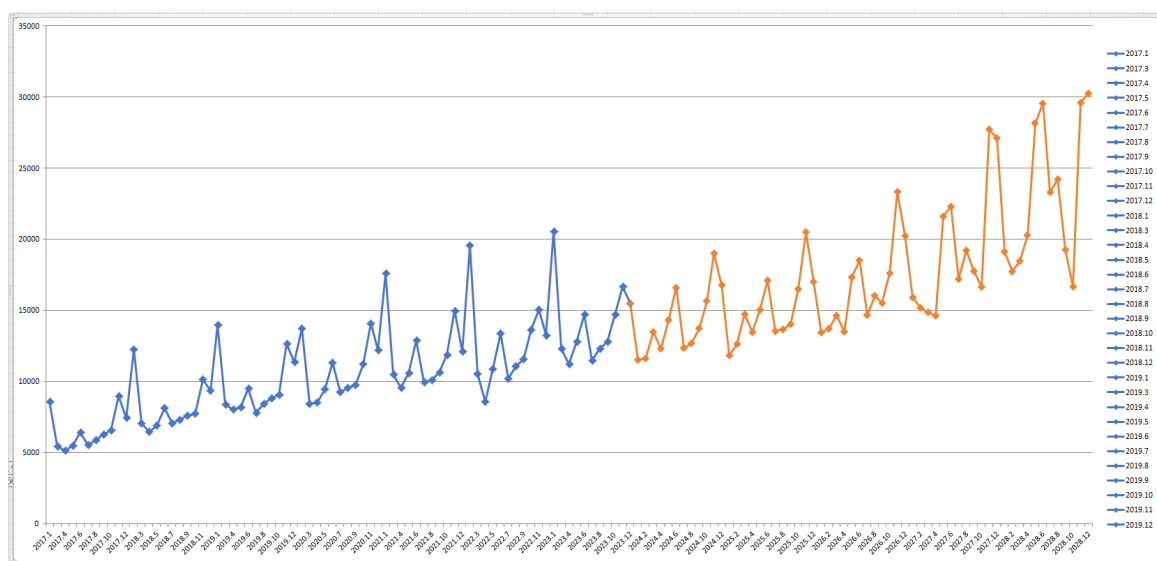


图 4: 对网上销售额的未来 5 年预测

其中从 2023 年 8 月开始预测至 2029 年 5 月, 以已有数据的 8-11 月作为模型预测的准确度参考, 可以表明预测的起伏与实际的起伏相近

| | | | | | |
|--|------------|------------|-----------|------------|------------|
| | 2023. 8 | 2023. 9 | 2023. 10 | 2023. 11 | 2023. 12 |
| | 12290 | 12811 | 14717 | 16656 | 15493. 083 |
| | | | | | |
| | 2023. 8 | 2023. 9 | 2023. 10 | 2023. 11 | 2023. 12 |
| | 11277. 985 | 12551. 346 | 14322. 24 | 16965. 729 | 15493. 083 |

图 5: 对网上销售额的未来 5 年预测数据

从而预测出未来 5 年内每月的可能网上销售额数据，再将销售额数据进行每年的 sum()，用于之后与直播电商的数据比较计算占比。

2. 直播电商的预测额

模型使用的是原网上销售额的预测模型，仅修改了原始数据和每次选取的窗口长度。

```

1 x_data=[
2     2017.0,2018.0,2019.0,2020.0,2021.0,2022.0,2023.0
3 ]
4 y_data=[
5     196.4,1354.1,4437.5,12850,23615.1,35000,45657
6 ]

```

更改窗口长度：

```

1 x_data_split,y_data_split=split_data(y_data,3,1)

```

在预测中使用对应的更改后窗口长度

```

1 for i in range(num_pred):
2     if(i==0):x_pred=[y_data[-3:]]
3     else:x_pred=[np.append(x_pred[0][-2:],y_pred[0])]
4     x_pred_split=np.expand_dims(x_pred,axis=-1)
5     y_pred=model.predict(x_pred_split,verbose=0)
6     y_data=np.append(y_data,y_pred[0][0])

```

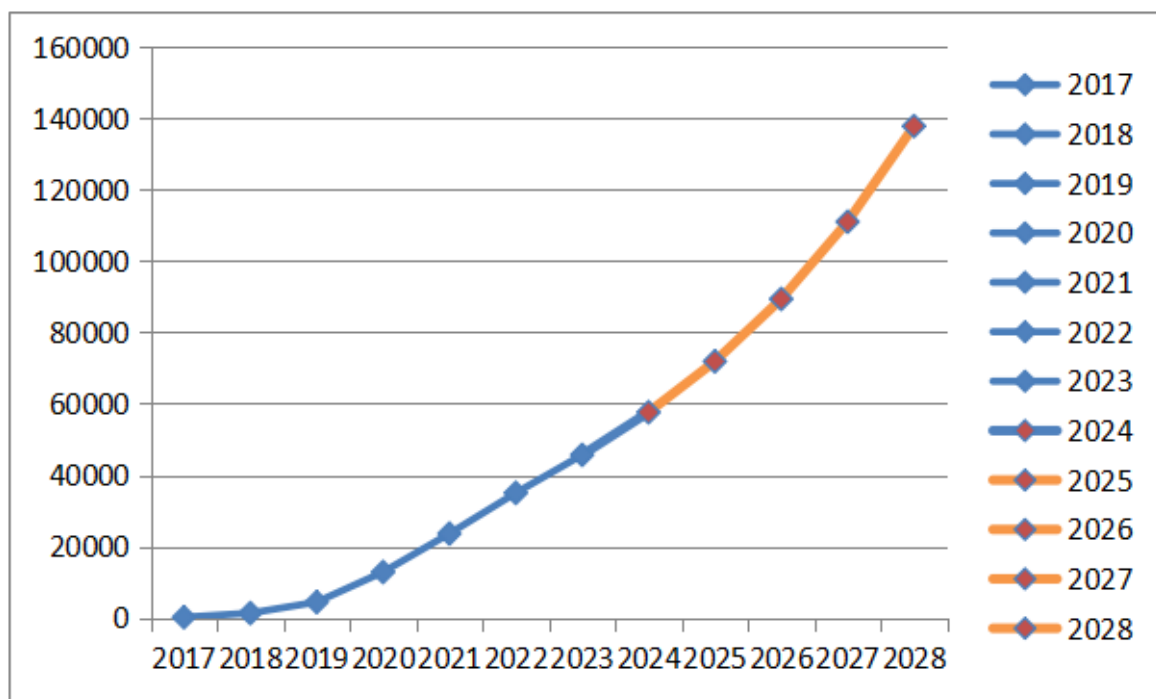


图 6: 直播电商未来 5 年预测额

3. 直播电商预测额与网上销售额预测的占比

| | | | | | | | | | | | | | |
|----|-------------|-------------|-------------|-------------|-------------|-------------|------------|-------------|-----------|-----------|-----------|-----------|-------------|
| 28 | 电商与直播电商数据对比 | | | | | | | | | | | | |
| 29 | | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 | 2023 | 2024 | 2025 | 2026 | 2027 | 2028 |
| 30 | 直播电商 | 196.4 | 1354.1 | 4437.5 | 12850 | 23615.1 | 35000 | 45657 | 57591.3 | 71848.57 | 89328.99 | 110940.8 | 137729.62 |
| 31 | 网上销售额 | 71751 | 90065 | 106324 | 117601 | 130884 | 137853 | 155064.083 | 170158.26 | 180134.23 | 198658.44 | 230273.47 | 276758.406 |
| 32 | | 0.002737244 | 0.015034697 | 0.041735638 | 0.109267778 | 0.180427707 | 0.25389364 | 0.294439558 | 0.3384573 | 0.3988613 | 0.4496612 | 0.4817785 | 0.497652888 |

图 7: 直播电商占电商总额的百分比预测数据

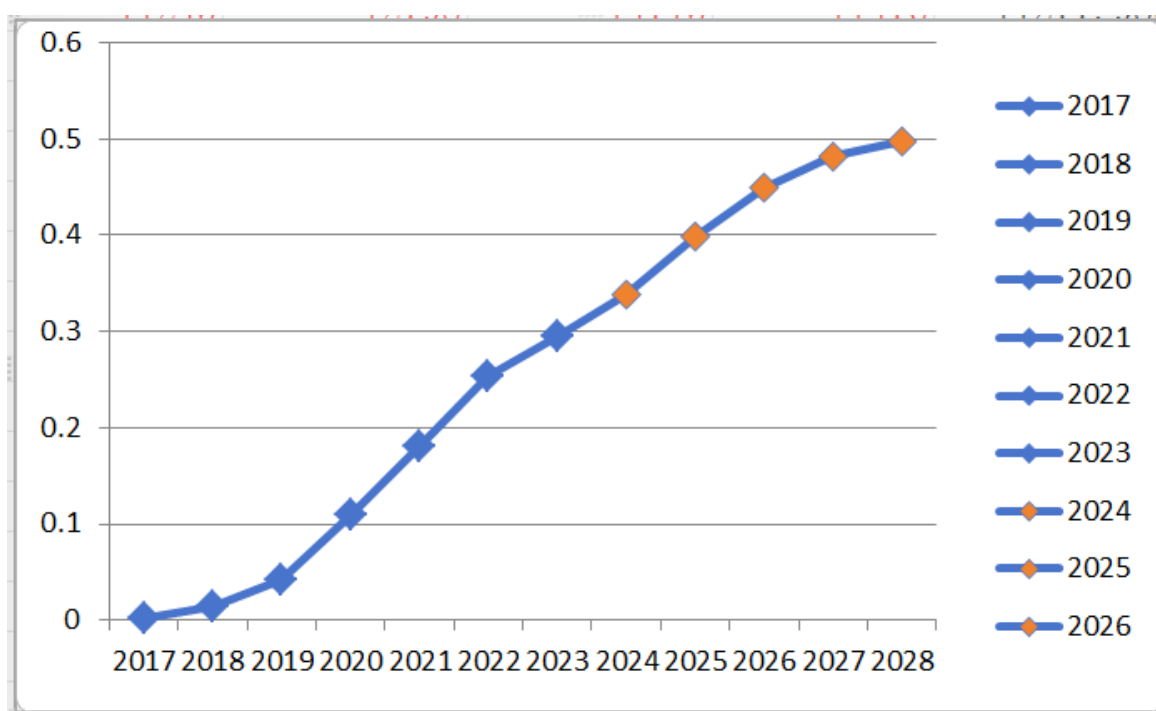


图 8: 直播电商占电商总额的百分比预测图

由图可以得出，在没有任何管控下，直播电商的份额增长呈一个逐步增长的态势，尽管电商的总体量在增大，但是直播电商的增长速度远高于其他电商。

§3.3 直播电商与头部主播的比较分析

本节讨论头部主播在直播电商中所占的市场份额与影响力，并尝试预测未来数年的比例变化。

3.3.1 头部主播的定义

头部主播是在网络直播平台上拥有极高人气和影响力的主播。各平台对头部主播的定义不一且非常模糊，难以定量分析。例如，《2019 快手直播生态报告》认为头部主播是收入位于前一万名的带货主播 [2]，而更多的行业报告仅仅提到了头部主播的概念，并没有给出可量化的数值，这给研究带来非常大的困难。

3.3.2 带货主播的分类

目前带货主播大致可以分为几类：头部主播，大多成立自己的公司，通过专业团队运营；机构签约主播；企业主播；个人主播等 [3]。

3.3.3 企业自播与达人播的市场占比分析与预测

根据 36 氪研究院的资料，主播领域马太效应明显，2.16% 的头部主播占据近 90% 的市场份额，腰部主播投产比很低 [4]。而信达证券的数据显示，2022 年企业自播的市场占有率超过 50% [5]。由于各大平台对于头部主播的定义不一，我们难以找到精确的头部主播的收入或销售额市场占比发展趋势。尽管如此，我们仍然可以通过企业自播与达人播的 5 年数据走向推断头部主播在带货主播中的市场份额。

由于可用的数据量非常少，我们可以建立一个简单的线性神经网络模型来对未来五年的发展做出预测。模型的训练数据集由 2018 年至 2023 年的市场占比数据组成，该数据反映了达人播市场份额的年度变化。

在本研究中，我们构建了一个简单的线性回归模型，以预测达人播在直播电商市场中的占比变化。模型的训练数据集由 2018 年至 2023 年的市场占比数据组成，该数据反映了达人播市场份额的年度变化。

首先对年份数据进行归一化处理，以消除由于数据量纲引起的影响，并提高模型训练的效率。归一化的公式如下：

$$X_{\text{normalized}} = \frac{X - \mu}{\sigma}$$

其中， X 是原始数据， μ 和 σ 分别是数据的均值和标准差。

定义一个线性模型 LinearModel，该模型仅包含一个线性层，用于学习输入年份和市场占比之间的线性关系。模型的前向传播函数定义了如何计算预测值。

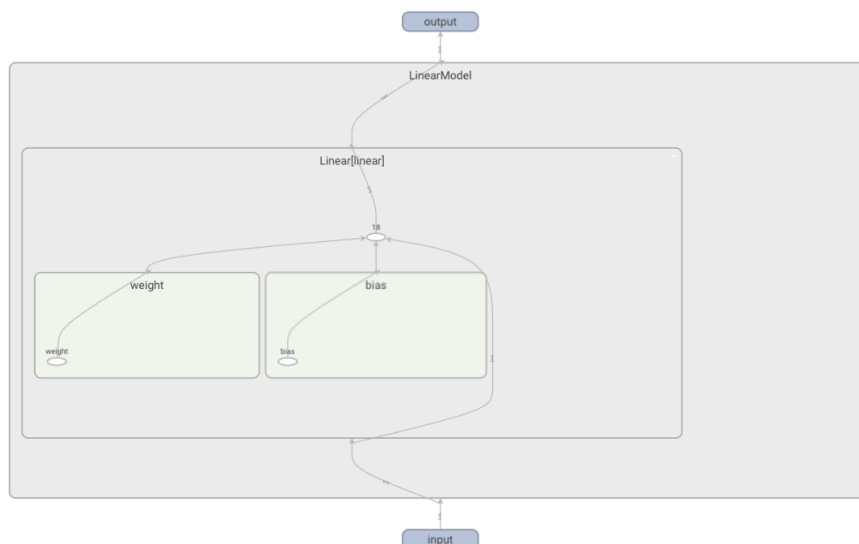


图 9: 神经网络结构

在模型训练过程中，使用均方误差（MSE）作为损失函数，它衡量的是模型预测值与真实值之间差异的平方的均值 [6]。优化器选择了 Adam，它是一种自适应学习率的优化算法，可以在训练过程中调整学习率，以改善模型的收敛速度和效果。

```

1 # 线性模型
2 class LinearModel(nn.Module):
3     def __init__(self):
4         super(LinearModel, self).__init__()
5         self.linear = nn.Linear(1, 1) # 线性层
6
7     def forward(self, x): # 前向传播
8         return self.linear(x)
9
10
11 model = LinearModel()
12
13 # 损失函数和优化器
14 criterion = nn.MSELoss() # 均方误差
15 optimizer = torch.optim.Adam(model.parameters(), lr=0.01) # Adam 优化器,
    学习率 0.01
16
17 # from torch.utils.tensorboard import SummaryWriter
18
19 # writer = SummaryWriter("runs/linear_model")
20 # sample_input = torch.tensor([[0.0]]) # 使用归一化后的输入形式
21 # writer.add_graph(model, sample_input)
22 # writer.close()
23
24 # 训练模型

```

```

25 epochs = 2000
26 loss_values = [] # 损失
27 for epoch in range(epochs):
28     model.train()
29     optimizer.zero_grad() # 清除旧的梯度
30
31     # 前向传播
32     outputs = model(years_normalized.unsqueeze(1))
33     loss = criterion(outputs, talent_broadcast_normalized.unsqueeze(1))
34
35     # 反向传播
36     loss.backward() # 计算梯度
37     optimizer.step() # 更新权重
38
39     # 记录损失值
40     loss_values.append(loss.item())
41
42     # if epoch % 100 == 0:
43     # print(f"Epoch: {epoch+1}/{epochs}, Loss: {loss.item()}")
44
45
46 # 预测未来5年的市场占比
47 future_years = torch.tensor([2024, 2025, 2026, 2027, 2028]).float()
48 future_years_normalized = (future_years - years.mean()) / years.std()
49
50 model.eval()
51 with torch.no_grad():
52     future_predictions =
53         model(future_years_normalized.unsqueeze(1)).flatten()
54     # 反归一化
55     future_predictions = (
56         future_predictions * talent_broadcast.std() +
57         talent_broadcast.mean()
58     )

```

训练完成后，使用模型对 2024 年至 2028 年的市场占比进行了预测。预测结果表明，达人播的市场占比有可能继续下降，这可能是由于企业自播的稳步增长和市场策略调整等因素的影响。

为了直观展示预测结果，使用 matplotlib 库绘制了达人播市场占比的历史数据和预测值的折线图。

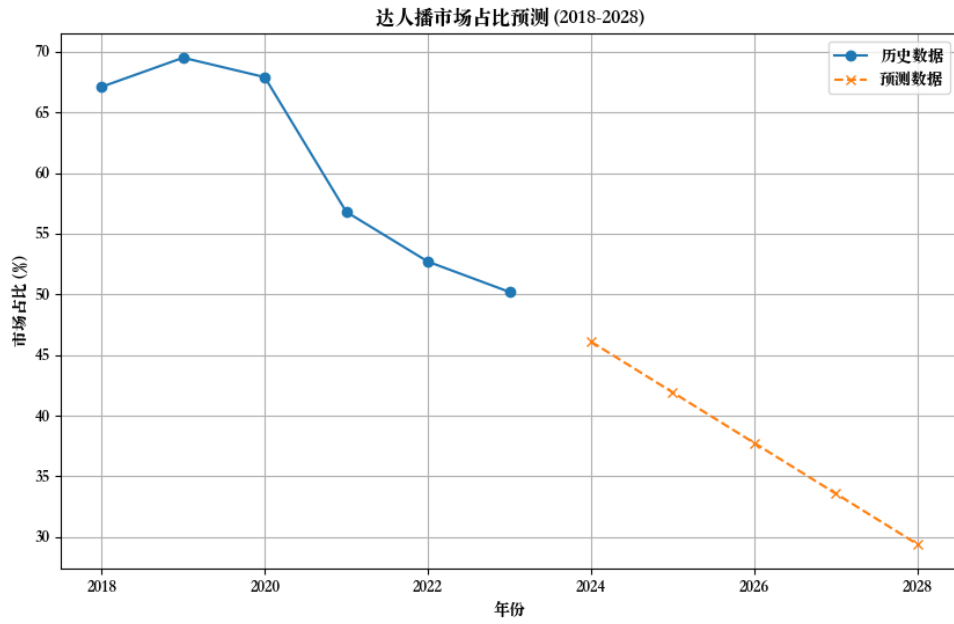


图 10: 预测结果

```

1 plt.figure(figsize=(10, 6))
2 plt.plot(years, talent_broadcast, label="历史数据", marker="o")
3 plt.plot(future_years, future_predictions, label="预测数据", marker="x",
4         linestyle="--")
5 plt.xlabel("年份")
6 plt.ylabel("市场占比 (%)")
7 plt.title("达人播市场占比预测 (2018-2028)")
8 plt.legend()
9 plt.grid(True)
10 plt.show()

```

通过以上分析可以看出，尽管达人播在早期占据了直播电商市场的主导地位，但随着时间的推移，企业自播逐渐展现出其在市场中的竞争力。

§3.4 实体店、直播电商和头部主播的份额对比

通过以上三个部分的分析和预测，根据比例关系可以得出以下数据结果。由图示关系可以清晰的了解实体店、直播电商和头部主播的份额情况。头部主播的市场份额逐渐下滑，但直播电商依然保持高速增长

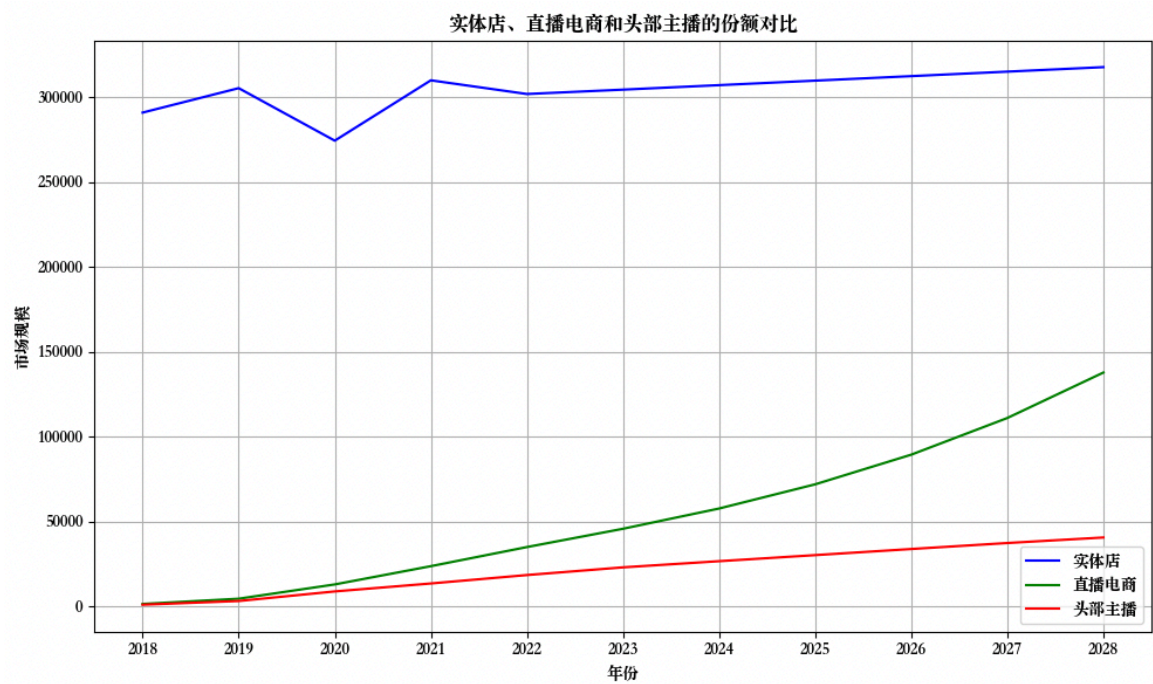


图 11: 实体店、直播电商和头部主播的份额对比图

§4 模拟和政策建议

根据我们的实证分析，可以发现头部主播的影响力增长放缓甚至有下降趋势，对实体店的影响有限。所以我们认为没有必要对头部主播进行过多限制。然而，根据我们的预测，到 2028 年直播电商的市场规模将达到实体店的 43%，增长迅速，可能影响线下零售的就业情况。因此，对于整体的直播电商行业进行定量限制是有意义的。

通过模拟 5% 到 50% 的抑制比例，并假设市场需求一定，即对直播电商的抑制规模会以 100% 的比例转移到实体店中。结果发现，直播电商的定量限制对实体店市场规模的影响作用不是非常显著。究其原因，直播电商的体量与实体店不能一概而论。

```

1 import matplotlib
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5 matplotlib.rcParams["font.family"] = "Songti SC"
6
7 physical_store_sales = np.array(
8     [
9         290922, 305325, 274380, 309939,
10        301880, 304448, 307101, 309754,
11        312407, 315060, 317713,
12    ]
13 )
14 live_streaming_sales = np.array(
15     [
16         1354, 4437, 12850, 23615,

```

```

17         34879,45657,57591,71848,
18         89328,110940,137729,
19     ]
20 )
21
22 years = len(physical_store_sales)
23 simulations = 1000 # 每个比例的模拟次数
24 reduction_percentages = np.linspace(0.05, 0.50, 11) #
    直播电商市场减少的比例，等差数列
25
26 # 存储模拟结果 二维0数组
27 simulation_results = np.zeros((len(reduction_percentages), years))
28
29 for i, reduction in enumerate(reduction_percentages):
30     for j in range(simulations):
31         # 每年减少的市场规模
32         reduction_amounts = live_streaming_sales * reduction
33         # 实体店市场增加的规模
34         adjusted_physical_store_sales = physical_store_sales +
            reduction_amounts
35         simulation_results[i] += adjusted_physical_store_sales
36     simulation_results[i] /= simulations
37
38 plt.figure(figsize=(12, 6))
39 for i, reduction in enumerate(reduction_percentages):
40     plt.plot(
41         range(2018, 2018 + years),
42         simulation_results[i],
43         label=f"{reduction*100:.1f}%",
44     )
45 plt.title("不同幅度的政策影响对实体店规模的反应")
46 plt.xlabel("年份")
47 plt.ylabel("实体店市场规模")
48 plt.legend()
49 plt.ylim(100000, 400000)
50 plt.grid(True)
51 plt.show()

```

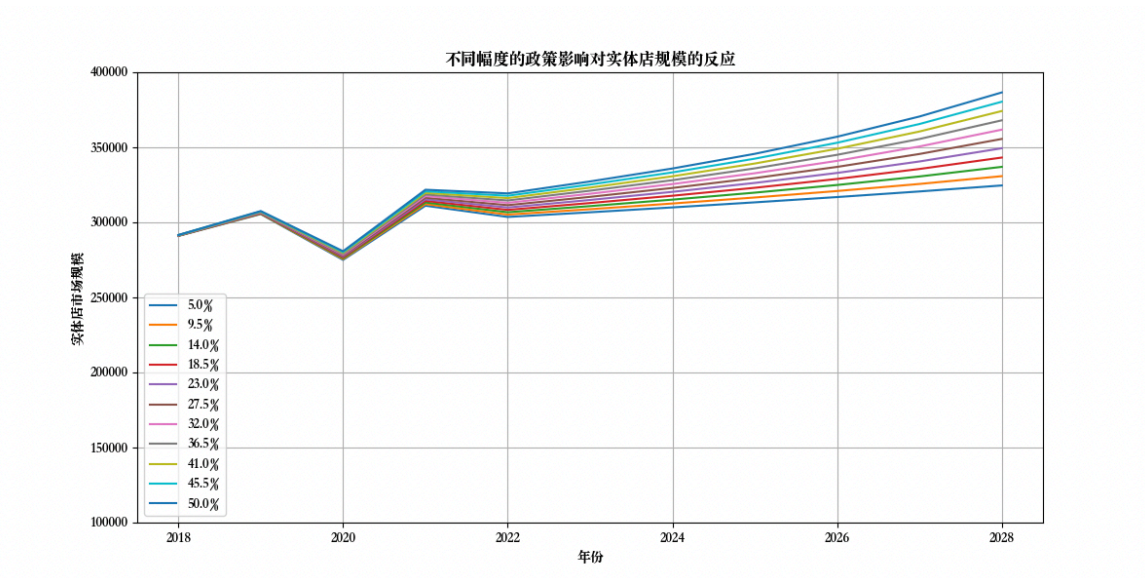


图 12: 不同幅度的政策影响对实体店规模的反应

§5 结语

本研究深入探讨了直播电商及其头部主播对传统实体店的影响。通过全面的数据分析和预测，我们发现直播电商市场规模呈现快速增长趋势，而且头部主播在直播电商领域的占比相对较高。尽管头部主播的影响力增长在未来几年有放缓的趋势，但直播电商整体上对实体店的冲击仍然显著。

通过模拟分析，我们发现对直播电商进行一定程度的定量限制对缓解实体店市场压力的效果有限。原因在于直播电商的总体市场规模尽管迅速增长，但与实体店相比仍有差距。直播电商市场的快速扩张更多地依赖于网络技术的发展和新型营销模式的创新，这使得其在短时间内迅速占据市场份额。

参考文献

- [1] 商务部. 商务部解读《“十四五”电子商务发展规划》, 11 2021.
- [2] 快手大数据研究院. 2019 快手直播生态报告, 12 2019.
- [3] 经济日报调研组. 高质量发展产业调研：直播带货透析, 1 2024.
- [4] 36 氪研究院. 2020 年中国直播电商行业研究报告, 12 2020.
- [5] 信达证券. 直播电商行业研究报告：分析直播电商产业链，把握行业发展浪潮, 3 2023.
- [6] Wikipedia contributors. Minimum mean square error, 2023.