

---

---

# What is Data Science

— Introduction and Examples —

---

---

# Outline

- What is Data Science?
- Examples from Industry. Amazon, Netflix, Booking.com, New York Times.
  - Predictive Learning (Supervised)
  - Descriptive Learning (Unsupervised)
  - Prescriptive Learning (Targeted/Uplifted)
- Why is Data Science Important?
- Overview of methods of Machine Learning.
- What will you learn in this course?
- What should you know before taking this course?

# What is Data Science?

- **Predictive (Supervised Learning):** The science of using data to predict an outcome (clicking, subscription, cancerous cells, user churn, virality, etc.)
- **Descriptive (Unsupervised Learning):** Using data to group items/users into categories (ie. Risky Customers, )
- **Prescriptive (Targeted Modeling):** Using data to find actionable insights to optimize for some metric (ie. who should receive a marketing email)
- **Exploratory:** Can we describe characteristics of items/users with particular attributes we are interested in? (ie. Are new users who sign up for the new york times mostly Democrats?)

---

---

# Predictive Learning

(Supervised)

---

---

# Predictive Learning - Summary

- Predictive learning attempts to learn a model from data  $X$  which predicts a variable  $y$  (ie. type of movie, number of views would be  $X$ ,  $y$  is your rating).
- Learns from data which has 'correct' answers given data inputs - this is why it's "supervised".
- This course will focus largely on predictive learning.
- Algorithms:
  - Linear Regression.
  - Random Forest/Decision Tree Regression.
  - SVM
  - Naives Bayes (For Multiarmed Bandits for instance)
  - Maximum Likelihood (more advanced).

# Problem Motivation 1. Amazon.com

Can we predict how a user will rate an item? Why do we care?



## Nikon COOLPIX S33 Waterproof Digital Camera (Blue)

by Nikon



582 customer reviews | 196 answered questions

**#1 Best Seller** in Digital Point & Shoot Cameras

List Price: \$149.96

Price: **\$129.00** & **FREE Shipping**. [Details](#)

You Save: \$20.95 (14%)

**In Stock.**

**Want it Friday, Sept. 30?** Order within **19 hrs 2 mins** and choose **Same-Day Delivery** at checkout. [Details](#)

Ships from and sold by Amazon.com. Gift-wrap available.

Color: **Blue**



Style: **Base**

Accessory Bundle **Base**

- Waterproof up to 33 feet deep; shockproof up to 5 feet; freezeproof down to 14° F
- 3x wide-angle NIKKOR glass zoom lens
- 13.2-MP CMOS sensor
- Full HD 1080p videos with stereo sound
- Oversized buttons and easy menus

- Can we predict how you would rate this item based on what we know about you?
- Why do we care? Will increase purchase rate and this can be measured in \$ via an experiment.
- Good recommendations = \$\$.

# Problem Motivation 2. Booking.com

Can we predict that a hotel is likely to sell out soon? If so when? Why do we care?

The screenshot displays two hotel listings on the Booking.com interface. The top listing is for 'Dorsett Shepherd's Bush', featuring a '34% off' badge, a 'Value Deal' icon, and a 'Fabulous 8.6' rating from 524 reviews. It mentions 'Hammersmith and Fulham, London' and 'Subway access'. A 'Popular now!' alert states that 13 people are looking at the hotel, with the latest booking occurring less than 1 minute ago. The bottom listing is for 'City Marque Albert Serviced Apartments', showing a 'Value Deal' icon, a 'Good 7.8' rating from 85 reviews, and 'Central London, London' with 'Subway access'. A red-bordered box highlights a warning: 'It's likely that these apartments will be sold out within the next 2 days.' Below this, it says 'Latest booking: 2 hours ago'. A 'Last chance! We have only 1 left on our site!' message is also visible. Both listings include a 'Book now' button. A thin black arrow points from the 'Book now' button of the City Marque listing towards the text on the right.

**Dorsett Shepherd's Bush**  
★★★★★ Value Deal 1100  
Hammersmith and Fulham, London – Subway access  
Popular now! There are 13 people looking at this hotel.  
Latest booking: Less than 1 minute ago  
Dorsett Double Room FREE cancellation - PAY LATER  
Just booked!  
2 more room types >

**City Marque Albert Serviced Apartments**  
★★★★★ Value Deal 400  
Central London, London – Subway access  
There are 2 people looking at these apartments.  
It's likely that these apartments will be sold out within the next 2 days.  
Latest booking: 2 hours ago  
Studio Apartment - 377 HP  
Last chance! We have only 1 left on our site!

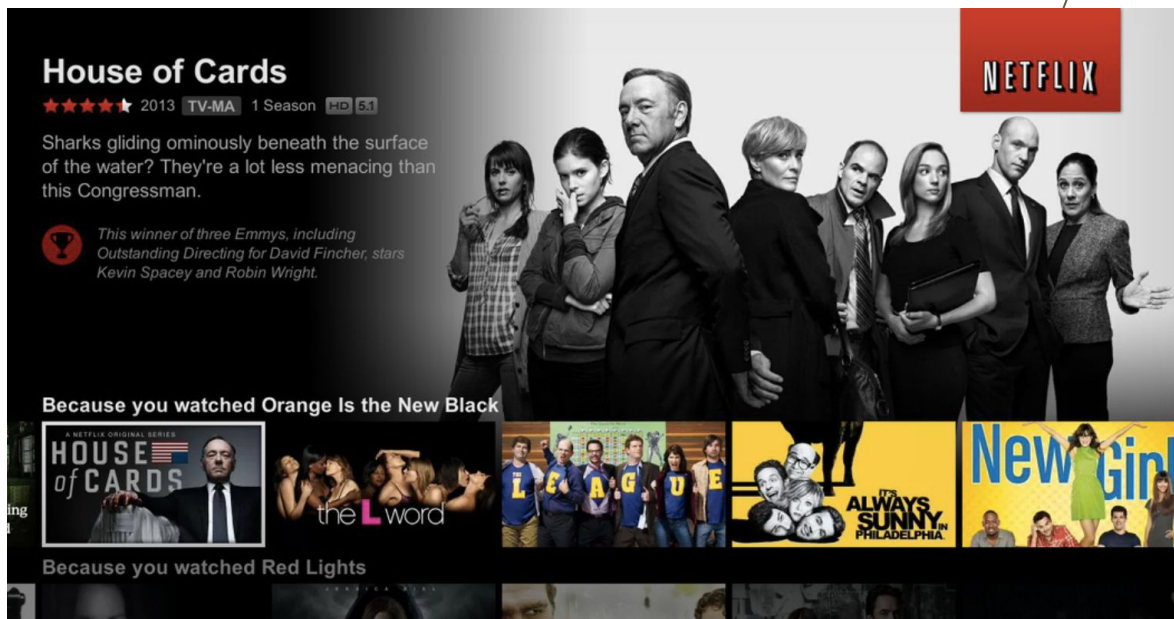
**Fabulous 8.6**  
Score from 524 reviews  
-34% £180- £120  
Book now

**Good 7.8**  
Score from 85 reviews  
£181.01  
Book now

- **Conversion:** Users may be **more inclined to purchase** if they are aware the room may sell out soon (improve purchase rate).
- **Retention:** Users may be **only interested in certain hotel options** and not be aware that they don't have the luxury of waiting - **this could upset customers** if the hotel sells out without warning (customer service).

# Problem Motivation 3. Netflix.com

Can we predict how you would rate a movie?



- **Conversion:** Users may be **more inclined to purchase** if they are aware the room may sell out soon (improve purchase rate).
- **Retention:** Users may be **only interested in certain hotel options** and not be aware that they don't have the luxury of waiting - **this could upset customers** if the hotel sells out without warning (customer service).



# Problem Motivation 4. Nytimes.com

Can we predict which articles you would like to read?


HOME PAGE TODAY'S PAPER VIDEO MOST POPULAR U.S. Edition ▼ dorian.goldman... ▼ Help

## The New York Times Recommendations


### Recommended for You

Recommendations are based on what you've viewed recently.

THE LEARNING NETWORK  
**News Quiz | Sept. 20-26, 2016**  
By KATHERINE SCHULTEN | Sep 27th 2016  
Have you been following the news? Take our quiz to see what you know and to learn more.



U.S.  
**Who Won the Debate? Commentators Give Hillary Clinton the Edge**  
By ALAN RAPPEPORT | Sep 27th 2016  
Mrs. Clinton drew praise on social media for her grasp of policy and her preparedness for her first debate with Donald J. Trump.



#### Your Recent Activity

dorian.goldman (Not you? [Log out](#))

**40** ARTICLES AND MULTIMEDIA

#### Most Viewed Sections

Politics	12
Today's Opinion	9
N.Y.	6
U.S.	5
Television	2

**Engagement:** Users may be more inclined to purchase if they are aware the room may sell out soon (improve purchase rate).

**Retention:** Users may be only interested in certain hotel options and not be aware that they don't have the luxury of waiting - this could upset customers if the hotel sells out without warning (customer service).

# Mathematical Formulation - Predictive Learning

We wish to find  $f$ :  $y_t \sim f(x_{t-1})$

$y_t$

The number of rooms available at time  $t$   
(days)

$x_{t-1}$

Everything we know now about the room and  
hotel.

$x_{t-1} \mapsto f(x_{t-1})$

A regressor which predicts the number of  
rooms that will be available at time  $t$ .

**Note:** Regression allows **more flexibility** in fitting against **various count distributions**.

---

---

# Descriptive Learning

— (Unsupervised) —

---

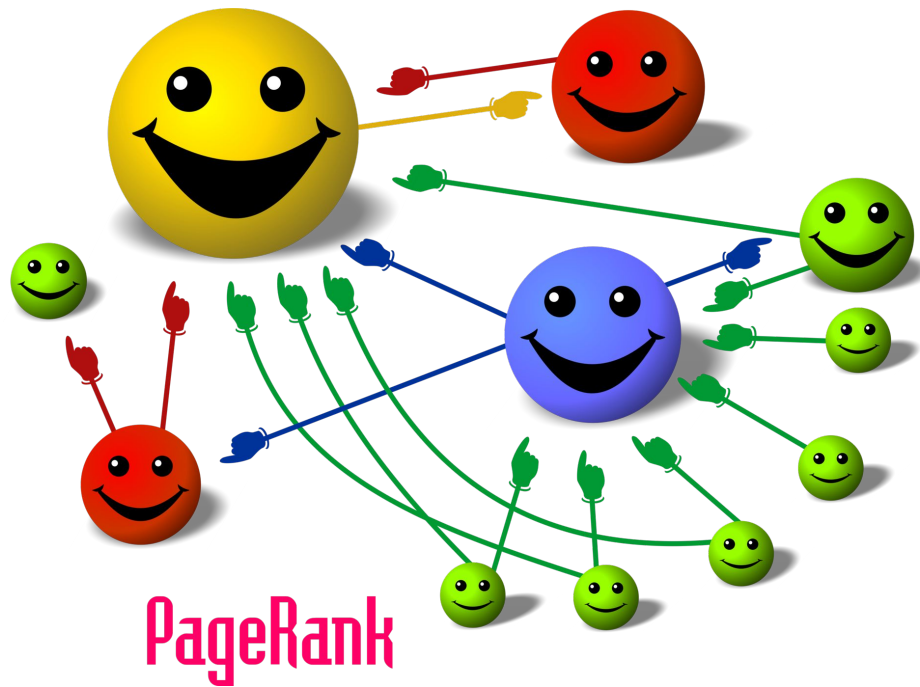
---

# Descriptive Learning - Summary

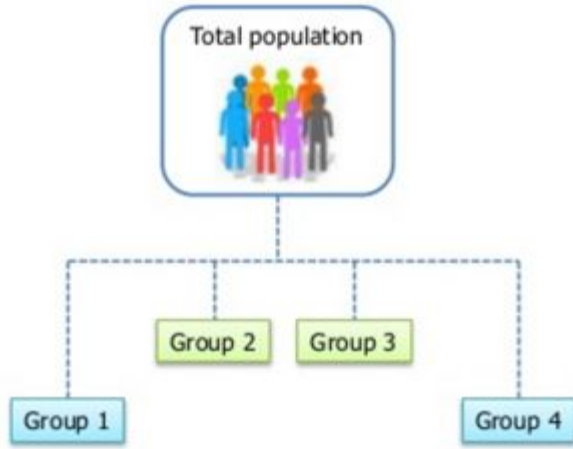
- Try to infer a hidden structure in the data without proper training examples (no teacher, hence 'unsupervised').
- I.e. Group certain customers into 'wealthy, educated' - no clear way of evaluating performance of model.
- This course will focus less on unsupervised learning.
- **Algorithms:**
  - K-means clustering.
  - Decision Tree Clustering.
  - SVM

# Problem Motivation 1. Google's Page Rank

Cartoon illustrating the basic principle of PageRank. The size of each face is proportional to the total size of the other faces which are pointing to it. (Source: Wiki)



# Problem Motivation 2. NyTimes.com



- How to cluster subscribers into different categories ?
- **Possible categories:** “Young, uneducated, low income”, “Wealthy, older, educated”, etc.
- Clustering algorithms may naturally find these categories, and this can inform marketing and sales strategy.
- **Possible features:** Age, income, location, reading history (online), number of complaints, subscription tenure, etc.

---

---

# Prescriptive Learning

Examples

---

---

# Prescriptive Learning - Summary

- Prescriptive learning attempts to find an optimal action  $a$  to maximize the expected reward/outcome (ie. who should we show this kind of ad to, or who should we send this marketing email to?)
- A well defined metric is used to determine the performance of such a model (will be seen later).
- **Algorithms:**
  - Relies entirely on maximizing expectation of reward conditioned on user attributes and action.
  - Incredibly useful since it's actionable.
  - Can be “live” (multiarmed bandit) or from logged data (uplift modeling).
  - Easiest when we have an A/B (randomized controlled trial) where we can measure causal inference of an outcome conditioned on an action.



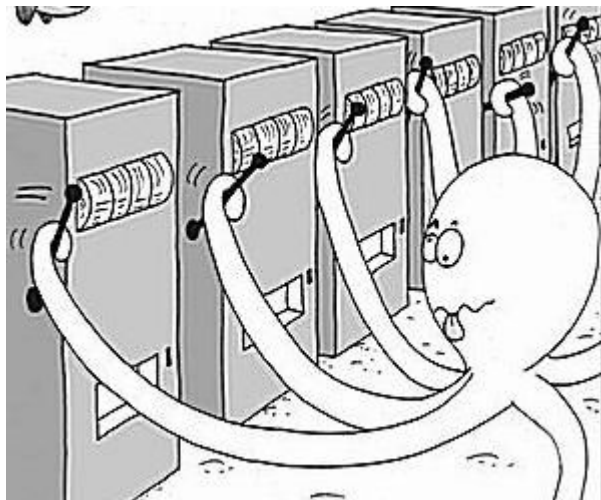
# Uplift Modeling - what is it?



- Not everyone should receive the same action.
- Maybe some users will be angered by an advertisement, and others will only sign up if you send them a particular ad.

# Multiarmed Bandits and Nytimes.com

For a new user with no data, how do we predict what you should see?



- **Engagement:** Users may be **more inclined to purchase** if they are aware the room may sell out soon (improve purchase rate).
- **Retention:** Users may be **only interested in certain hotel options** and not be aware that they don't have the luxury of waiting - **this could upset customers** if the hotel sells out without warning (customer service).

# Model 1 - Linear Regression

$$\hat{y}_t = \alpha_0 X_{t-1} + \alpha_1 R + \alpha_2 S + \alpha_3 P + \alpha_4 L$$

$X_{t-1}$  is the number of rooms booked last week

$R$  is the rating of the hotel

$S$  is the number of sellouts the hotel had last year

$P$  is the price of the room

$L$  is the location rating

$\hat{y}_t$  is the number of rooms that will be booked tomorrow

**Regression in this case provides more information than a simple classification model would.**

## **Advantages:**

- Simple and fast
- Easy to implement

## **Disadvantages:**

- Gaussian priors don't model count data well.
- Better model learns different coefficient for each hotel. This could be slow however.

# Linear Regression - Performance Evaluation

## 2. Split into training and testing data and evaluate model on testing data

```
In [634]: # Split the data into training/testing sets
X_train = X[0:int(size*0.8)]
X_test = X[int(size*0.8):]

# Split the targets into training/testing sets
y_train = y[0:int(size*0.8)]
y_test = y[int(size*0.8):]

# Create linear regression object
regr = LinearRegression()

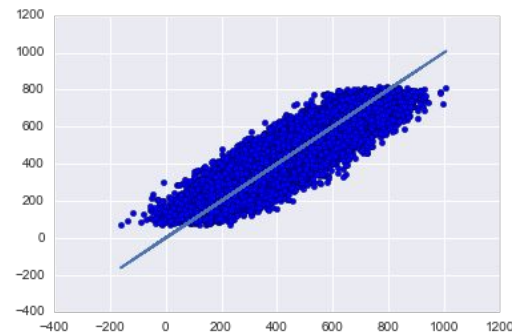
# Train the model using the training sets
regr.fit(X_train, y_train)

# The coefficients
print('Coefficients: \n', regr.coef_)
# The mean square error
print("Residual sum of squares: %.2f"
      % np.mean((regr.predict(X_test) - y_test) ** 2))
# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % regr.score(X_test, y_test))

('Coefficients: \n', array([ 7.19732079e-02,  4.30278779e-02,  6.00717352e-02,
 9.01371446e-02, -7.56655448e-05,  5.55775648e-02]))
Residual sum of squares: 63.89
Variance score: 0.79
```

	account	date	hotel_rating	location	price_per_night_avg	purchase_velocity_lastweek	rooms_left	sellouts_total
0	66623	2015-06-01	5.4	8.9	598	246	173	7
1	74008	2015-06-01	9.4	0.8	281	298	180	9
2	16859	2015-06-01	9.9	0.6	423	408	153	1
3	15919	2015-06-01	2.4	9.6	334	374	2	2
4	15407	2015-06-01	0.2	5.8	247	246	290	8
5	37124	2015-06-01	5.7	3.5	250	205	286	9

Reality



Predicted

Need such a table for each date and hotel. (hotel, date)

# Model 2 - Poisson Regression

We assume here that:

$$\log \mathbb{E}(y|x) = \alpha + \beta \cdot x$$

$$Y_{t+1} \sim \text{Pois}(\mathbb{E}(y|x_t))$$

$$\log \mathbb{E}(y_t|x) = 0.01X_{t-1}$$

## Advantages:

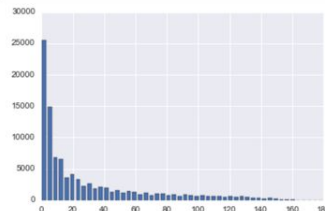
- Models count data better.
- Structure allows for testing a wider variety of distributions.

## Disadvantages:

- Slower
- Harder to implement

```
In [577]: # Simple linear model  
rooms_sold_lambdas = np.exp( 0.01 * purchase_velocity_lastweek )  
rooms_sold = np.random.poisson(rooms_sold_lambdas)
```

```
In [578]: hist, bins = np.histogram(rooms_sold, bins=50)  
width = 0.7 * (bins[1] - bins[0])  
center = (bins[-1] + bins[1]) / 2  
plt.bar(center, hist, align='center', width=width)  
plt.show()
```



# Poisson Regression - Performance evaluation

## 3. Evaluate performance in terms of $R^2$

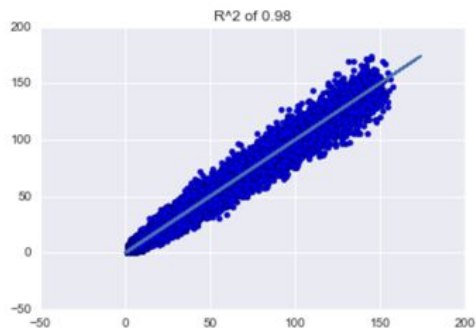
```
In [593]: predictions = np.exp(np.dot(X_test.values, res.params))
```

```
In [604]: SSreg = np.mean((predictions - y_test) ** 2)
SSStot = np.mean((y_test - np.mean(y_test)) ** 2)
print 1 - SSreg/SSStot
```

```
0.975778913506
```

```
In [606]: r2 = (predictions - np.sum()))
plt.title('R^2 of 0.98')
plt.plot(y_test, y_test)
plt.scatter(predictions, y_test)
```

```
Out[606]: <matplotlib.collections.PathCollection at 0x13a236d90>
```



Fitting is done via **maximum likelihood**.

```
from statsmodels.base.model import GenericLikelihoodModel
class NBin(GenericLikelihoodModel):
    def __init__(self, endog, exog, **kws):
        super(NBin, self).__init__(endog, exog, **kws)
    def nloglikeobs(self, params):
        ll = _ll_nb2(self.endog, self.exog, params)
        return -ll
    def fit(self, start_params=None, maxiter=10000, maxfun=5000, **kws):
        if start_params == None:
            # Reasonable starting values
            start_params = [0.001, 0, 0, 0, 0]
            start_params[0] = np.log(self.endog.mean())
            return super(NBin, self).fit(start_params=start_params,
                                         maxiter=maxiter, maxfun=maxfun,
                                         **kws)
    def predict(self, exog):
        print 'hello'
        return super(NBin, self).predict(exog)
```

```
mod = NBin(y_train, X_train)
res = mod.fit()

Optimization terminated successfully.
Current function value: 2.657540
Iterations: 241
Function evaluations: 396
```

	coef	std err	z	P> z	[95.0% Conf. Int.]
hotel_rating	0.0004	0.000	1.709	0.087	-5.49e-05 0.001
location	0.0057	0.000	26.109	0.000	0.005 0.006
price_per_night_avg	3.519e-06	4.16e-06	0.846	0.398	-4.64e-06 1.17e-05
purchase_velocity_lastweek	0.0100	5.07e-06	1974.943	0.000	0.010 0.010
rooms_left	2.06e-05	4.35e-06	4.733	0.000	1.21e-05 2.91e-05
sellouts_total	-0.0093	0.000	-38.774	0.000	-0.010 -0.009

# How to predict when it will sell out?

## 4. Predict for days into the future.

```
In [637]: day1_predictions=regr.predict(X_test)
X_test.loc[:, 'purchase_velocity_lastweek'] += day1_predictions
day2_predictions=regr.predict(X_test)
X_test.loc[:, 'purchase_velocity_lastweek'] += day2_predictions
day3_predictions=regr.predict(X_test)
X_test.loc[:, 'purchase_velocity_lastweek'] += day3_predictions
```

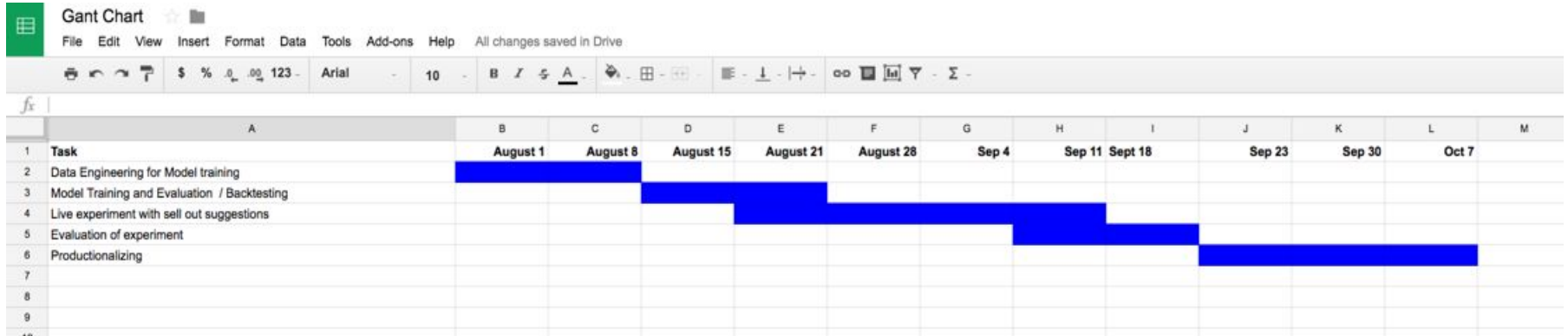
```
In [638]: X_test.loc[:, 'day1']=day1_predictions
X_test.loc[:, 'day2']=day2_predictions
X_test.loc[:, 'day3']=day3_predictions

def get_sellout_time(row):
    if row['day1'] > row['rooms_left']:
        return 1
    elif row['day2'] > row['rooms_left']:
        return 2
    elif row['day3'] > row['rooms_left']:
        return 3
    else:
        return -1

X_test.loc[:, 'sellout_days']=X_test.apply(lambda row : get_sellout_time(row), axis=1)
```

- Further model evaluation can be done using **recall** (probably most important in this case)

# Timeline for Project - Gantt chart





# Conclusion

- Linear model is simple and fast but not ideal for count data.
- Poisson Regression is better but could be slow.
- Most important thing is to build a simple model, evaluate via experiment and then decide if we should move forward.

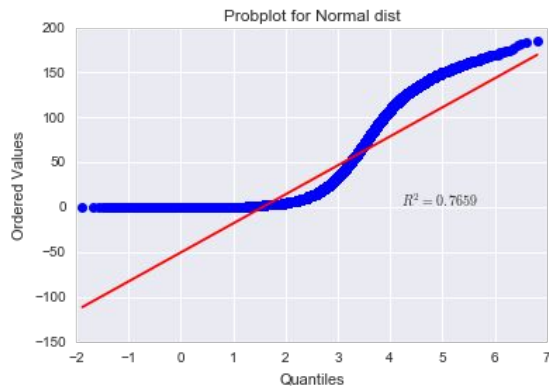
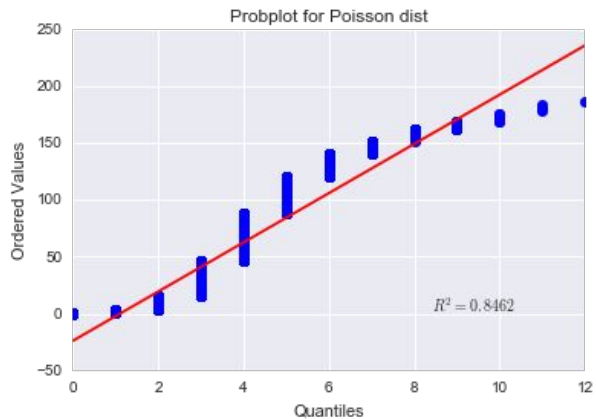
# Appendix

- Engineering details
- Fake data generation

# How would you train the model?

- Choose all hotels and days for which there was a sell out and random selection of others.
- Train model for time  $t-1$  to predict the amount of rooms sold at time  $t$ , the day of the sell out (or before).
- This gives a collection of (hotels (attributes), dates).
- Split into some percentage for training, then another percentage for testing.

# Which distribution is best?



# Poisson Regression - Fitting

```
from statsmodels.base.model import GenericLikelihoodModel
class NBin(GenericLikelihoodModel):
    def __init__(self, endog, exog, **kws):
        super(NBin, self).__init__(endog, exog, **kws)
    def nloglikeobs(self, params):
        ll = _ll_nb2(self.endog, self.exog, params)
        return -ll
    def fit(self, start_params=None, maxiter=100000, maxfun=5000, **kws):
        if start_params == None:
            # Reasonable starting values
            start_params = [0.001, 0, 0, 0, 0, 0]
            start_params[0] = np.log(self.endog.mean())
        return super(NBin, self).fit(start_params=start_params,
                                     maxiter=maxiter, maxfun=maxfun,
                                     **kws)
    def predict(self, exog):
        print 'hello'
        return super(NBin, self).predict(exog)

mod = NBin(y_train, X_train)
res = mod.fit()
```

Optimization terminated successfully.  
Current function value: 2.657540  
Iterations: 241  
Function evaluations: 396

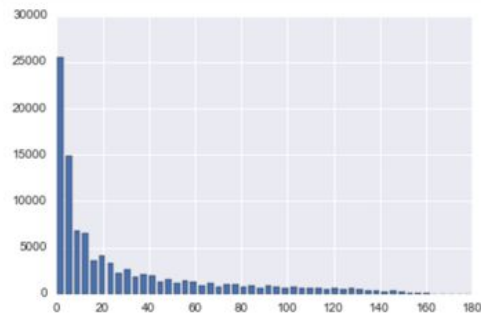
Fitting is done via **maximum likelihood**.

	coef	std err	z	P> z	[95.0% Conf. Int.]
hotel_rating	0.0004	0.000	1.709	0.087	-5.49e-05 0.001
location	0.0057	0.000	26.109	0.000	0.005 0.006
price_per_night_avg	3.519e-06	4.16e-06	0.846	0.398	-4.64e-06 1.17e-05
purchase_velocity_lastweek	0.0100	5.07e-06	1974.943	0.000	0.010 0.010
rooms_left	2.06e-05	4.35e-06	4.733	0.000	1.21e-05 2.91e-05
sellouts_total	-0.0093	0.000	-38.774	0.000	-0.010 -0.009

# Poisson Regression - Fake data generation

```
In [577]: # Simple linear model
rooms_sold_lambdas = np.exp( 0.01 * purchase_velocity_lastweek )
rooms_sold = np.random.poisson(rooms_sold_lambdas)
```

```
In [578]: hist, bins = np.histogram(rooms_sold, bins=50)
width = 0.7 * (bins[1] - bins[0])
center = (bins[:-1] + bins[1:]) / 2
plt.bar(center, hist, align='center', width=width)
plt.show()
```



# Linear Regression - Fake data generation

$$\hat{y}_t = \alpha_0 X_{t-1} + \alpha_1 R + \alpha_2 S + \alpha_3 P + \alpha_4 L$$

$X_{t-1}$  is the number of rooms booked last week

$R$  is the rating of the hotel

$S$  is the number of sellouts the hotel had last year

$P$  is the price of the room

$L$  is the location rating

$\hat{y}_t$  is the number of rooms that will be booked tomorrow

```
In [199]: df.head()
```

```
Out[199]:
```

	hotel_rating	location	price_per_night_avg	purchase_velocity_lastweek	rooms_left	sellouts_total
0	9.1	6.1	412	266	197	6
1	8.0	1.5	409	398	1	7
2	3.6	9.7	151	192	87	7
3	3.0	1.8	172	286	398	7
4	6.5	5.6	297	51	200	1

**Fake linear data was generated with random Gaussian noise.**

