Stock Market Comparisons and Predictions

James P. StClair

Northwest Missouri State University, Maryville MO 64468, USA
S555947@nwmissouri.edu

Abstract. I began this project trying to see what impact COVID may have had on several fields of work- namely, Insurance, Healthcare, and Delivery, and attempt to predict changes from there. I reviewed a small data set of several companies stock prices based on what they were at the closing of per day, from early 2020 to late 2022. I downloaded the csvs for each company, removed any missing or superfluous data, then trained and tested the full data remaining. Based on the results I received, I would believe that COVID still had a significant impact on each field, as the stock prices were very clearly affected at the beginning of 2020, but as time went on those changes either expanded or reversed. For Healthcare, their stock price rose and then continued to rise based on the data, and should continue to do so. Insurance on the other hand, had an initial decrease at the beginning of 2020 but then started to stabilize again by mid 2021, and should continue to rise from there, albeit slower than Healthcare. Delivery as a field was the most impacted I believe, as it initially saw a rather large rise as more people were staying home, but as it started approaching the end of 2022, it started going back down, in rare cases even lower than it had started, and should continue to lower a small amount further.

Keywords: COVID · Stock · Delivery · Insurance · Healthcare

1 Introduction

The domain that I have chosen for my project is Stock Market Predictions, as I want to look into the differences in Stock price of several fields, especially since COVID-19. I want to try to compare and contrast differences in the fields of Insurance, Healthcare, and Delivery services(such as Amazon), to show the differences, what trends exist, and make predictions for these companies.

I am planning on looking into Kaggle, Google Datasets, and AWS to try to find data-sets that would have what I am looking for- stock markets for these fields over time, including showing any impact during and since COVID.

1.1 Goals of this Project

I want to analyze the difference in some stock market fields since COVID, and show whether, as a whole, they are trending upwards since then, or downwards, and make predictions based on the data obtained. The fields I will be looking at are Insurance, Healthcare, and Delivery Services, to try to showcase some of the fields that would have a direct impact from COVID and its spread. I will want to show predictions for each field as a whole, and several individual companies, to show how they have declined or gone up since COVID.

1.2 Steps in the Project Process

First I would create a Github repo and Overleaf outline regarding this project. Next I will look over the possible data sources on hand, to try to find as clean of data as possible, and fitting what I am attempting to look for. I will look into similar projects on order to give myself some background information, and source anything that I use I will start cleaning and assessing the data I have pulled in order to present the information I am looking for, and only use the relevant data, keeping a note of every process I use to do so. I will use the finished data to run predictive modeling of the data, to try to see what the differences will look like further on I will bring everything together, as a final presentation to show what I did, what I predict, and how I think these fields will progress from here.

2 Key Components and Limitations

The key components of my approach and results will be the use of accurate data regarding stock prices of the companies in these fields, and using data analysis from there to show the trends for each company as well as each field as a whole. It will require accurate data, proper use of analytical skills and tools, and predictive modeling. Some limitations however is that it would not take into account many outside factors that could happen in the meantime, for the companies individually or another event such as COVID that could affect several companies at once. Another limitation I came across was that as much data as this data set had, there were few companies actually working in the fields I was looking for in this list.

3 Data Set

The data set I will be using, Stock Market Data(NASDAQ, NYSE, S and P500) was found in Kaggle, and was made by Paul Mooney. This data-set appears to be several csv's, 1 for each company as part of the 3 subheaders in the title. I used batch processing to go through the data, after I made sure to confirm which companies I would need data from. I am planning to use the names of the companies, the dates, and the stock price on the end of business day, in a separate chart for each company. I will be doing this for each company, in order to have viable data for each one, before working with this data both for each company, and for each field I will split them into as a whole(healthcare, insurance, and delivery). However, for any company that works in 2 or more fields(such as UNH working in both Healthcare and Insurance), I will list it under both fields. I have use the following companies for the previously named fields:

- Delivery
  - UPS
  - FDX
  - EBAY
  - AMZN
- Healthcare
  - LLY
  - UHS
  - HCA
  - JNJ
  - UNH
- Insurance
  - ALL
  - PGRE
  - BRO
  - CB
  - MET
  - UNH
  - TRV

The individual data, as well as each process for the individual companies and the fields as a whole, can be found here, on the Github Repository: CapstoneProject. https://github.com/jpstclair/CapstoneProject.git

3.1 Cleaning the Data

The data-set I had was already pretty well organized by company into many separate csvs, but it did also contain a large amount of data that for this project I did not need. The first thing I needed to do was remove the columns for the attributes I did not need, and then add into each one a new column simply to state which of the 3 fields I was looking into was. After that, I had 1 main attribute remaining for each field- which was Date. Each field also included several different individual companies that had data in this data-set, each labelled by their stock symbol(such as ALL for Allstate), as well as one last field labelled Average, which was the average of each of these previous fields. Any dates that had missing values were removed, as they would not be able to affect the data and would simply be clutter where it wasn't necessary. These were primarily removed by utilizing Excels own programs to remove rows that had missing values. This left me with roughly 299 dates for each company, with which I will attempt to plot both a line plot of the individual companies and the averages(shown in 3 figures below, labelled as each fields stock price from 2020 to 2022) to see if the stock prices appear to generally be going up or down, as well as applying that data to attempt to see if I can predict what will occur with that stock after the end of that period.

3.2 Analyzing the Data

After looking over the data for some time, I was able to find a few things. As a whole, most of these companies dipped down in early 2020, though Johnson and Johnson(Healthcare), Eli Lily and Co(Healthcare), Progressive(Insurance), Regeneron(Healthcare), and United Health Group Inc(Healthcare) are all exceptions to this, as they have all been generally rising up gradually since 2020. This is, of course accounting for the general oscillations that stock prices tend to have, as while there were a lot of ups and downs, the trend lines were absolutely positive for these companies throughout. On the whole, it appears that COVID may have affected some of these companies quite a bit. For example, delivery companies saw large decreases in early 2020, and massive rises in stock in late 2020, but as of late 2022 saw that stock come back down for almost every single one of them, though Fedex and UPS are slight outliers here in that they did not dip back down. Insurance companies and healthcare companies, on the other hand have been a bit more varied. They all generally had a small dip in 2020, but some went further down by 2022, some went back to roughly the same price by 2022, and some have kept increasing to well over where they started, showing that while COVID did have some affect on some of these companies, it didn't seem to affect them as much as delivery companies on the whole were affected. Specifically, I used line plots and to get these basic ideas in place, as well as looking at the line plots to get a sense of when they started going downwards or upwards. The source for this data can be looked at here:

https://www.kaggle.com/datasets/paultimothymooney/stock-market-data

4 Results and Predictions

The first thing that I did was split the data I had into training data and test data. After selecting the data involved, I ran a few different machine learning algorithms across the average results from each field as a whole, to better show the impact COVID had on those fields. I had immediately ruled out linear regression, as while it could have been good to use as a trend-line, with how often stocks oscillate even on the same week, I decided that it likely wouldn't be as useful for predicting the stocks in the future. As shown in the following figures, I ran each average through Polynomial Regression and Elastic Net, in order to try to find approximate estimates, or at least a solid footing for finding those estimates. Based on the results that were shown, I would believe that these trainings were at least mostly successful with showing how the average for each field will progress. Specifically, for the Insurance data, the Elasticnet generally fit better, with the R2 being .9309; for the Healthcare data, the Elasticnet also fit better, with the R2 being .9455; for the Delivery dataset, it was less accurate though still had an R2 of .7997. As such, I would believe any predictions made with these results to be mostly accurate, though with some possible variance based on events and how the individual companies may be perceived. I find it interesting that Healthcare started going down for a small amount of time at the beginning of COVID before generally rising up, while Insurance on the whole

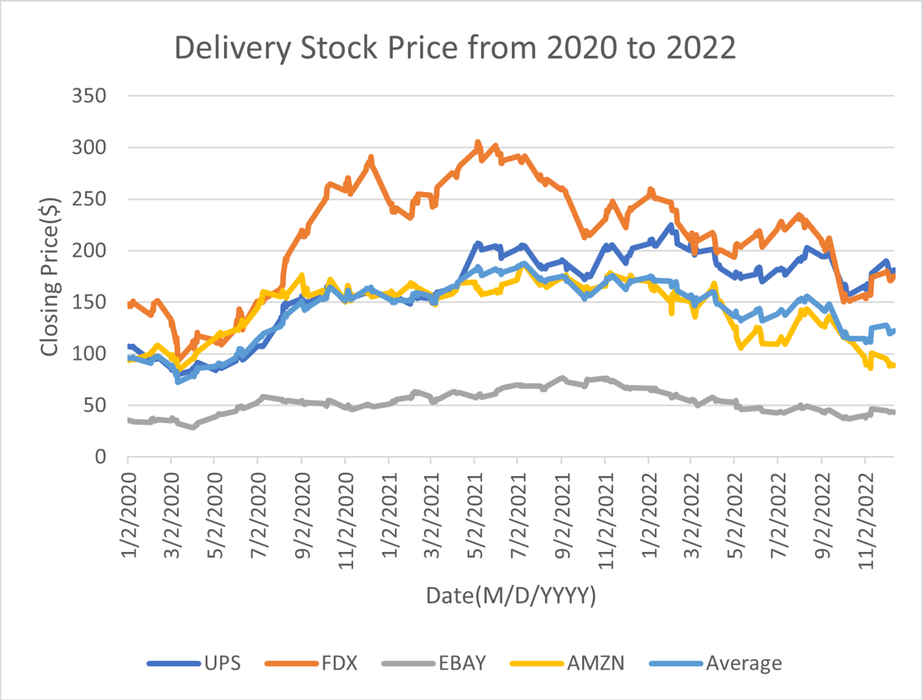


Fig. 1. Delivery Stock Prices

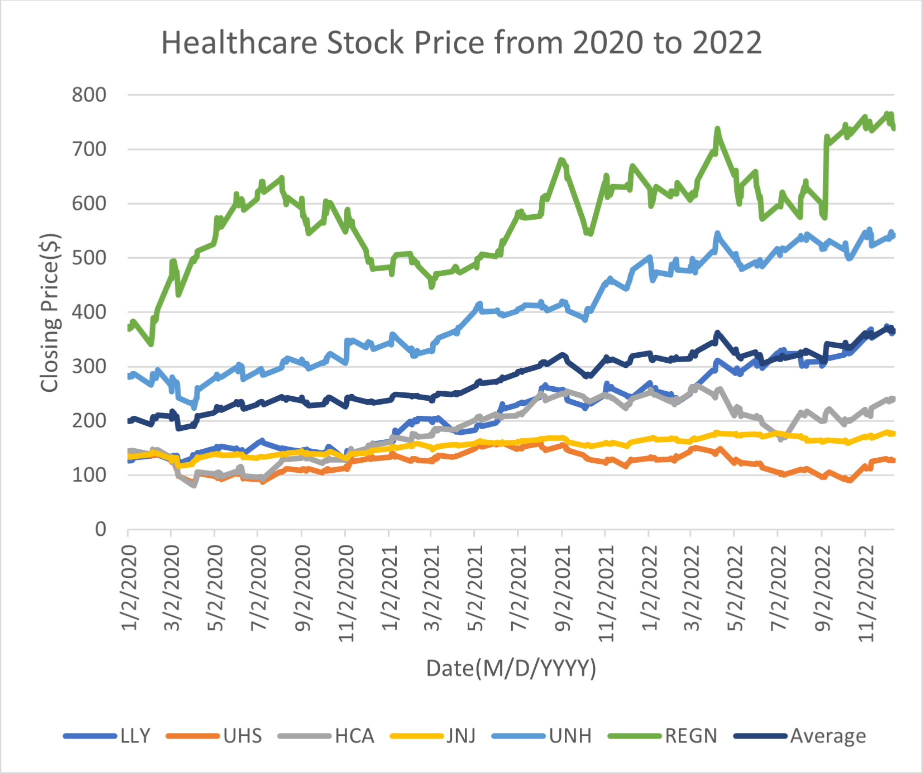


Fig. 2. Healthcare Stock Prices

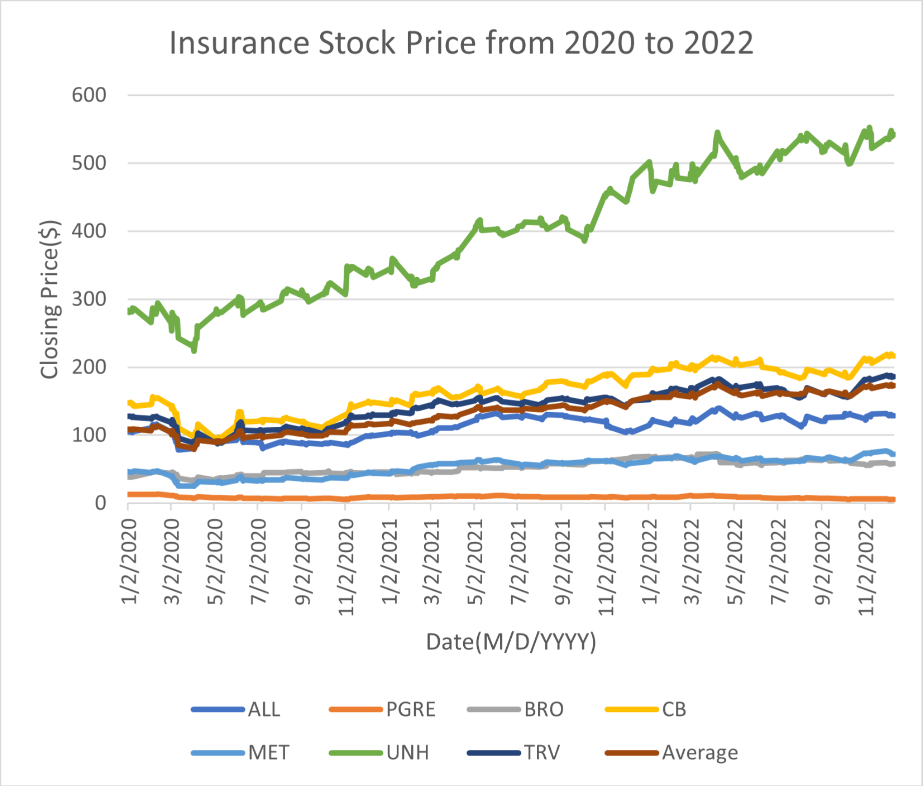


Fig. 3. Insurance Stock Prices

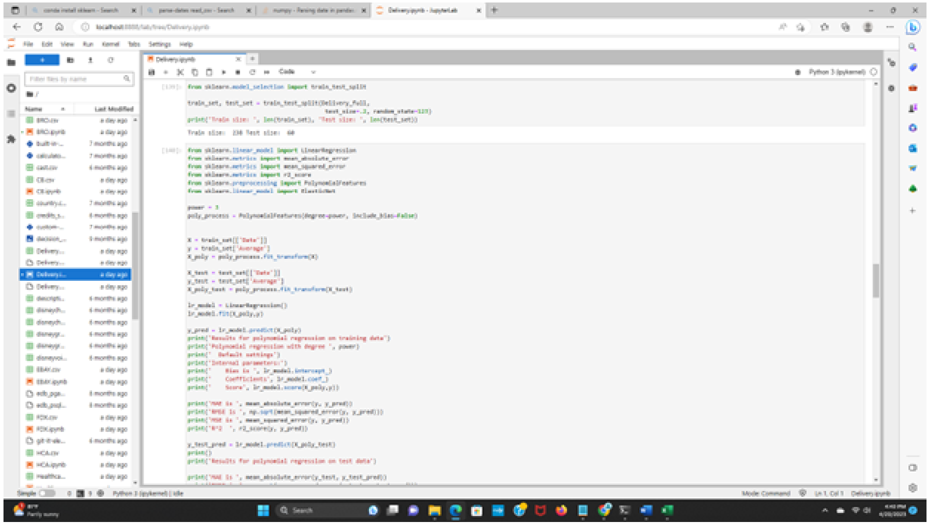
generally stayed around the same, or just went down a little bit across the entire 2020 to 2022 timeframe. At the same time, Delivery stock went up for the beginning of the worst of the COVID timeframe, but then as people started being more active and going out more often again it seemed to fall back down.

These trainings, as well as line plots for each individual company and the average for each field can be found here:

https://github.com/jpstclair/CapstoneProject.git

5 Conclusions and Future Work

I would believe that based on the results I received overall, that each of these fields were impacted in some way by COVID, as several companies either rose or fell at the beginning of COVID, and several companies that rose started going back down once COVID started to go down. It is interesting to me how these fields were affected, such as Healthcare stock prices rising as more attention was given to it and more was needed from it, but almost more interesting was the way that Delivery stock prices went up initially, but then did start to noticeably decay during mid 2022. For future work, there are 2 different things I would like to do. first, I would want to find more dataset in this same timeframe for more companies in these fields, to get a better idea for the whole of the impact on these fields. Second, I would also want to start looking into other fields as well, to see what impact may have been had on some of those fields as well, as there could be some fields that were impacted that, at first glance, we would think wouldn't be as affected by COVID.

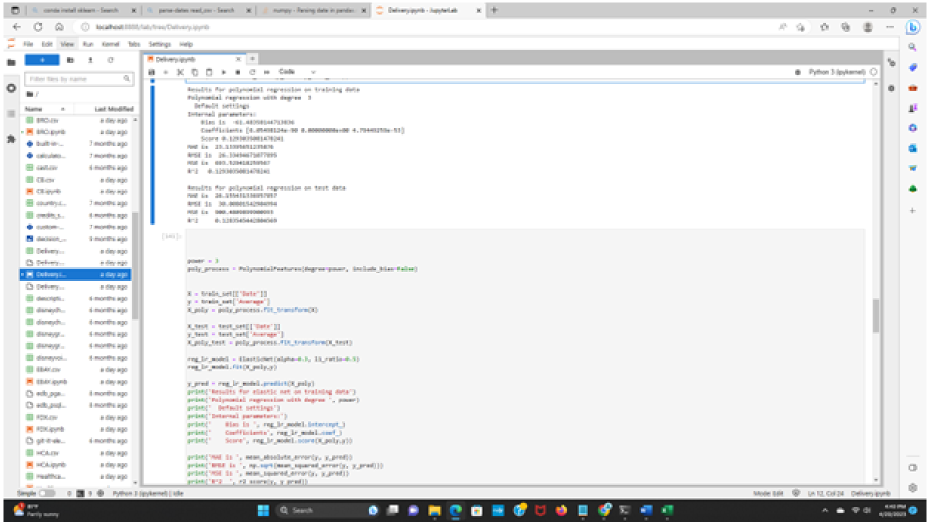


```
from sklearn.model_selection import train_test_split

train_set, test_set = train_test_split(Delivery_Full,
                                       test_size=.2, random_state=123)
print('Train size: ', len(train_set), 'Test size: ', len(test_set))
```

```
Train size:  238 Test size:  60
```

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import ElasticNet

power = 3
poly_process = PolynomialFeatures(degree=power, include_bias=False)


X = train_set[['Date']]
y = train_set['Average']
X_poly = poly_process.fit_transform(X)

X_test = test_set[['Date']]
y_test = test_set['Average']
X_poly_test = poly_process.fit_transform(X_test)

lr_model = LinearRegression()
lr_model.fit(X_poly,y)

y_pred = lr_model.predict(X_poly)
print('Results for polynomial regression on training data')
print('Polynomial regression with degree ', power)
print('  Default settings')
print('Internal parameters:')
print('    Bias is ', lr_model.intercept_)
print('    Coefficients', lr_model.coef_)
print('    Score', lr_model.score(X_poly,y))

print('MAE is ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2  ', r2_score(y, y_pred))

y_test_pred = lr_model.predict(X_poly_test)
print()
print('Results for polynomial regression on test data')

print('MAE is ', mean_absolute_error(y_test, y_test_pred))
```

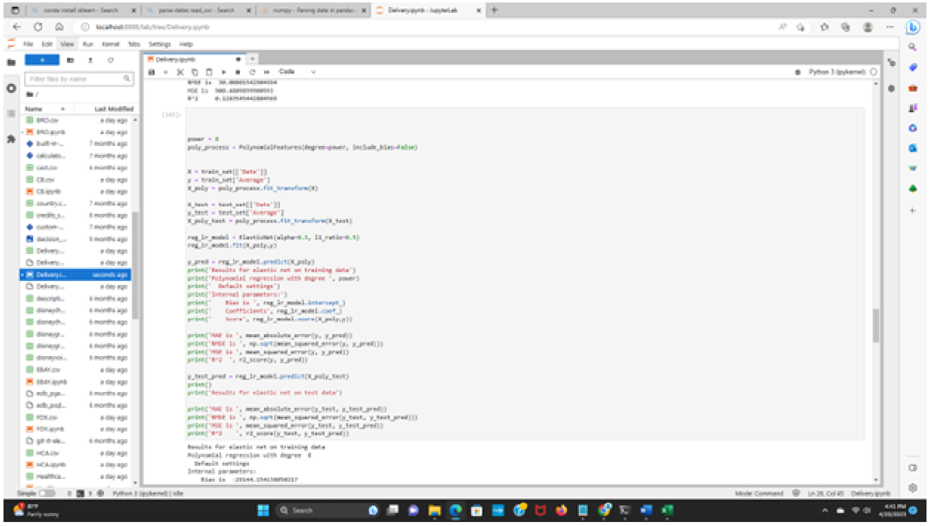
Fig. 4. Splitting the data and first part of Polynomial Regression - Delivery



```
Results for polynomial regression on training data
Polynomial regression with degree  3
  Default settings
Internal parameters:
    Bias is  -61.48358144713836
    Coefficients [4.05438124e-30 0.00000000e+00 4.75440250e-53]
    Score 0.1293035001478241
MAE is  23.133985812355876
RMSE is  26.33494671077095
MSE is  693.5204182509567
R^2   0.1293035001478241

Results for polynomial regression on test data
MAE is  26.155415336097057
RMSE is  30.00001542984954
MSE is  900.4800059008093
R^2     0.12835454420684569
```

```python
power = 3
poly_process = PolynomialFeatures(degree=power, include_bias=False)


X = train_set[['Date']]
y = train_set['Average']
X_poly = poly_process.fit_transform(X)

X_test = test_set[['Date']]
y_test = test_set['Average']
X_poly_test = poly_process.fit_transform(X_test)

reg_lr_model = ElasticNet(alpha=0.1, l1_ratio=0.5)
reg_lr_model.fit(X_poly,y)

y_pred = reg_lr_model.predict(X_poly)
print('Results for elastic net on training data')
print('Polynomial regression with degree ', power)
print('  Default settings')
print('Internal parameters:')
print('    Bias is ', reg_lr_model.intercept_)
print('    Coefficients', reg_lr_model.coef_)
print('    Score', reg_lr_model.score(X_poly,y))

print('MAE is ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2  ', r2_score(y, y_pred))
```

Fig. 5. Polynomial Regression and results - Delivery

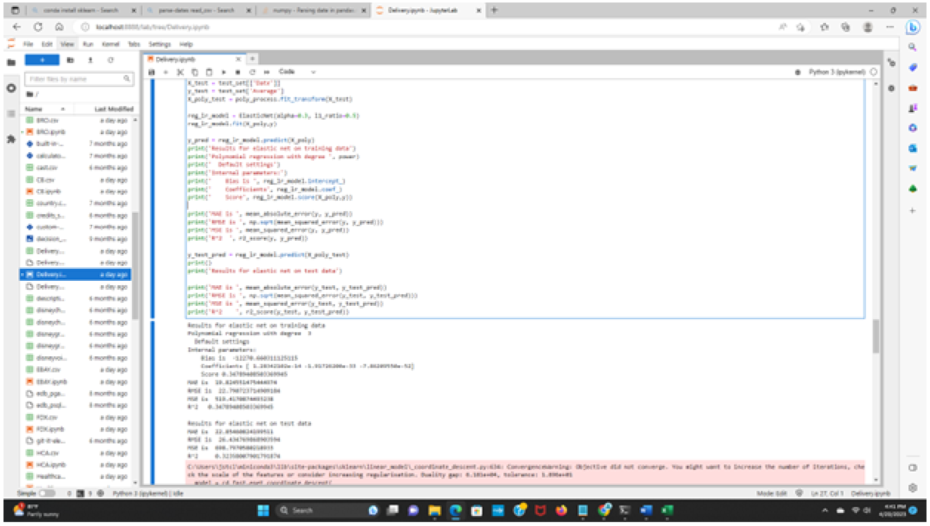


```
RMSE is  30.00001542984954
MSE is  900.4800059008093
R^2     0.12835454420684569
```

```python
power = 8
poly_process = PolynomialFeatures(degree=power, include_bias=False)


X = train_set[['Date']]
y = train_set['Average']
X_poly = poly_process.fit_transform(X)

X_test = test_set[['Date']]
y_test = test_set['Average']
X_poly_test = poly_process.fit_transform(X_test)

reg_lr_model = ElasticNet(alpha=0.1, l1_ratio=0.5)
reg_lr_model.fit(X_poly,y)

y_pred = reg_lr_model.predict(X_poly)
print('Results for elastic net on training data')
print('Polynomial regression with degree ', power)
print('  Default settings')
print('Internal parameters:')
print('    Bias is ', reg_lr_model.intercept_)
print('    Coefficients', reg_lr_model.coef_)
print('    Score', reg_lr_model.score(X_poly,y))

print('MAE is ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2  ', r2_score(y, y_pred))

y_test_pred = reg_lr_model.predict(X_poly_test)
print()
print('Results for elastic net on test data')

print('MAE is ', mean_absolute_error(y_test, y_test_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y_test, y_test_pred)))
print('MSE is ', mean_squared_error(y_test, y_test_pred))
print('R^2    ', r2_score(y_test, y_test_pred))
```

```
Results for elastic net on training data
Polynomial regression with degree  8
  Default settings
Internal parameters:
    Bias is  -25144.154158050217
```

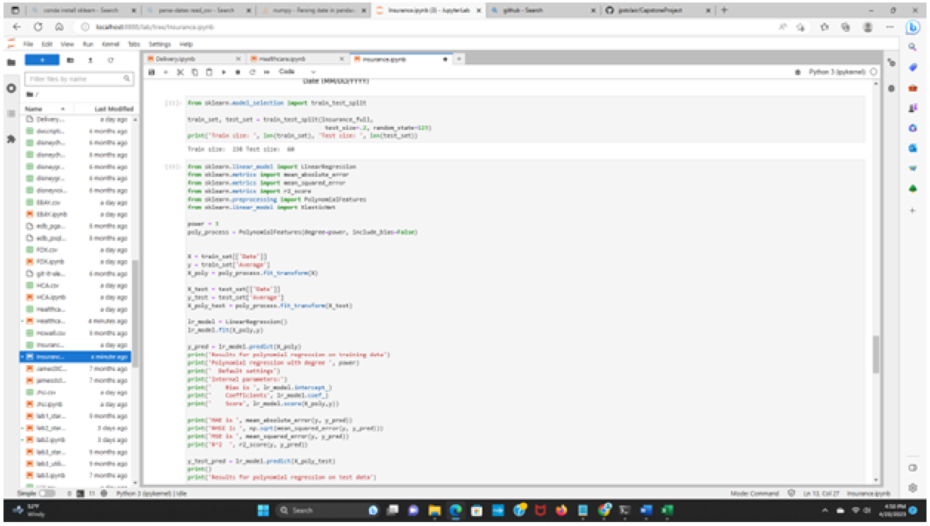
Fig. 6. First part of ElasticNet- Delivery



```
X_test = test_set[['Date']]
y_test = test_set['Average']
X_poly_test = poly_process.fit_transform(X_test)

reg_lr_model = ElasticNet(alpha=0.3, l1_ratio=0.5)
reg_lr_model.fit(X_poly,y)

y_pred = reg_lr_model.predict(X_poly)
print('Results for elastic net on training data')
print('Polynomial regression with degree ', power)
print('  Default settings')
print('Internal parameters:')
print('    Bias is ', reg_lr_model.intercept_)
print('    Coefficients', reg_lr_model.coef_)
print('    Score', reg_lr_model.score(X_poly,y))

print('MAE is ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2  ', r2_score(y, y_pred))

y_test_pred = reg_lr_model.predict(X_poly_test)
print()
print('Results for elastic net on test data')

print('MAE is ', mean_absolute_error(y_test, y_test_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y_test, y_test_pred)))
print('MSE is ', mean_squared_error(y_test, y_test_pred))
print('R^2    ', r2_score(y_test, y_test_pred))
```

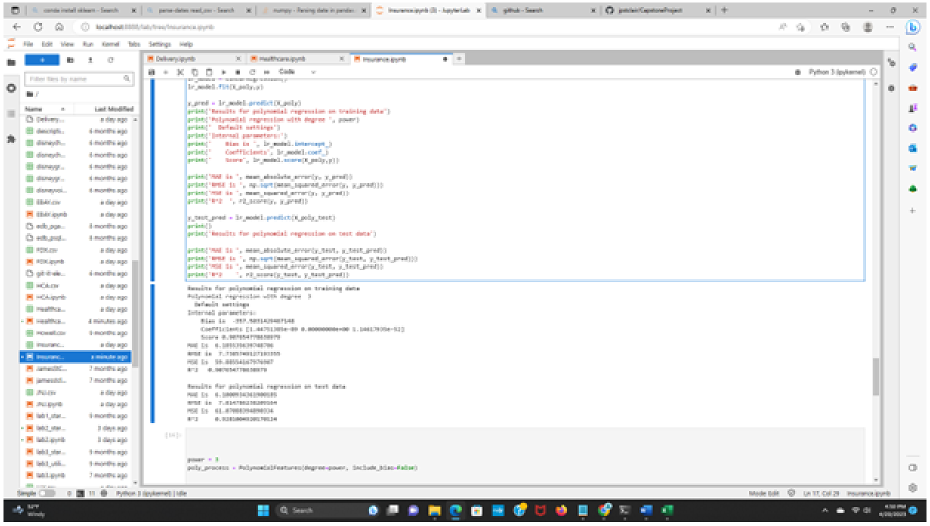
```
Results for elastic net on training data
Polynomial regression with degree  3
  Default settings
Internal parameters:
    Bias is  -11270.660311125115
    Coefficients [ 1.28342582e-14 -1.35726206e-33 -7.86289958e-52]
    Score 0.3478948858336945
MAE is  18.824551475444074
RMSE is  22.798723714909184
MSE is  519.4176074493238
R^2   0.3478948858336945

Results for elastic net on test data
MAE is  22.854008224199511
RMSE is  26.434769868903594
MSE is  698.7970580218015
R^2     0.3235000790179187
C:\Users\jstcl\miniconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:634: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations, che
ck the scale of the features or consider increasing regularization. Duality gap: 6.101e+04, tolerance: 1.096e+01
  model = cd_fast.enet_coordinate_descent(
```

Fig. 7. Second part of ElasticNet and Results- Delivery



```
[11]: from sklearn.model_selection import train_test_split

      train_set, test_set = train_test_split(insurance_full,
                                             test_size=.2, random_state=123)
      print('Train size: ', len(train_set), 'Test size: ', len(test_set))

      Train size:  238 Test size:  60

[13]: from sklearn.linear_model import LinearRegression
      from sklearn.metrics import mean_absolute_error
      from sklearn.metrics import mean_squared_error
      from sklearn.metrics import r2_score
      from sklearn.preprocessing import PolynomialFeatures
      from sklearn.linear_model import ElasticNet

      power = 3
      poly_process = PolynomialFeatures(degree=power, include_bias=False)


      X = train_set[['Date']]
      y = train_set['Average']
      X_poly = poly_process.fit_transform(X)

      X_test = test_set[['Date']]
      y_test = test_set['Average']
      X_poly_test = poly_process.fit_transform(X_test)

      lr_model = LinearRegression()
      lr_model.fit(X_poly,y)

      y_pred = lr_model.predict(X_poly)
      print('Results for polynomial regression on training data')
      print('Polynomial regression with degree ', power)
      print('  Default settings')
      print('Internal parameters:')
      print('    Bias is ', lr_model.intercept_)
      print('    Coefficients', lr_model.coef_)
      print('    Score', lr_model.score(X_poly,y))

      print('MAE is ', mean_absolute_error(y, y_pred))
      print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
      print('MSE is ', mean_squared_error(y, y_pred))
      print('R^2  ', r2_score(y, y_pred))

      y_test_pred = lr_model.predict(X_poly_test)
      print()
      print('Results for polynomial regression on test data')
```

Fig. 8. Splitting the data and first part of Polynomial Regression - Insurance

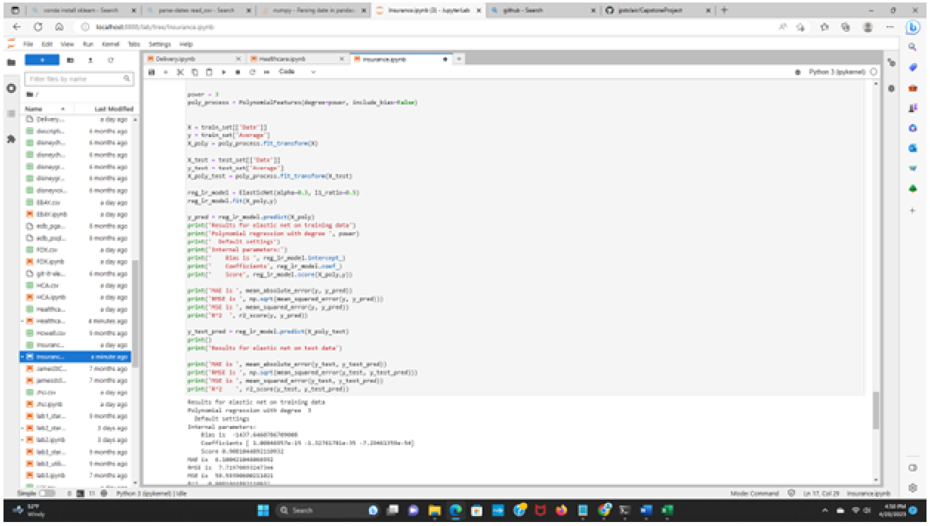


```
lr_model.fit(X_poly,y)

y_pred = lr_model.predict(X_poly)
print('Results for polynomial regression on training data')
print('Polynomial regression with degree ', power)
print('  Default settings')
print('Internal parameters:')
print('    Bias is ', lr_model.intercept_)
print('    Coefficients', lr_model.coef_)
print('    Score', lr_model.score(X_poly,y))

print('MAE is ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2  ', r2_score(y, y_pred))

y_test_pred = lr_model.predict(X_poly_test)
print()
print('Results for polynomial regression on test data')

print('MAE is ', mean_absolute_error(y_test, y_test_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y_test, y_test_pred)))
print('MSE is ', mean_squared_error(y_test, y_test_pred))
print('R^2    ', r2_score(y_test, y_test_pred))
```

```
Results for polynomial regression on training data
Polynomial regression with degree  3
  Default settings
Internal parameters:
    Bias is  -357.5031429407148
    Coefficients [1.44751385e-09 0.00000000e+00 1.14617935e-51]
    Score 0.987054778638979
MAE is  6.185535639748796
RMSE is  7.7385749127193355
MSE is  59.88554167976987
R^2   0.987054778638979

Results for polynomial regression on test data
MAE is  6.1880934361986535
RMSE is  7.814786238209104
MSE is  61.07088394898334
R^2     0.0281800692857014
```

```
[14]:

power = 3
poly_process = PolynomialFeatures(degree=power, include_bias=False)
```

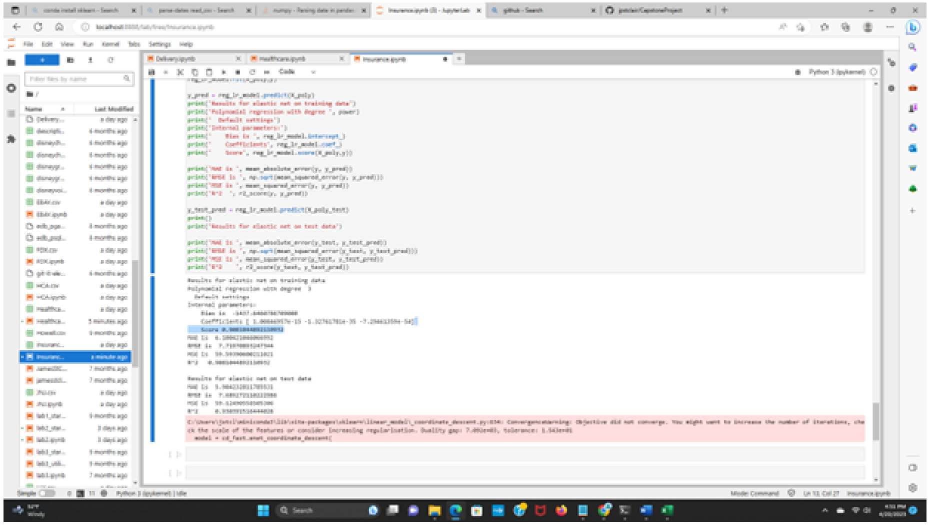
Fig. 9. Second part of Polynomial Regression and Results- Insurance



```
power = 3
poly_process = PolynomialFeatures(degree=power, include_bias=False)


X = train_set[['Date']]
y = train_set['Average']
X_poly = poly_process.fit_transform(X)

X_test = test_set[['Date']]
y_test = test_set['Average']
X_poly_test = poly_process.fit_transform(X_test)

reg_lr_model = ElasticNet(alpha=0.3, l1_ratio=0.5)
reg_lr_model.fit(X_poly,y)

y_pred = reg_lr_model.predict(X_poly)
print('Results for elastic net on training data')
print('Polynomial regression with degree ', power)
print('  Default settings')
print('Internal parameters:')
print('    Bias is ', reg_lr_model.intercept_)
print('    Coefficients', reg_lr_model.coef_)
print('    Score', reg_lr_model.score(X_poly,y))

print('MAE is ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2  ', r2_score(y, y_pred))

y_test_pred = reg_lr_model.predict(X_poly_test)
print()
print('Results for elastic net on test data')

print('MAE is ', mean_absolute_error(y_test, y_test_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y_test, y_test_pred)))
print('MSE is ', mean_squared_error(y_test, y_test_pred))
print('R^2    ', r2_score(y_test, y_test_pred))
```

```
Results for elastic net on training data
Polynomial regression with degree  3
  Default settings
Internal parameters:
    Bias is  -1437.6460786709008
    Coefficients [ 1.00846957e-15 -1.32761781e-35 -7.20461350e-54]
    Score 0.9881044892110032
MAE is  6.188421046068692
RMSE is  7.71570053247344
MSE is  59.53590688211621
R^2   0.9881044892110032
```

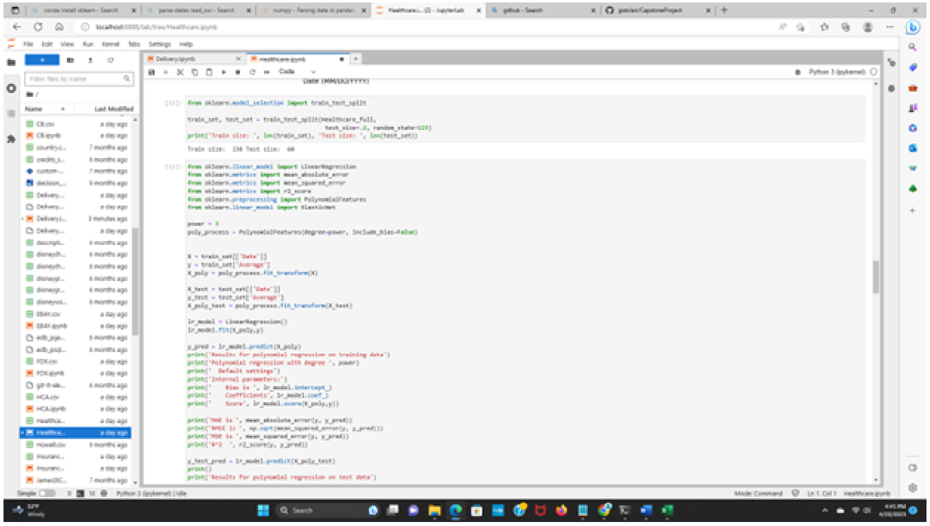
Fig. 10. First part of ElasticNet- Insurance



```
y_pred = reg_lr_model.predict(X_poly)
print('Results for elastic net on training data')
print('Polynomial regression with degree ', power)
print('  Default settings')
print('Internal parameters:')
print('    Bias is ', reg_lr_model.intercept_)
print('    Coefficients', reg_lr_model.coef_)
print('    Score', reg_lr_model.score(X_poly,y))

print('MAE is ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2  ', r2_score(y, y_pred))

y_test_pred = reg_lr_model.predict(X_poly_test)
print()
print('Results for elastic net on test data')

print('MAE is ', mean_absolute_error(y_test, y_test_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y_test, y_test_pred)))
print('MSE is ', mean_squared_error(y_test, y_test_pred))
print('R^2    ', r2_score(y_test, y_test_pred))
```

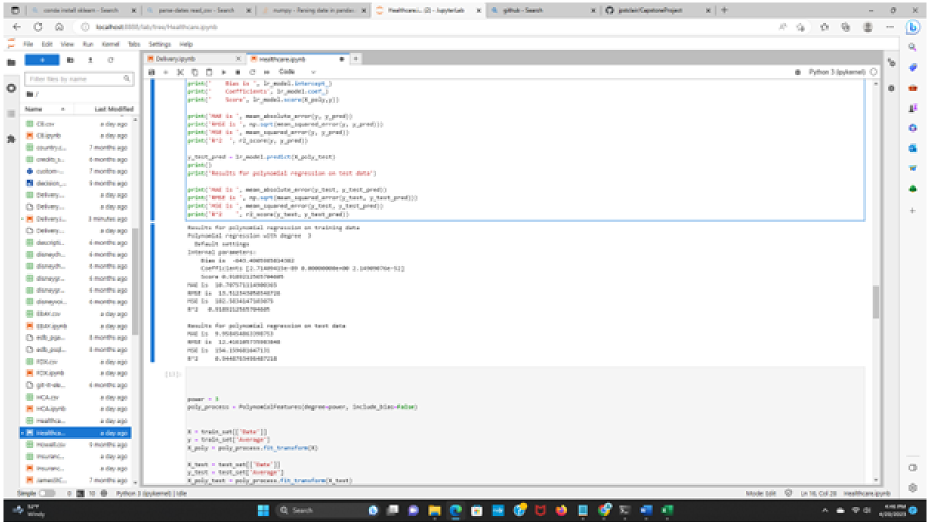
```
Results for elastic net on training data
Polynomial regression with degree  3
  Default settings
Internal parameters:
    Bias is  -3457.0460786709008
    Coefficients [ 1.00866957e-15 -1.32761781e-35 -7.29461359e-54]
    Score 0.0001044892110952
MAE is  6.100621066066992
RMSE is  7.7107889534794
MSE is  59.59390606211021
R^2   0.0001044892110952

Results for elastic net on test data
MAE is  5.904231011785531
RMSE is  7.689271316022098
MSE is  59.12490558505306
R^2     0.558591514444028
C:\Users\jstcl\miniconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:634: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations, che
ck the scale of the features or consider increasing regularization. Duality gap: 7.092e+03, tolerance: 5.543e+01
  model = cd_fast.enet_coordinate_descent(
```

Fig. 11. Second part of ElasticNet and results- Insurance



```
[11]: from sklearn.model_selection import train_test_split

      train_set, test_set = train_test_split(Healthcare_full,
                                             test_size=.2, random_state=123)
      print('Train size: ', len(train_set), 'Test size: ', len(test_set))

      Train size:  238 Test size:  60

[12]: from sklearn.linear_model import LinearRegression
      from sklearn.metrics import mean_absolute_error
      from sklearn.metrics import mean_squared_error
      from sklearn.metrics import r2_score
      from sklearn.preprocessing import PolynomialFeatures
      from sklearn.linear_model import ElasticNet

      power = 3
      poly_process = PolynomialFeatures(degree=power, include_bias=False)


      X = train_set[['Date']]
      y = train_set['Average']
      X_poly = poly_process.fit_transform(X)

      X_test = test_set[['Date']]
      y_test = test_set['Average']
      X_poly_test = poly_process.fit_transform(X_test)

      lr_model = LinearRegression()
      lr_model.fit(X_poly,y)

      y_pred = lr_model.predict(X_poly)
      print('Results for polynomial regression on training data')
      print('Polynomial regression with degree ', power)
      print('  Default settings')
      print('Internal parameters:')
      print('    Bias is ', lr_model.intercept_)
      print('    Coefficients', lr_model.coef_)
      print('    Score', lr_model.score(X_poly,y))

      print('MAE is ', mean_absolute_error(y, y_pred))
      print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
      print('MSE is ', mean_squared_error(y, y_pred))
      print('R^2  ', r2_score(y, y_pred))

      y_test_pred = lr_model.predict(X_poly_test)
      print()
      print('Results for polynomial regression on test data')
```

Fig. 12. Splitting the data and first part of Polynomial Regression - Healthcare

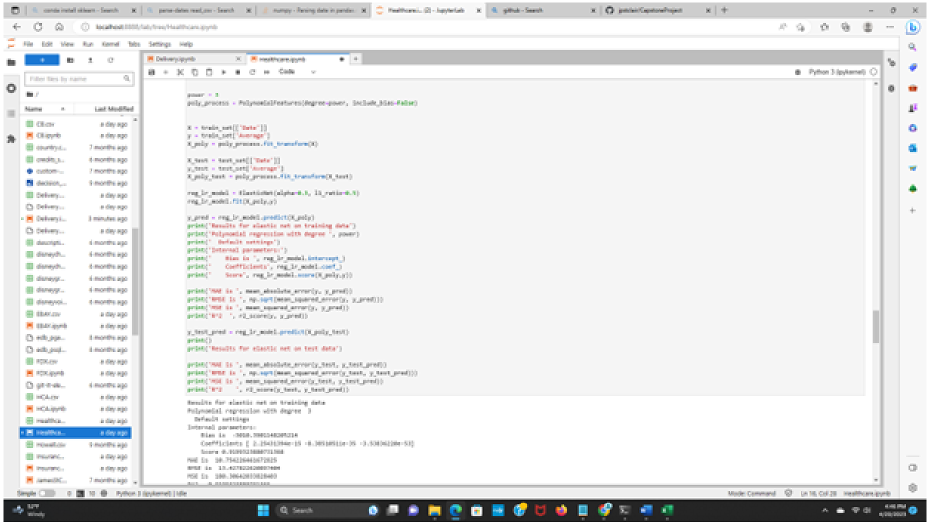


```
print('    Bias is ', lr_model.intercept_)
print('    Coefficients', lr_model.coef_)
print('    Score', lr_model.score(X_poly,y))

print('MAE is ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2  ', r2_score(y, y_pred))

y_test_pred = lr_model.predict(X_poly_test)
print()
print('Results for polynomial regression on test data')

print('MAE is ', mean_absolute_error(y_test, y_test_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y_test, y_test_pred)))
print('MSE is ', mean_squared_error(y_test, y_test_pred))
print('R^2     ', r2_score(y_test, y_test_pred))
```

```
Results for polynomial regression on training data
Polynomial regression with degree  3
  Default settings
Internal parameters:
    Bias is  -845.4985985834582
    Coefficients [2.71489421e-09 0.00000000e+00 2.14989076e-51]
    Score 0.9189212565784605
MAE is  10.707571114900365
RMSE is  13.512343058548728
MSE is  182.58341471583075
R^2   0.9189212565784605

Results for polynomial regression on test data
MAE is  9.958454863398753
RMSE is  12.416185735983848
MSE is  154.15968144713
R^2     0.0448765498487218
```

```
[13]:

power = 3
poly_process = PolynomialFeatures(degree=power, include_bias=False)


X = train_set[['Date']]
y = train_set['Average']
X_poly = poly_process.fit_transform(X)

X_test = test_set[['Date']]
y_test = test_set['Average']
X_poly_test = poly_process.fit_transform(X_test)
```

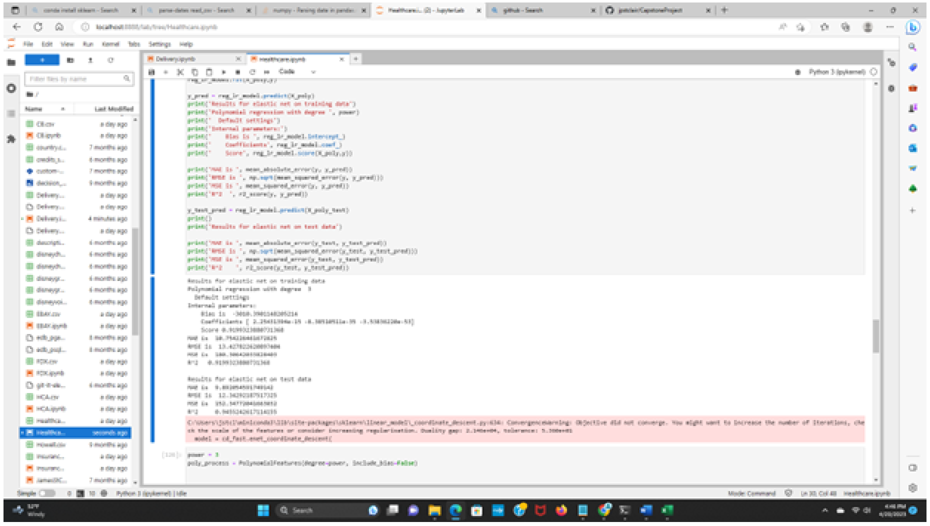
Fig. 13. Second part of Polynomial Regression and Results- Healthcare



```
power = 3
poly_process = PolynomialFeatures(degree=power, include_bias=False)


X = train_set[['Date']]
y = train_set['Average']
X_poly = poly_process.fit_transform(X)

X_test = test_set[['Date']]
y_test = test_set['Average']
X_poly_test = poly_process.fit_transform(X_test)

reg_lr_model = ElasticNet(alpha=0.3, l1_ratio=0.9)
reg_lr_model.fit(X_poly,y)

y_pred = reg_lr_model.predict(X_poly)
print('Results for elastic net on training data')
print('Polynomial regression with degree ', power)
print('  Default settings')
print('Internal parameters:')
print('    Bias is ', reg_lr_model.intercept_)
print('    Coefficients', reg_lr_model.coef_)
print('    Score', reg_lr_model.score(X_poly,y))

print('MAE is ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2  ', r2_score(y, y_pred))

y_test_pred = reg_lr_model.predict(X_poly_test)
print()
print('Results for elastic net on test data')

print('MAE is ', mean_absolute_error(y_test, y_test_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y_test, y_test_pred)))
print('MSE is ', mean_squared_error(y_test, y_test_pred))
print('R^2     ', r2_score(y_test, y_test_pred))
```

```
Results for elastic net on training data
Polynomial regression with degree  3
  Default settings
Internal parameters:
    Bias is  -3058.3985148285214
    Coefficients [ 2.25431394e-15 -8.38510511e-35 -3.53836228e-53]
    Score 0.9199523888731368
MAE is  10.754226441672825
RMSE is  13.427822628897404
MSE is  180.30642033828403
```

Fig. 14. First part of ElasticNet- Healthcare



```
reg_lr_model.fit(X_poly,y)

y_pred = reg_lr_model.predict(X_poly)
print('Results for elastic net on training data')
print('Polynomial regression with degree ', power)
print('  Default settings')
print('Internal parameters:')
print('    Bias is ', reg_lr_model.intercept_)
print('    Coefficients', reg_lr_model.coef_)
print('    Score', reg_lr_model.score(X_poly,y))

print('MAE is ', mean_absolute_error(y, y_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y, y_pred)))
print('MSE is ', mean_squared_error(y, y_pred))
print('R^2  ', r2_score(y, y_pred))

y_test_pred = reg_lr_model.predict(X_poly_test)
print()
print('Results for elastic net on test data')

print('MAE is ', mean_absolute_error(y_test, y_test_pred))
print('RMSE is ', np.sqrt(mean_squared_error(y_test, y_test_pred)))
print('MSE is ', mean_squared_error(y_test, y_test_pred))
print('R^2    ', r2_score(y_test, y_test_pred))
```

```
Results for elastic net on training data
Polynomial regression with degree  3
  Default settings
Internal parameters:
    Bias is  -3010.3901148205214
    Coefficients [ 2.25431394e-15 -8.38910511e-35 -3.53836220e-53]
    Score 0.9199323888731368
MAE is  10.754226461672825
RMSE is  13.427822620097404
MSE is  180.30042035820469
R^2   0.9199323888731368

Results for elastic net on test data
MAE is  9.893054501749142
RMSE is  12.342921875171325
MSE is  152.34772044065852
R^2     0.9495242617114155
C:\Users\jstcl\miniconda3\lib\site-packages\sklearn\linear_model\_coordinate_descent.py:634: ConvergenceWarning: Objective did not converge. You might want to increase the number of iterations, che
ck the scale of the features or consider increasing regularization. Duality gap: 2.140e+04, tolerance: 5.398e+01
  model = cd_fast.enet_coordinate_descent(
```

```
[128]: power = 3
       poly_process = PolynomialFeatures(degree=power, include_bias=False)
```

Fig. 15. Second part of ElasticNet and results- Healthcare

References

1. Mehrotra, N.: The aftermath and impact of covid-19 on stock markets, https://www.forbes.com/sites/theyec/2023/02/10/the-aftermath-and-impact-of-covid-19-on-stock-markets/?sh=4bdb943ec120
2. Mooney, P.: Tstock market data(nasdaq, nyse, sp500), https://www.kaggle.com/datasets/paultimothymooney/stock-market-data
3. Wang, K.M., Lee, Y.M.: Are life insurance futures a safe haven during covid-19? Finance Innov. 2023 **9**(1), 13 (2023). https://doi.org/10.1186/s40854-022-00411-z

[1]
[3] [2]