Lyft Case Study

Harshel Bahl

1. Case Study Prompt

Pretend you're the pricing product manager for Lyft's ride-scheduling feature, and you're launching a new city like Toledo, Ohio.

The prevailing rate that people are used to paying for rides from the airport to downtown (either direction, one way) is 25 dollars. The prevailing wage that drivers are used to earning for this trip is 19 dollars.

You launch with exactly this price: 25 dollars per ride charged to the rider, 19 dollars per ride paid to the driver. It turns out only 60 or so of every 100 rides requested are finding a driver at this price.

(While there is more than one route to think about in Toledo, for the sake of this exercise you can focus on this one route.)

Here's your current unit economics for each side:

Drivers: Customer acquisition cost (CAC) of a new driver is between 400 dollars - 600 dollars. CAC is sensitive to the rate of acquisition since channels are only so deep.

At the prevailing wage, drivers have a 5% monthly churn rate and complete 100 rides / month

Riders: CAC of a new rider is 10 dollars to 20 dollars (similar to driver CAC it's sensitive to the rate of acquisition, since existing marketing channels are only so deep) Each rider requests 1 ride / month on average

Churn is interesting: riders who don't experience a "failed to find driver" event churn at 10% monthly, but riders who experience one or more "failed to find driver" events churn at 33% monthly

You've run one pricing experiment so far: when you reduced Lyft's take from 6 dollars/ride to 3 dollars/ride across the board for a few weeks, match rates rose nearly instantly from 60% to roughly 93%.

Your task is to maximize the company's net revenue (the difference between the amount riders pay and the amount Lyft pays out to drivers) for this route in Toledo for the next 12 months. Let's assume that you cannot charge riders more than the prevailing rate.

The core question is: how much more or less do you pay drivers per trip (by changing Lyft's take)? Your goal is to maximize net revenue for the next 12 months on this route.

2. Study Breakdown

2.1. Trip Economics

- $25 price for trip (revenue)
- $19 cost as wage to driver for trip
- $6 lyft's current take
- 60% of ride requests are finding a driver at this baseline

2.2. Riders

- Monthly churn probability if 'failed to find driver' = 33%
- Monthly churn probability if not 'failed to find driver' = 10%
- Requests 1 ride per month on average
- CAC for new rider: $10 - 20$

2.3. Drivers

- $19 earned per ride
- Churns at 5% monthly at current wage
- Completes 100 rides per month on average
- CAC for new driver: 400 dollars - 600 dollars

2.4. Initial Price Experiment

- 19 dollars to 22 dollars increase in driver wage also increased ride match rates from 60% to 93%

2.5. Study Aim

- Maximize company's net revenue (trip price - driver wage) by adjusting Lyft's take
- Assumption 1: Trip price cannot be increased
- Assumption 2: A new route cannot be utilized
- Assumption 3: Strategy to be played out over 12 months
- Assumption 4: Driver churn increases from 0% to 100% as wage decreases from 25 dollars to 0 dollars, with a wage of $19 seeing a churn rate of 5%
- Assumption 5: A match rate of 0% is achieved as driver wage tends to 0
- Assumption 6: There are exactly the required numbers of driver's to service riders and no surplus

3. Initial Analysis on Trip Economics

3.1. Current Net Revenue

For simplifying calculations 100 rides are used below, but the way this model scales will need to be considered.

Since 60% of ride requests find a match => 60 successful trips, 40 failed trips:

$$R = 60 * 6 = 360$$

where R is monthly net revenue in dollars per 100 trips.

3.2. Current Monthly Rider Churn

$$TRC = 60 * 0.1 + 40 * 0.33 = \ \sim 19$$

where TRC is monthly total rider churn in number of riders.

With a CAC of 10-20 dollars, this equals 190-380 dollars required to replace 19 riders every month per 100 rides.

3.3. Current Monthly Driver Churn

$$TDC = 0.05 * TD$$

where TDC is monthly total driver churn in number of drivers, and TD is total drivers.

With a CAC of 400-600 dollars, this gives 20*TD - 30*TD$ required to replace the riders every month that churn.

3.4. Analysis on Initial Model

- Since monthly net revenue depends on the match rate and lyft's take, maximising the match rate and lyft's take would maximise revenue.
- Since match rate is dependent on driver wage, finding the optimal driver wage to maximise match rate is crucial.
- However, increasing driver wage decreases lyft's take decreasing net revenue, so this will need to be optimised with a model using assumption 4 above.
- It will be useful to optimise the net revenue for 1 month, then consider expanding the model for a rolling 12 months, as this will likely result in adjusting the optimisations to account for the wildly differing CACs between drivers and riders.
- It is also important to model the fact that 1 driver can service 100 different riders on average per month since each driver completes 100 rides, while riders book 1 per month on average.

4. Generate Models

4.1. Model the Monthly Net Revenue

$$R = Match\ Rate * Lyft\ Take = f(DWage) * (Trip\ Price - DWage) = f(DWage) * (25 - DWage)$$

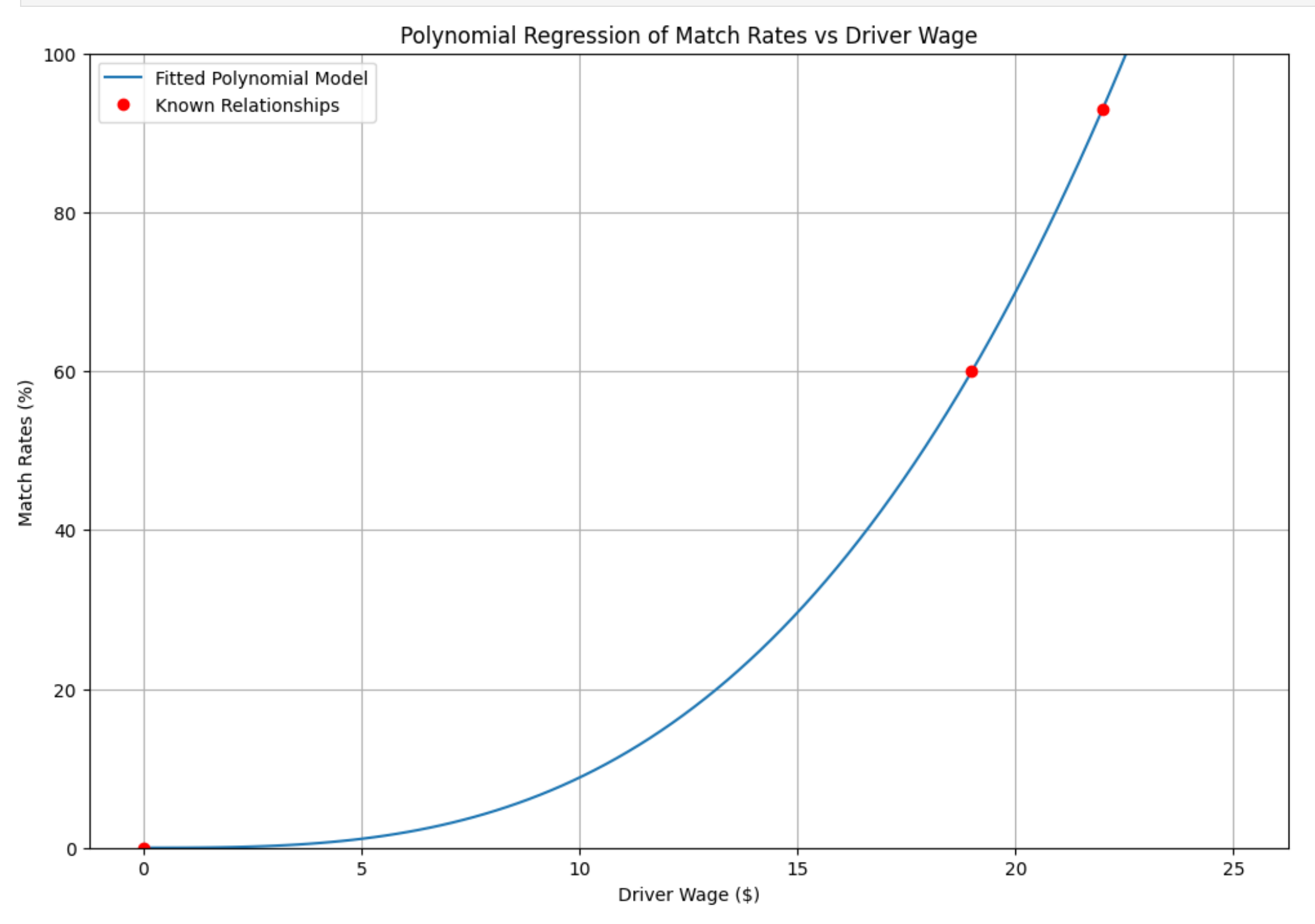
where R is the expected net revenue of a trip in dollars, DWage is the driver's wage, f(DWage) is some function of Driver's wage that will be modelled using assumptions.

Therefore, the monthly net revenue would simply be the number of rides completed * R.

4.2. Model for Optimizing Monthly Net Revenue

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize_scalar
```

```
x = np.array([19, 22])
y = np.array([60, 93])

A = np.vstack([x**3, x**2]).T
coefficients = np.linalg.lstsq(A, y, rcond=None)[0]
a, b = coefficients

xp = np.linspace(0, 25, 100)
yp = a*xp**3 + b*xp**2

plt.figure(figsize=(12, 8))
plt.plot(xp, yp, label='Fitted Polynomial Model')
plt.plot(x, y, 'ro', label='Known Relationships')
plt.plot(0, 0, 'ro')
plt.ylim(0, 100)
plt.xlabel('Driver Wage ($)')
plt.ylabel('Match Rates (%)')
plt.title('Polynomial Regression of Match Rates vs Driver Wage')
plt.legend()
plt.grid(True)
plt.show()

print('The polynomial model is: y = {}x^3 + {}x^2'.format(round(a, 4), round(b, 4)))
```

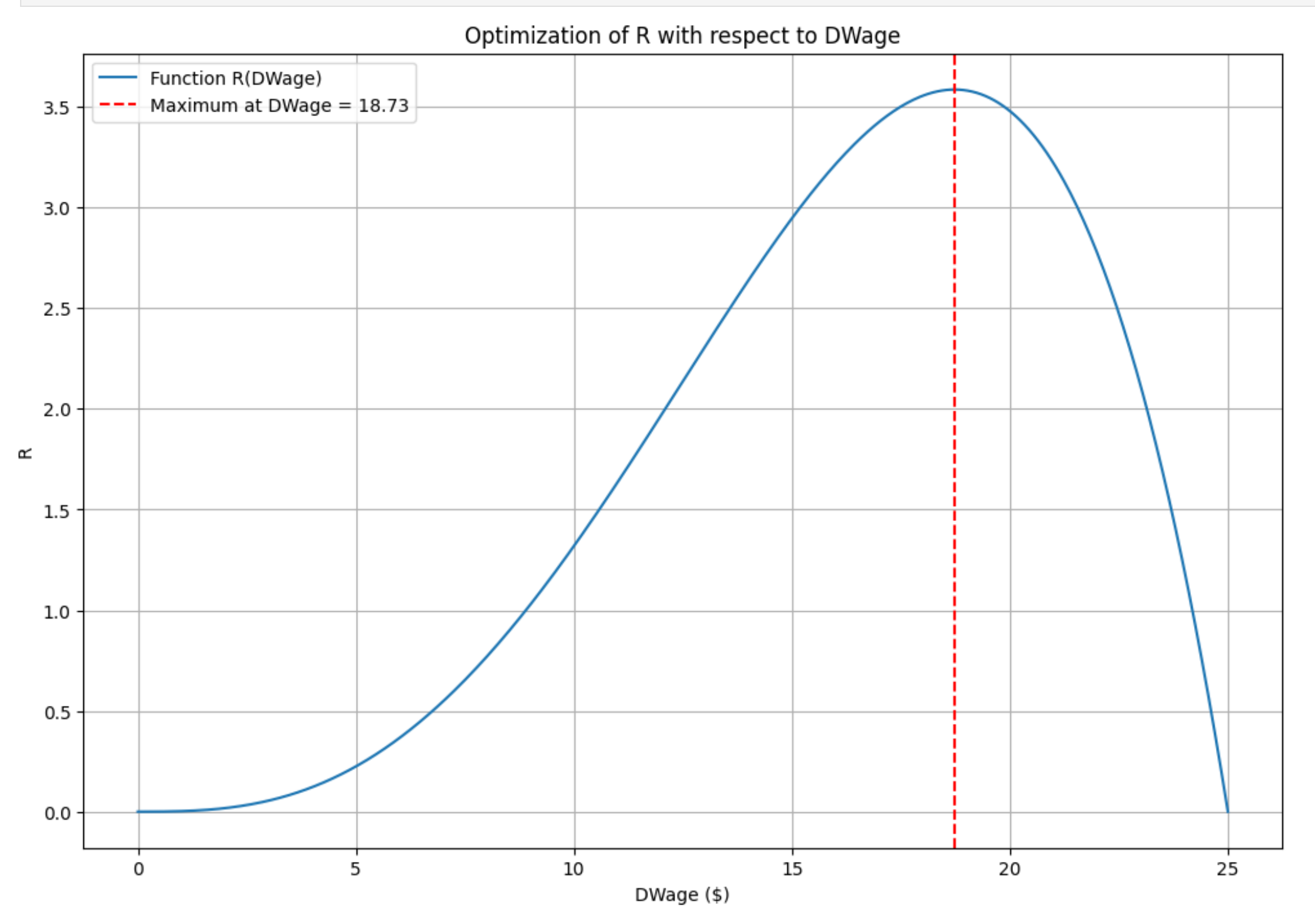


```
The polynomial model is: y = 0.0086x^3 + 0.0019x^2
```

Therefore, R is now:

$$R = \frac{(0.0086 * (DWage)^3) + (0.0019 * (DWage)^2)}{100} * (25 - DWage)$$

```
def R(DWage):
    return ((0.0086*DWage**3) + (0.0019*DWage**2))/100 * (25 - DWage)

res = minimize_scalar(lambda x: -R(x), bounds=(0, 25), method='bounded')
max_dwage = res.x
max_value = -res.fun

dwage_values = np.linspace(0, 25, 500)
r_values = R(dwage_values)

plt.figure(figsize=(12, 8))
plt.plot(dwage_values, r_values, label='Function R(DWage)')
plt.axvline(max_dwage, color='red', linestyle='--', label=f'Maximum at DWage = {max_dwage:.2f}')
plt.title('Optimization of R with respect to DWage')
plt.xlabel('DWage ($)')
plt.ylabel('R')
plt.legend()
plt.grid(True)
plt.show()

print(f"The maximum value of R occurs at DWage = {max_dwage:.2f} and R = {max_value:.2f}")
```



```
The maximum value of R occurs at DWage = 18.73 and R = 3.58
```

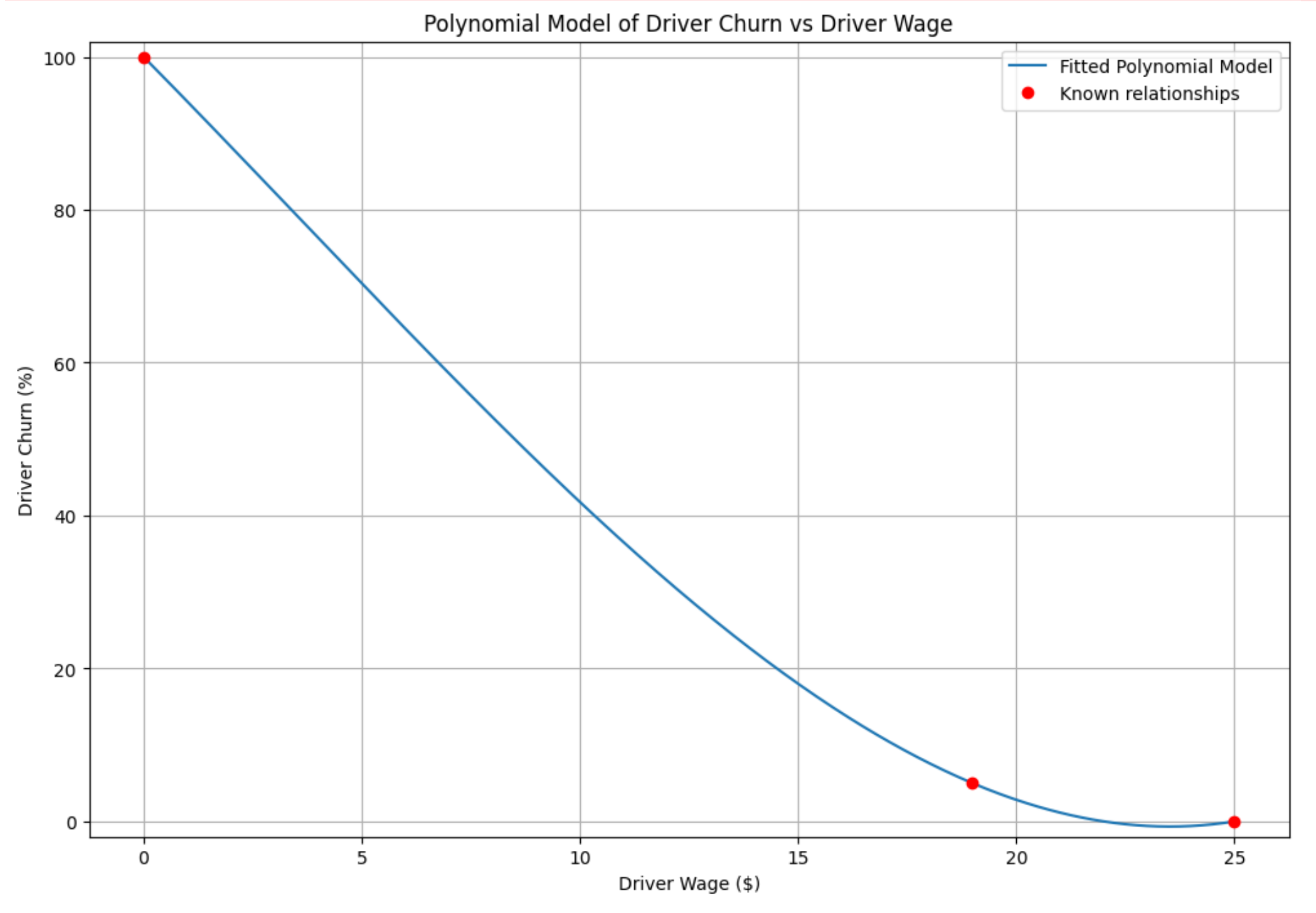
- Therefore, the net revenue for a given month would be maximised by setting the driver's wage to $18.73.
- However, the fact that the rider's churn is affected by driver's wage by proxy of the match rate, similar to driver's churn, the situation becomes more complex.
- The optimal 12-month net revenue may not be reached by this driver's wage due to additional CAC costs to replace driver's and customers lost from previous months. This will be considered below.

4.3. Model for Driver Churn vs Driver Wage

The model to be utilised for customers has already been specified in the prompt parameters above, but the model for driver churn needs to be defined as below.

```
x = np.array([0, 19, 25])
y = np.array([100, 5, 0])

coefficients = np.polyfit(x, y, 3)

polynomial = np.poly1d(coefficients)

xp = np.linspace(0, 25, 400)
plt.figure(figsize=(12, 8))
plt.plot(xp, polynomial(xp), label='Fitted Polynomial Model')
plt.plot(x, y, 'ro', label='Known relationships')
plt.title('Polynomial Model of Driver Churn vs Driver Wage')
plt.xlabel('Driver Wage ($)')
plt.ylabel('Driver Churn (%)')
plt.ylim(-2, 102)
plt.legend()
plt.grid(True)
plt.show()

print("The polynomial is: ")
print(polynomial)
```

```
/var/folders/_4/mds0ttjx4s90r_nvk7c53j080000gn/T/ipykernel_9686/1207391935.py:4: RankWarning: Polyfit may be poorly conditioned
  coefficients = np.polyfit(x, y, 3)
```

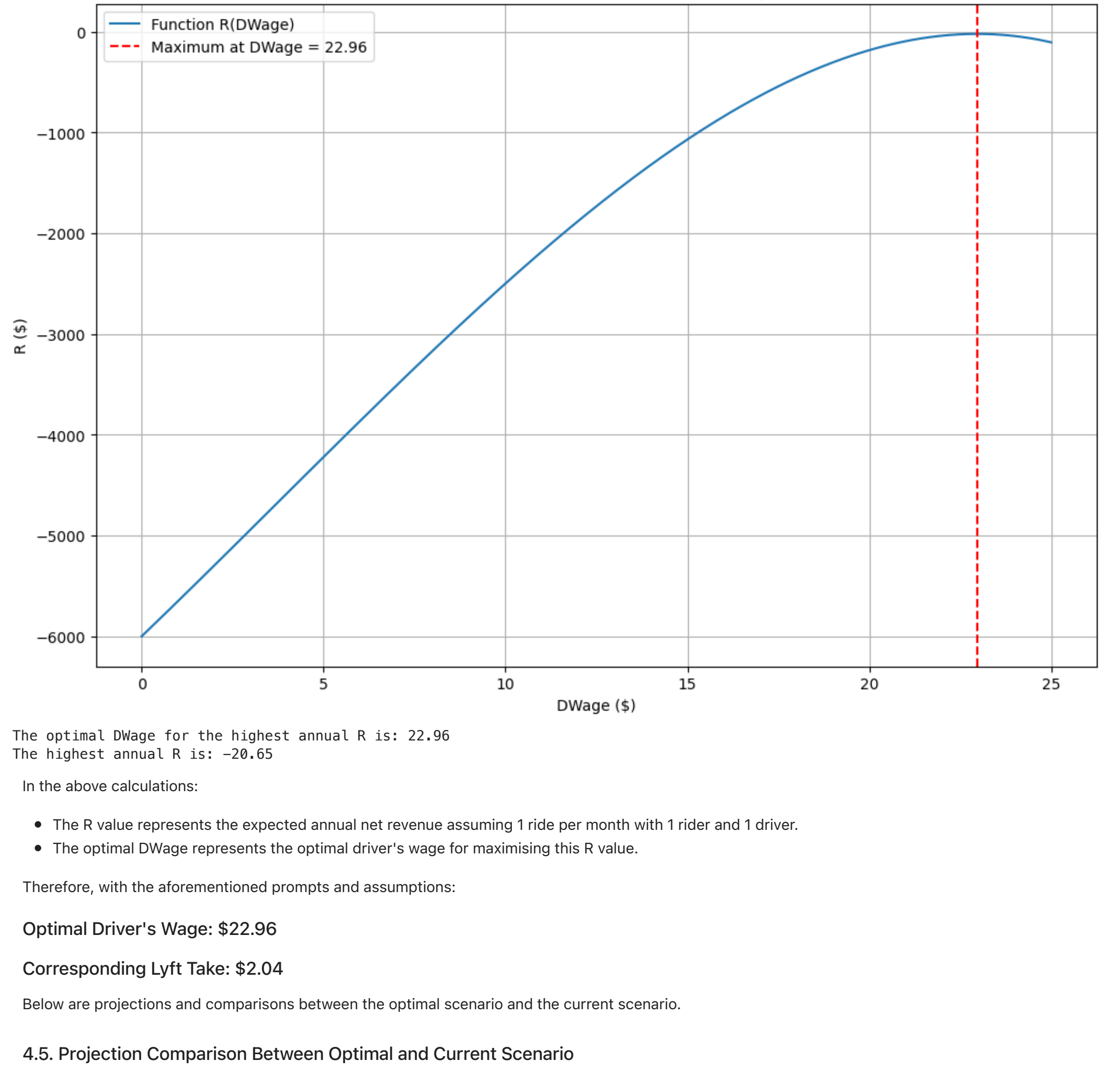


```
The polynomial is: 
          3           2
0.005022 x - 0.05432 x - 5.781 x + 100
```

4.4. 12-Month Rolling Net Revenue

Since 1 driver can do 100 rides per month, we'll calculate this based on 1 driver, knowing that 1 driver churns with a probability defined by the model above.

```
def MonthlyR(DWage):

    rideMatchRate = ((0.0086*DWage**3) + (0.0019*DWage**2))/100

    monthlyR = rideMatchRate * (25 - DWage)

    riderCRate = (rideMatchRate * 0.1) + (rideMatchRate * 0.33) # rider churn rate
    totRiderCAC = riderCRate * 15 # expected total CAC for riders, used average CAC of $15

    driverCRate = ((0.005022*DWage**3) - (0.05432*DWage**2) - 5.781*DWage + 100)/100 # driver churn rate
    totDriverCAC = driverCRate * 500 # expected total CAC for drivers, used average CAC of $500

    return monthlyR - totRiderCAC - totDriverCAC

def AnnualR(DWage):
    return 12 * MonthlyR(DWage)

res = minimize_scalar(lambda x: -AnnualR(x), bounds=(0, 25), method='bounded')
max_dwage = res.x

dwage_values = np.linspace(0, 25, 400)
r_values = [AnnualR(dwage) for dwage in dwage_values]

plt.figure(figsize=(12, 8))
plt.plot(dwage_values, r_values, label='Function R(DWage)')
plt.axvline(max_dwage, color='red', linestyle='--', label=f'Maximum at DWage = {max_dwage:.2f}')
plt.title('Optimization of R with respect to DWage')
plt.xlabel('DWage ($)')
plt.ylabel('R ($)')
plt.legend()
plt.grid(True)
plt.show()

print(f"The optimal DWage for the highest annual R is: {res.x:.2f}")
print(f"The highest annual R is: {-res.fun:.2f}")
```



```
The optimal DWage for the highest annual R is: 22.96
The highest annual R is: -20.65
```

In the above calculations:

- The R value represents the expected annual net revenue assuming 1 ride per month with 1 rider and 1 driver.
- The optimal DWage represents the optimal driver's wage for maximising this R value.

Therefore, with the aforementioned prompts and assumptions:

Optimal Driver's Wage: $22.96

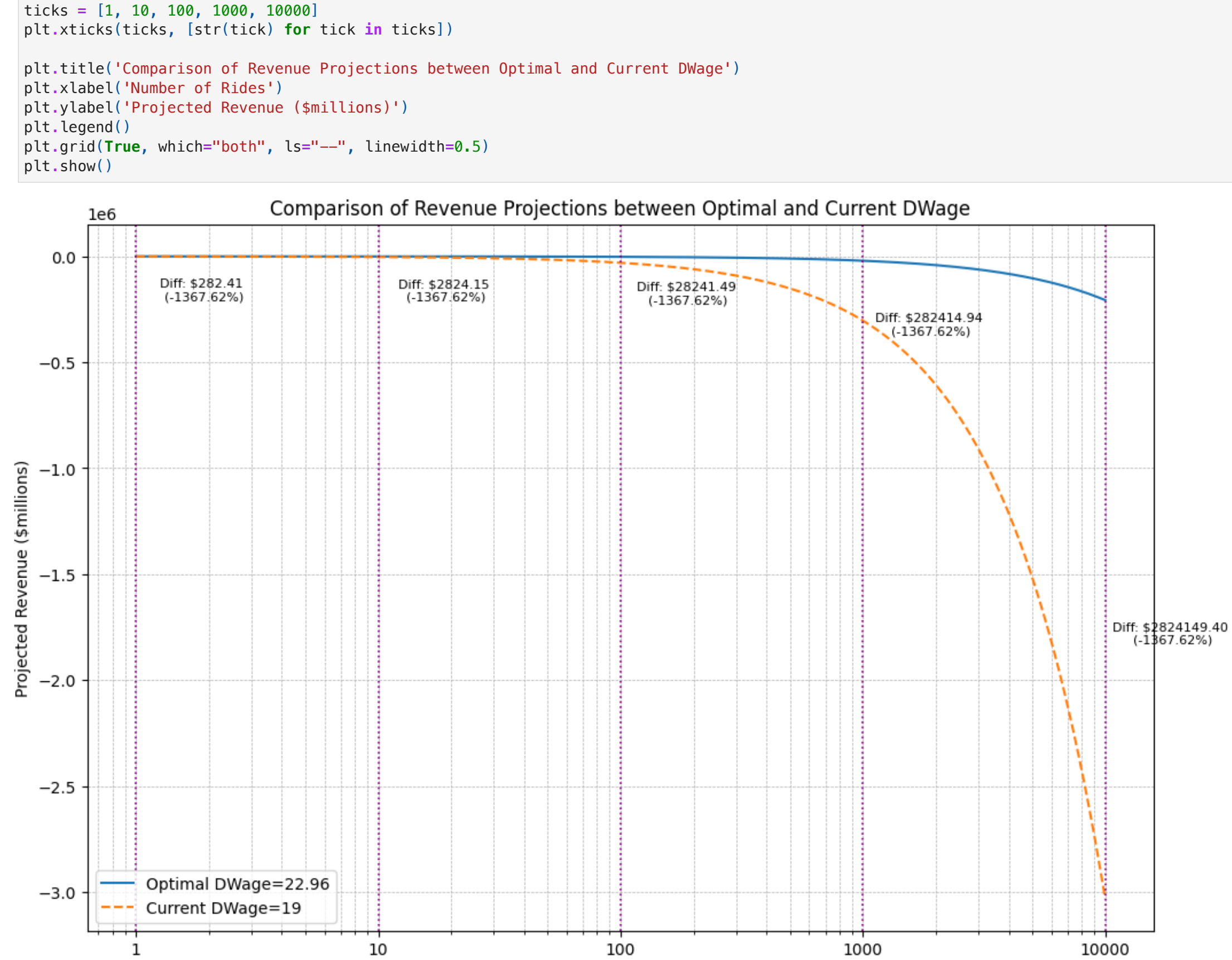
Corresponding Lyft Take: $2.04

Below are projections and comparisons between the optimal scenario and the current scenario.

4.5. Projection Comparison Between Optimal and Current Scenario

Since my model takes into account the expected annual R, it only needs to be multiplied by the number of rides to project annual net revenue.

```
rides = np.logspace(0, 4, 400, base=10)

revenue_proj_optimal = [AnnualR(max_dwage) * ride for ride in rides]
revenue_proj_19 = [AnnualR(19) * ride for ride in rides]

plt.figure(figsize=(12, 8))

plt.plot(rides, revenue_proj_optimal, label=f'Optimal DWage={max_dwage:.2f}')
plt.plot(rides, revenue_proj_19, label='Current DWage=19', linestyle='--')

points = [1, 10, 100, 1000, 10000]

for point in points:
    optimal_value = AnnualR(max_dwage) * point
    value_19 = AnnualR(19) * point

    nominal_diff = optimal_value - value_19
    percent_diff = (nominal_diff / optimal_value) * 100 if value_19 != 0 else 0

    plt.axvline(point, color='purple', linestyle=':', linewidth=1.2)
    label_text = f"Diff: ${nominal_diff:.2f} \n({percent_diff:.2f}%)"
    plt.annotate(label_text, (point, (optimal_value + value_19) / 2), textcoords="offset points", xytext=(42.5,-27.5), ha='center', fontsize=8)


plt.xscale('log')
ticks = [1, 10, 100, 1000, 10000]
plt.xticks(ticks, [str(tick) for tick in ticks])

plt.title('Comparison of Revenue Projections between Optimal and Current DWage')
plt.xlabel('Number of Rides')
plt.ylabel('Projected Revenue ($millions)')
plt.legend()
plt.grid(True, which="both", ls="--", linewidth=0.5)
plt.show()
```



5. Conclusion

5.1. Pricing Strategy

As can be seen by the projections above, it's clear that the optimal scenario would be to pay the driver's more for each ride (22.96 dollars), by roughly 3.96 dollars more to achieve the maximum possible net revenue. The current scenario is considerably worse when scaled up to 10,000 rides, with projected net revenues leading to a 3 million dollar loss per 10,000 rides if no other strategies are implemented.

5.2. Other Strategies

There are some key takeaways:

- The match rate needs to be maximised to capture as much rider revenue as possible, but not too far such that the diminishing returns of increasing driver's wage affect the net revenue by reducing Lyft's take.
  - Strategies that will increase the match rate for the same driver's wage will massively benefit Lyft's net revenue.
  - For example, improving the algorithm to give closer requests to driver's so they'll be more likely to accept would be a good strategy.
  - Another great strategy for this would be to vary the driver's wage according to proximity of the driver to a rider's request to make further away ones earn more, increasing the likelihood of request acceptance.
- It is hugely important to reduce the CAC of a driver, as this seems to be hurting the net revenue heavily.
  - Perhaps this can be done by streamling costs of driver onboarding.
  - Another strategy would be to find better organic marketing strategies for acquiring drivers to reduce marketing expenses.
  - It would also be better to decrease the churn rate for the same driver's wage, perhaps by offering perks or a company-specific insurance.