Introduction

- Limited dependent variable modeling: background and motivation
- OLS approach: linear probability models (LPMs)
- Issues with LPM models
- Introduction to logit/probit models
- Understanding logit function

Introduction

- Thresholding
- Confusion/classification Matrix
- Receiver operator characteristic (ROC) curve
- Parameter interpretation
- Summary and concluding remarks

Background and Motivation

Limited Dependent Variable/Qualitative Response Regression

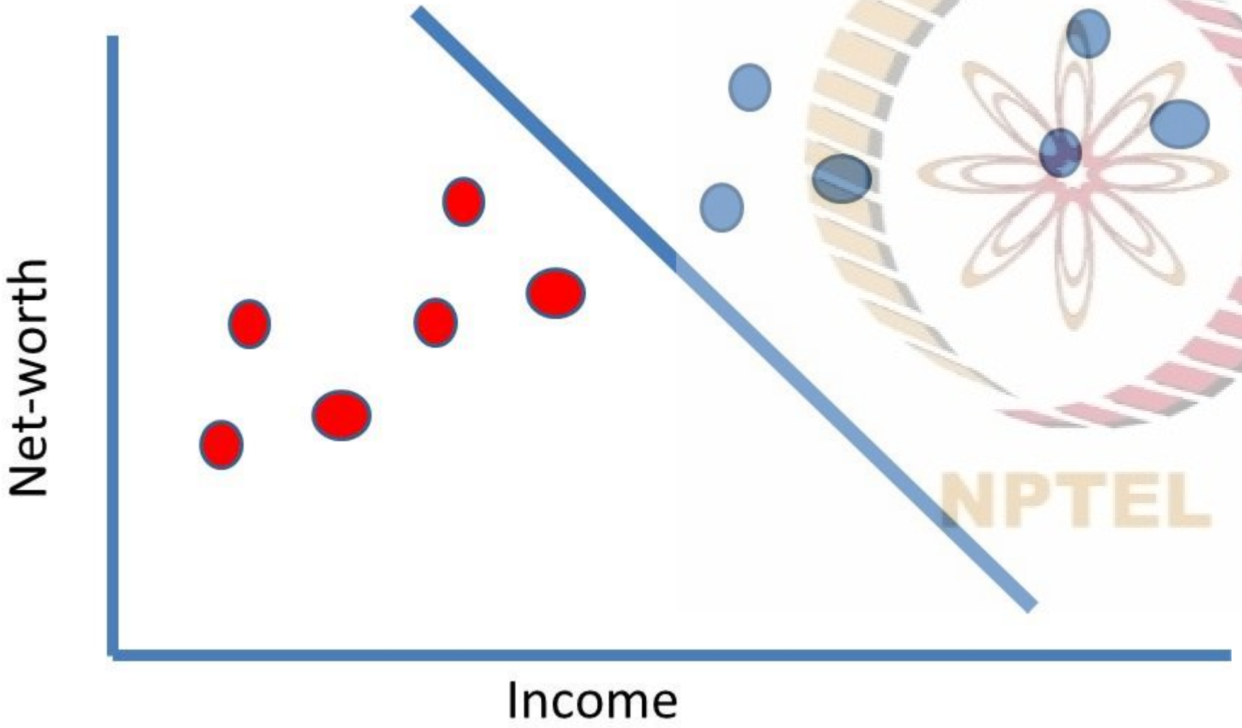
Discrete choice variables, limited dependent variables, or qualitative response variables are not suitable for modeling through linear regression models

Consider the following questions

- Why do firms choose to list their stocks on NSE vs. BSE?
- Why do some stocks pay dividends and others do not?
- What factors affect large corporate borrowers to default?
- What factors affect choices of internal vs. external financing?

Limited Dependent Variable/Qualitative Response Regression

Credit default scoring (classification problem)



Linear Probability Model (LPM)

Linear Probability Model (LPM)

- In such models, the dependent variable is Yes/No or 1/0 kind of variable
- First, we will examine a simple linear regression approach to deal with such models: linear probability model (LPM)
- This is the most simple approach to deal with binary dependent variables
- It is based on the assumption that the probability of an event (P_i) is linearly related to a set of explanatory variables, $x_{1i}, x_{2i}, \dots, x_{ki}$
- $P_i = p(y_i = 1) = \beta_1 + \beta_2 x_2 + \beta_3 x_{3i} + \cdots + \beta_k x_{ki} + u_i,\ i = 1, \dots \dots, N$

Linear Probability Model (LPM)

In such models, the actual probabilities cannot be observed, so your estimates (or dependent variables) would be 0s and 1s

- Consider the relationship between the size of a company "i" and its ability to pay dividends

$$Y_i = \beta_1 + \beta_2 X_i + u_i$$

where X_i= market capitalization of the firm, and Y_i=1 if the dividend is paid and 0 if the dividend is not paid.

Linear Probability Model (LPM)

In such models, the actual probabilities cannot be observed, so your estimates (or dependent variables) would be 0s and 1s

- This is called linear probability model. The conditional expectation of Y_i given X_i, i.e., $E(Y_i|X_i)$, can be interpreted that the event will occur given X_i: that is, $P(Y_i = 1|X_i)$
- $E(Y_i|X_i) = \beta_1 + \beta_2 X_i$ (assuming $E(u_i)$=0)

Summary

Issues with LPM

Issues with LPM

Non-normality and heteroscedasticity of error terms

- Y_i has the following distribution

  $E(Y_i|X_i) = 0\times(1-P_i) + 1\times(P_i) = P_i$

- This kind of model has a number of econometric issues

- What is the nature of errors: $u_i = Y_i - \beta_1 - \beta_2 X_i$?

Y_i	Probability
0	$1 - P_i$
1	P_i
Total	1

	u_i	Probability
When $Y_i = 1$	$1 - \beta_1 - \beta_2 X_i$	P_i
When $Y_i = 0$	$-\beta_1 - \beta_2 X_i$	$(1 - P_i)$

Issues with LPM

Non-normality and heteroscedasticity of error terms

- u_i is not normally distributed; although in large samples, it is not a problem
- u_is are heteroscedastic, i.e., they vary with Y_i

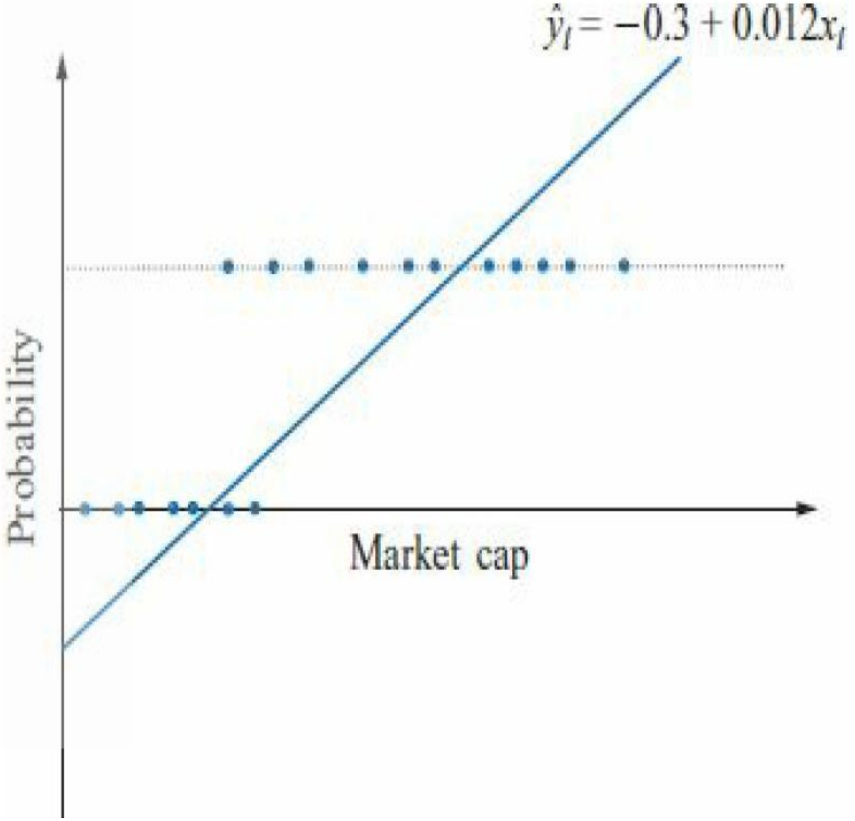
Y_i	Probability
0	$1 - P_i$
1	P_i
Total	1

	u_i	Probability
When $Y_i = 1$	$1 - \beta_1 - \beta_2 X_i$	P_i
When $Y_i = 0$	$-\beta_1 - \beta_2 X_i$	$(1 - P_i)$

Issues with LPM

Nonfulfillment of $0 \leq E(Y_i \mid X) \leq 1$

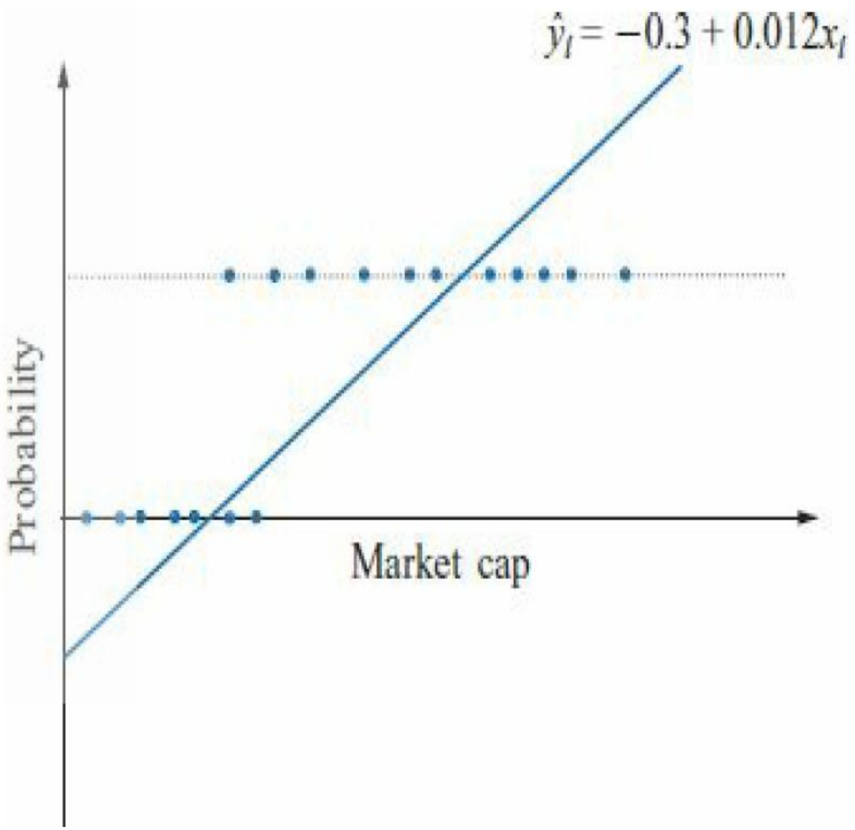
- $Y_i = -0.3 + 0.012X_i$; where X_i is in million dollars
- For every $1 million increase in size, the probability that the firm will pay dividend increases by 1.2%
- However, for X < $25 million and X > $88 million, the probabilities are less than 0 and more than 1



Issues with LPM

Nonfulfillment of $0 \leq E(Y_i \mid X) \leq 1$

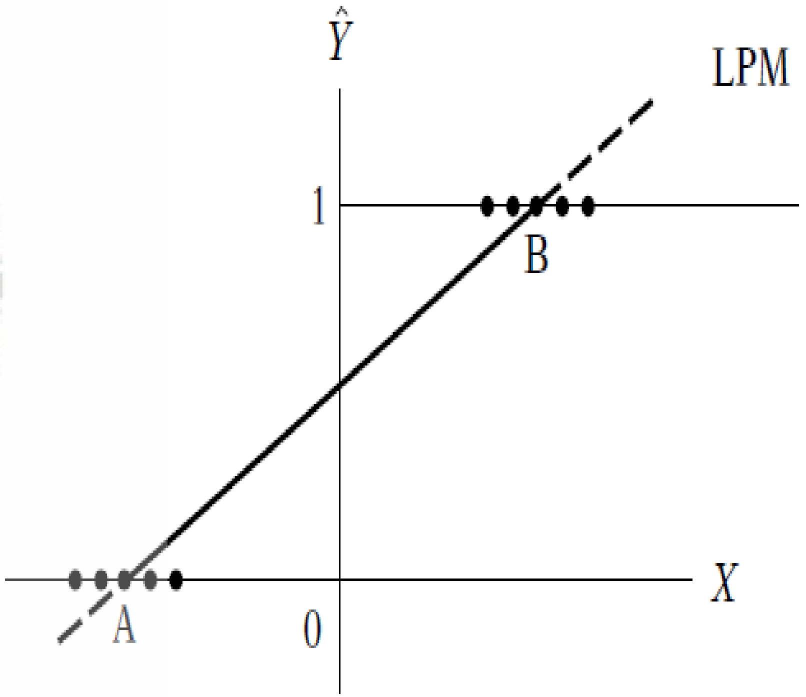
- What to do: set all negative as 0 and a those greater than 1 as 1?
- Implausible to suggest that small firms will never pay dividend and large firms will always pay dividends



Issues with LPM

Diminishing utility of R^2 as a goodness of fit measure

- All the Y values will be on a line $Y = 0$ or $Y = 1$
- The conventional LPM is not expected to fit well with such observations, except those cases where all the observations are scattered closely around points A and B
- Both logit and probit approaches are able to overcome the limitation of LPM that it produces values less than 0 and more than 1



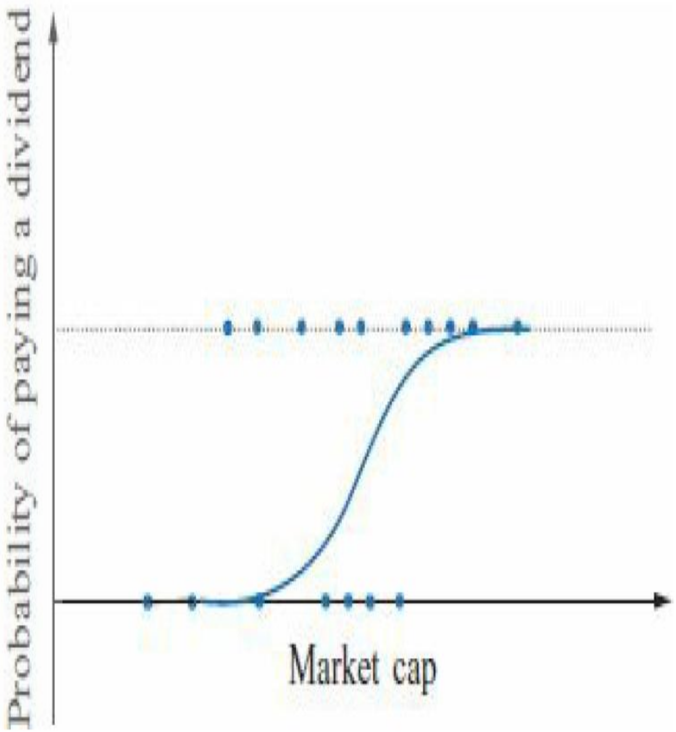
Introduction to Logit Model

Introduction to Logit Model

The logit (and probit) approaches overcome the limitations of the regression model by transforming to a function so that fitted values are bounded within (0,1) interval

- The fitted function looks like an S-shape curve

- The logistic function for a random variable z is: $F(z_i) = \frac{(e^{z_i})}{(1+e^{z_i})} = \frac{1}{(1+e^{-z_i})}$

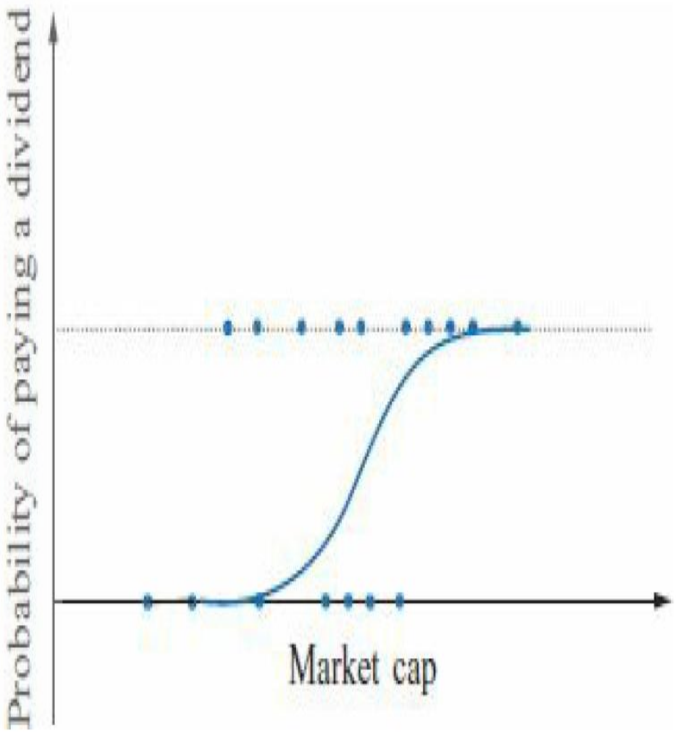


Introduction to Logit Model

The logit (and probit) approaches overcome the limitations of the regression model by transforming to a function so that fitted values are bounded within (0,1) interval

- Here *F* is the cumulative logistic distribution

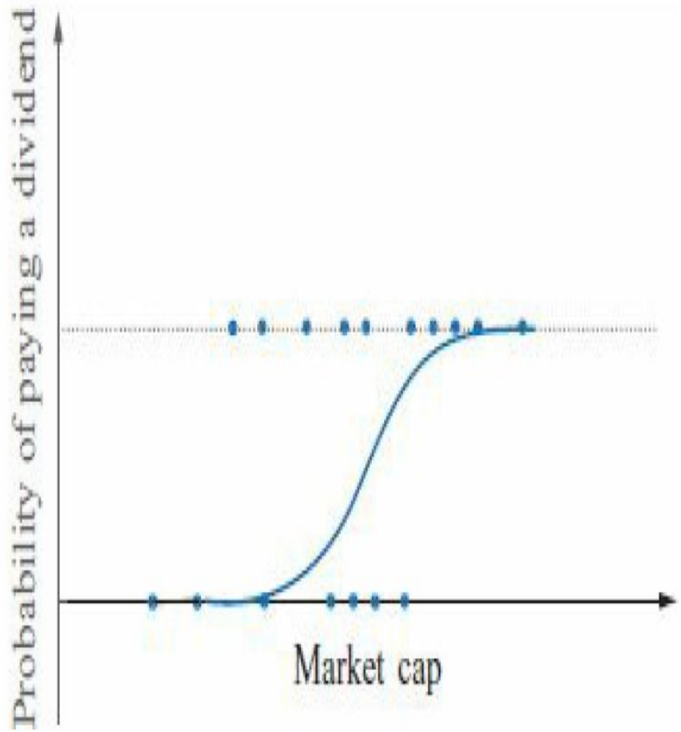
- The final logit model: $P_i(y_i = 1) = \frac{1}{(1+e^{-(\beta_1+\beta_2 x_{2i}+\beta_3 x_{3i}+\cdots+\beta_k x_{ki}+u_i)})}$



Introduction to Logit Model

$$P_i(y_i = 1) = \frac{1}{(1+e^{-(\beta_1+\beta_2 x_{2i}+\beta_3 x_{3i}+\cdots+\beta_k x_{ki}+u_i)})}$$

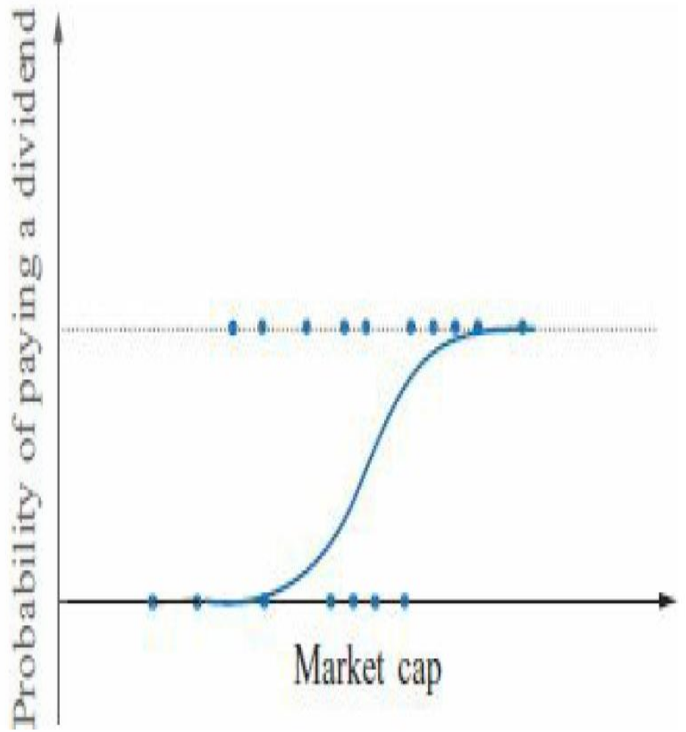
- Model asymptotically touches 0 ($z \to -\infty$) and 1 ($z \to \infty$)
- Is this model linear? Hence, not amenable to OLS estimation
- The model would predict that the probability, e.g., probability of bank loan default (dependent variable = y)



Introduction to Logit Model

$$P_i(y_i = 1) = \frac{1}{(1+e^{-(\beta_1+\beta_2 x_{2i}+\beta_3 x_{3i}+\cdots+\beta_k x_{ki}+u_i)})}$$

- $P\ (y = 1)$, then $P(y = 0) = 1 - P(y = 1)$
- Here independent variables are x_{2i}, x_{3i}, x_{4i}, x_{5i}, and so on
- This is essentially a non-linear transformation of the model to produce consistent probability results

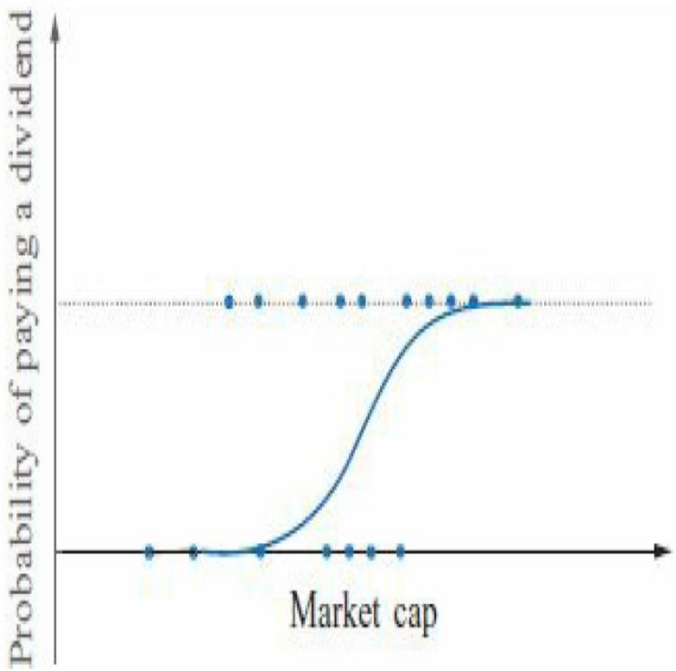


Understanding the Logit Function

Understanding the Logit Function

$$P_i(y_i = 1) = \frac{1}{(1+e^{-(\beta_1+\beta_2 x_{2i}+\beta_3 x_{3i}+\cdots+\beta_k x_{ki}+u_i)})}$$

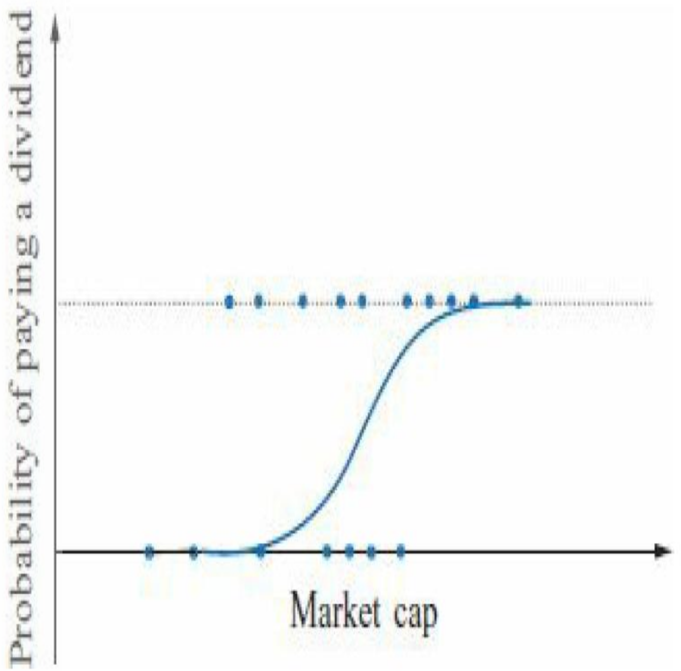
- Here extremely low and negative values of the linear function $\beta_1 + \beta_2 x_{2i} + \beta_3 x_{3i} + \cdots + \beta_k x_{ki}$ would predict No dividend (or non-default cases) with a high probability or $P_i(y_i = 0)$



Understanding the Logit Function

$$P_i(y_i = 1) = \frac{1}{(1+e^{-(\beta_1+\beta_2 x_{2i}+\beta_3 x_{3i}+\cdots+\beta_k x_{ki}+u_i)})}$$

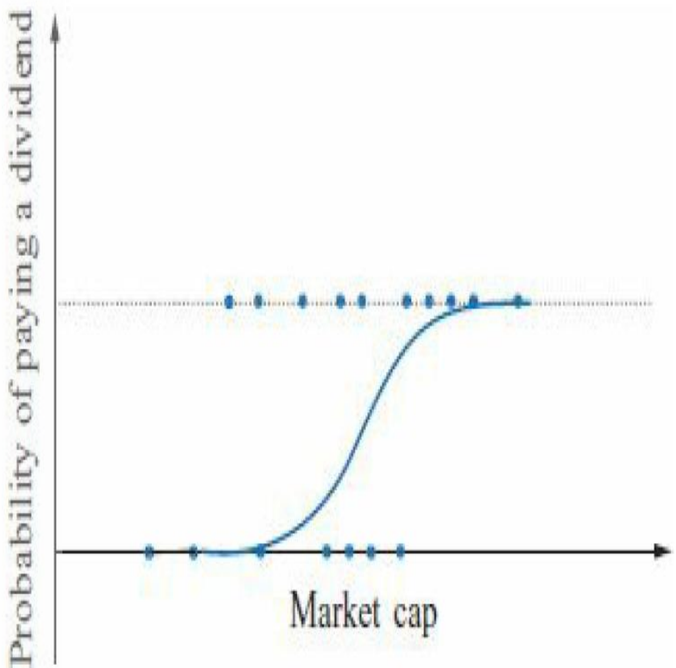
- Extremely high and positive values of the linear function $\beta_1 + \beta_2 x_{2i} + \beta_3 x_{3i} + \cdots + \beta_k x_{ki}$ would predict dividend payment (or default cases) with high probability or $P_i(y_i = 1)$



Understanding the Logit Function

$$P_i(y_i = 1) = \frac{1}{(1+e^{-(\beta_1+\beta_2 x_{2i}+\beta_3 x_{3i}+\cdots+\beta_k x_{ki}+u_i)})}$$

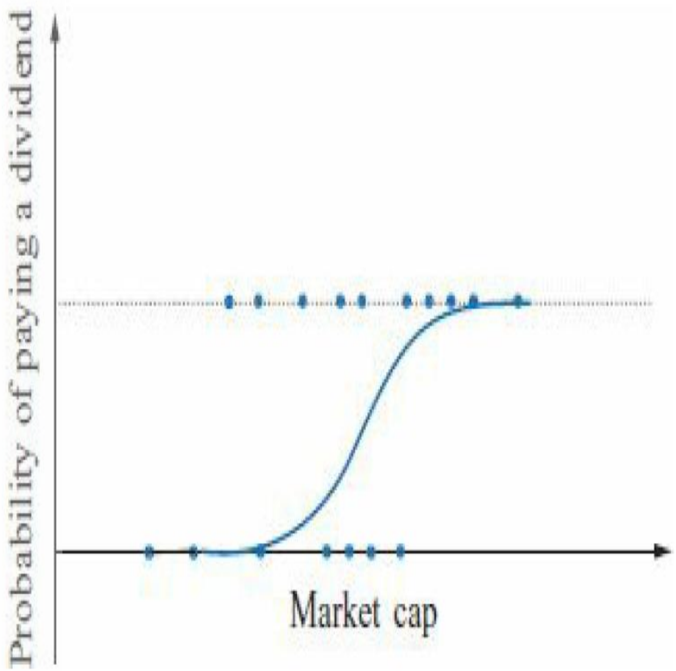
- This can also be expressed in the form of Odds
- $\text{Odds} = \frac{P(y=1)}{P(y=0)};$
- Odds > 1 if $y = 1$ is more likely
- Odds < 1 if $y = 0$ is more likely



Understanding the Logit Function

$$P_i(y_i = 1) = \frac{1}{(1+e^{-(\beta_1+\beta_2 x_{2i}+\beta_3 x_{3i}+\cdots+\beta_k x_{ki}+u_i)})}$$

- If we substitute the logit function in Odds equation, then
- $\mathrm{Odds} = \exp^{(\beta_1+\beta_2 x_{2i}+\beta_3 x_{3i}+\cdots+\beta_k x_{ki}+u_i)}$ or
- $\ln(\mathrm{Odds}) = \beta_1 + \beta_2 x_{2i} + \beta_3 x_{3i} + \cdots + \beta_k x_{ki} + u_i$
- The higher this logit (or $\ln(\mathrm{Odds})$) form, the higher the probability for $P_i(y_i = 1)$



Thresholding

Thresholding

The outcome of the regression model is a probability

- In real life, you would want to make a binary prediction, e.g., default or no default
- For this, we may consider a threshold value "t"
- If P(Default = 1) >= t, then predict a default case
- If P(Default = 0)< t, then predict a non-default case

Thresholding

What value should we select for "*t*"? What kind of error do you prefer?

- Given a *t* value, one can make two types of errors: (1) predict default, but the actual outcome is non-default: false positive; and (2) predict non-default, but the actual outcome is default: false negative
- A large threshold (e.g., $t = 0.8$) will have a very small probability of predicting defaulters and, at the same time, a high probability of predicting cases as non-defaulters

Thresholding

What value should we select for "*t*"? What kind of error do you prefer?

- A small threshold (e.g., $t = 0.1$) will have a very large probability of predicting defaulters and, at the same time, a small probability of predicting cases as non-defaulters
- An aggressive bank would like to have high t values to increase the possibility of converting a loan

Thresholding

What value should we select for "t"? What kind of error do you prefer?

- A more conservative bank may choose a very low t value to select those loan applications with a very low probability of default
- In the absence of any threshold, $t = 0.5$ is the correct value to pick

Classification Matrix

Selecting a Threshold: Confusion/Classification Matrix

	Predicted = 0 (Non-Default)	Predicted = 1 (Default)
Actual = 0	True Negatives (TN)	False Positives (FP)
Actual = 1	False Negatives (FN)	True Positives (TP)

Let us compute two outcome measures to determine what kind of errors we are making

- $\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} = \text{TP rate}$
- $\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} = \text{TN rate}$

Selecting a Threshold: Confusion/Classification Matrix

Let us compute two outcome measures to determine what kind of errors we are making

- $\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} = \text{TP rate}$
- $\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} = \text{TN rate}$
- A model with higher *t* will have lower sensitivity and higher specificity
- A model with lower *t* will have higher sensitivity and lower specificity

Selecting a Threshold: Confusion/Classification Matrix

- Overall accuracy $= \frac{(\mathrm{TN}+\mathrm{TP})}{N}$, where N = number of observations

- Overall error rate $= \frac{(\mathrm{FP}+\mathrm{FN})}{N}$

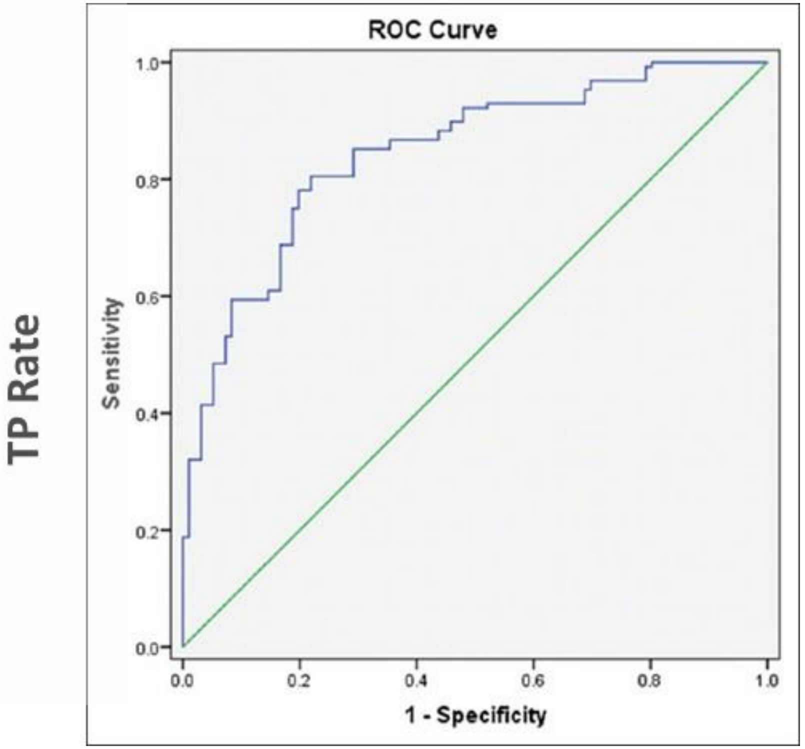
- False negative error rate $= \frac{\mathrm{FN}}{(\mathrm{TP}+\mathrm{FN})}$

- False positive error rate $= \frac{\mathrm{FP}}{(\mathrm{TN}+\mathrm{FP})}$

Receiver Operating Characteristic (ROC) Curve

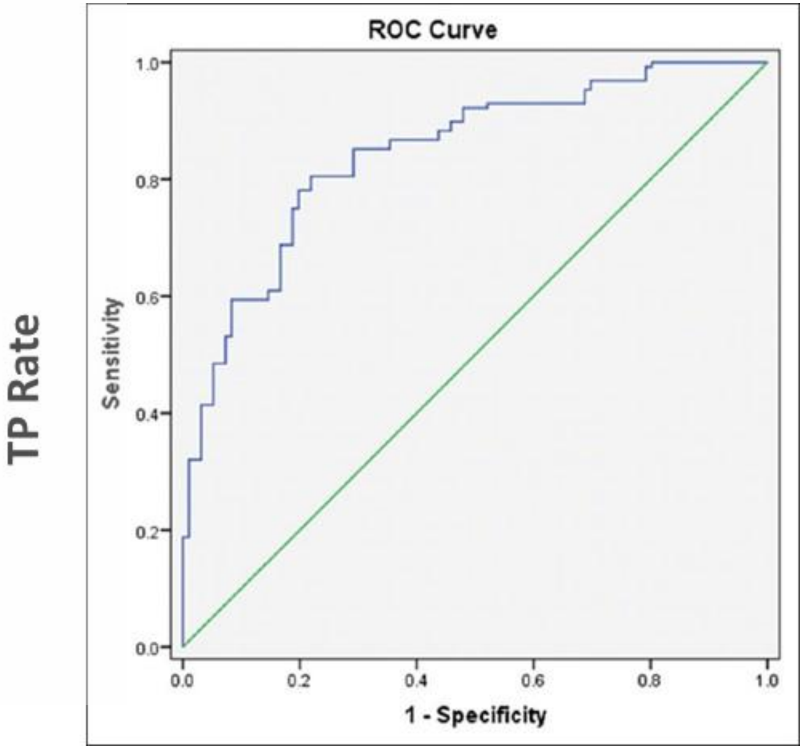
Receiver Operator Characteristic (ROC) Curve

- True positivity (TP) rate on the *y*-axis, i.e., the proportion of default correctly predicted
- False positive on the *x*-axis, i.e., the proportion non-default incorrectly predicted as default cases
- The curve shows how these two measures vary with different threshold values



Receiver Operator Characteristic (ROC) Curve

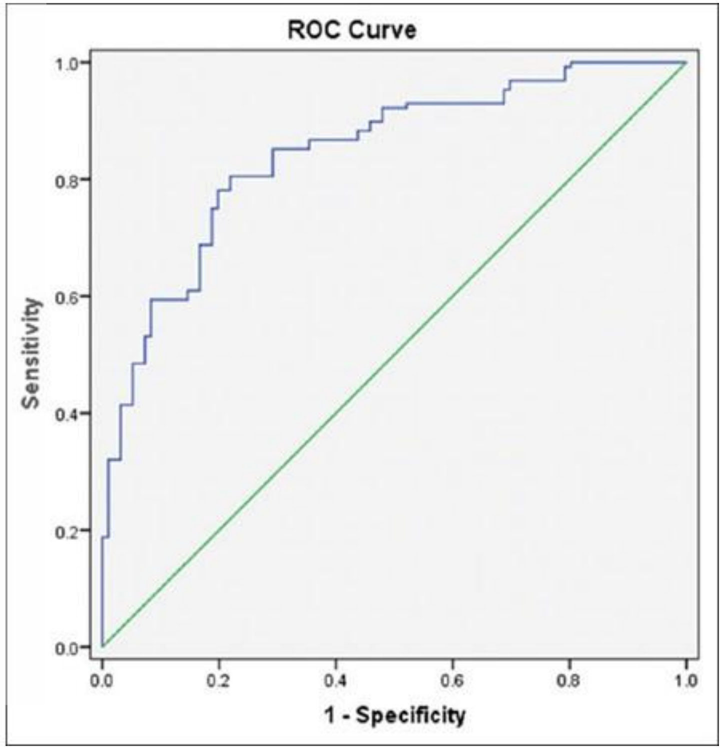
- For t = 1, TP = 0, and FP = 0 → will not be able to predict any default cases but correctly predict all the non-default cases
- For t = 0, TP = 1, and FP = 1 → will be able to correctly predict all the default cases but incorrectly predict all the non-default cases
- As we move from t = 1 to t = 0, different combinations of TP and FP are obtained



Receiver Operator Characteristic (ROC) Curve

- ROC curve captures all the complete threshold behavior
- High threshold: high specificity and low sensitivity
- Low threshold: low specificity and high sensitivity
- Thus, it is a tradeoff between cost in failing to detect default cases vs. incorrectly considering non-default cases as defaulters

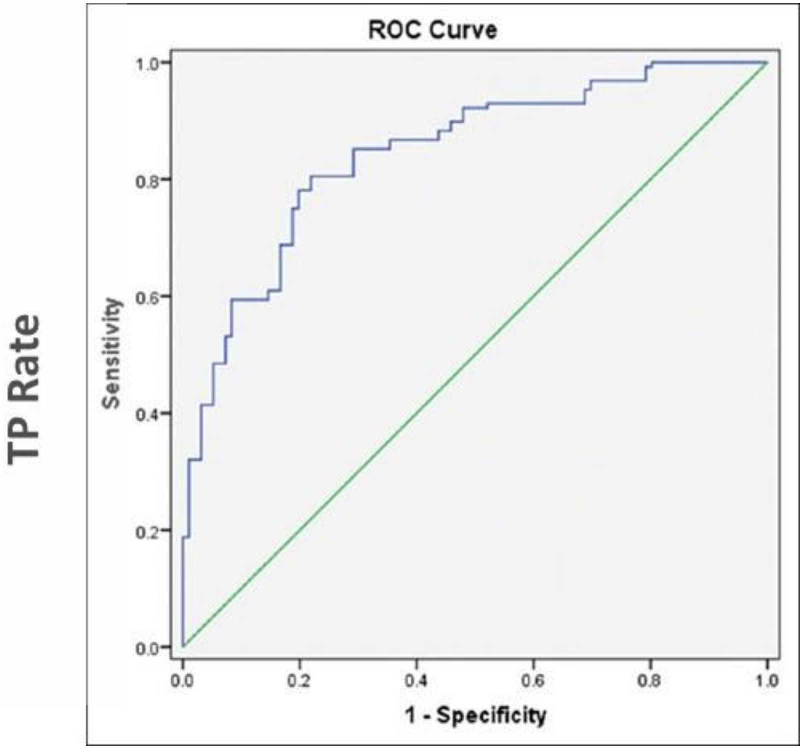
TP Rate



Receiver Operator Characteristic (ROC) Curve

- A 100% score area under the curve will indicate complete accuracy, i.e., all the observations are correctly identified
  TP = 1 and FP = 0

- A 50% score will indicate random guessing, that is, half TP = 0.5 and TN = 0.5 (FP = 0.5)



Parameter Interpretation

Parameter Interpretation

Parameter Interpretation

Unlike LPM, it is incorrect to state that 1 unit increase in x_{2i} will cause 100*β_2% increase in the probability of y_i = 1

- For logit model, we calculate $\frac{dP_i}{dx_{2i}}$; this works out to $\beta_2 F(x_{2i})(1 - F(x_{2i}))$ for the logit model
- So, a 1-unit increase in x_{2i} will increase the probability of y_i= 1 by $\beta_2 F(x_{2i})(1 - F(x_{2i}))$
- Usually, these marginal/incremental impacts are evaluated at mean values

Parameter Interpretation

Example: $P_i(y_i = 1) = \frac{1}{(1+e^{-(\beta_1+\beta_2 x_{2i}+\beta_3 x_{3i}+\cdots+\beta_k x_{ki}+u_i)})}$

- $F(z_i) = \widehat{P_i} = \frac{1}{(1+e^{-(0.1+0.3x_{2i}-0.6x_{3i}+0.9x_{4i})})}$;
- $\beta_1 = 0.1; \beta_2 = 0.3;\ \beta_3 = -0.6;\ \beta_4 = 0.9$
- What is $F(z_i)$? Given $\bar{x}_2 = 1.6, \bar{x}_3 = 0.20$, and $\bar{x}_4 = 0.10$?
- Marginal effects of x_{2i} = $\beta_2 F(x_{2i})(1 - F(x_{2i}))$

Parameter Interpretation

Example: $F(z_i) = \widehat{P_i} = \frac{1}{(1+e^{-(0.1+0.3x_{2i}-0.6x_{3i}+0.9x_{4i})})} = \frac{1}{1+e^{-0.55}} = 0.63$

- Thus, a 1-unit increase in x_{2i} will increase the probability of y_i by 0.3*0.63*(1 − 0.63) = 0.07
- Similarly, for x_{3i}, −0.6*0.63*(1 − 0.63), and x_{4i}, 0.9*0.63*(1 − 0.63)
- Sometimes, these are also called marginal effects

Probit Model
Maximum Likelihood Estimation (MLE)
Goodness-of-Fit Measures

Probit Model

- The probit model uses cumulative normal distribution: $F(z_i) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z_i} e^{-(z_i^2)/2}\ dz$
- Model asymptotically touches 0 ($z \rightarrow -\infty$) and 1 ($z \rightarrow \infty$)
- Marginal impact of unit change on an explanatory variable x_{2i} is given as $\beta_2 F(z_i)$, where β_2 is the parameter attached to x_{2i};

$$z_i = \beta_1 + \beta_2 x_{2i} + \beta_3 x_{3i} + \cdots + \beta_k x_{ki} + u_i$$

- Both logit and probit models give similar results; differences may occur when data is extremely imbalanced

Maximum Likelihood Estimation (MLE) of Logit/Probit Models

These are non-linear models, hence cannot be estimated with a simple OLS method

- They are estimated with MLE
- In MLE, parameters are chosen to maximize a log-likelihood function
- The log-likelihood function obtains the population estimates that maximize the joint probability of observed sample/sample estimates

Goodness-of-Fit Measures

Conventional R^2 and $\mathbf{adj.}-R^2$ measures do not work well with these models

MLE aims to maximize the log-likelihood function (LLF) and do not minimize RSS

(1) % of y_i values correctly predicted

(2) % of y_i = 1 values correctly predicted + % of y_i = 0 values correctly predicted

Goodness-of-Fit Measures

Conventional R^2 and $\mathrm{adj.}-R^2$ measures do not work well

(3) $\mathrm{Pseudo} - R^2 = 1 - \frac{\mathrm{LLF}}{\mathrm{LLF}_0}$, where LLF is the maximized value of the log-likelihood function for the logit and probit models, and LLF0 is the value of the log-likelihood function for a restricted model

Summary and Concluding Remarks

Summary and Concluding Remarks

- Among supervised learning algorithms, classification algorithm is a very important tool employed in the finance domain for applications such as credit scoring of loan applications
- Classification algorithms are very often implemented through Logit/Probit class of models; these are very simple yet powerful models
- These models account for a number of shortcomings of linear probability models: (a) non-normality and heteroscedasticity of error terms; (b) values of the dependent variable (probability) exceeding the 0–1 range; and (c) diminishing utility of conventional measures of goodness-of-fit (e.g., R^2)

Summary and Concluding Remarks

- Limited dependent variable models (e.g., Logit model) employ cumulative probability functions (e.g., logistic function)
- These models, although non-linear, are very useful for modeling limited dependent variables that are probabilistic in nature
- In the case of the logit model, the logit function is essential the odds ratio
- Since the estimated variable is in the form of probabilities, the thresholding process is needed to convert these probabilities into limited outcomes (e.g., Yes/No)

Summary and Concluding Remarks

- The conventional measures of goodness-of-fit (e.g., R^2) are not very useful for such models
- These measures are evaluated on their ability to accurately classify observations correctly
- For such purposes, a confusion/classification matrix is often employed
- The receiver operator characteristic (ROC) curve provides another useful tool to examine the efficiency of these models, and also facilitates the selection of thresholding values

Summary and Concluding Remarks

- Unlike simple linear models, the parameter estimates are interpreted in a different manner
- Marginal effects are computed to interpret the coefficients and their relationship with the dependent variable
- Other models (e.g., probit model) remain identical in all other aspects, except that a different cumulative probability function is considered (normal distribution in case of probit)
- Since the model is non-linear in nature, OLS cannot be employed for estimation; maximum likelihood method is often employed to estimate these models

Thanks!

Introduction

- Application of classification algorithm in the prediction of security prices
- Revisiting the ABC case study
- Logit/Probit modeling
- Training the model and testing the model
- Model performance evaluation
- Summary and concluding remarks

Case Study: ABC Stock Price Forecasting

Case Study: Stock Price Prediction

- Stock price prediction or stock return prediction is an attempt to determine the future value of a company based on an analysis of factors, which impact its price movement
- There are a number of factors that help in predicting stock prices
- These can be macroeconomic factors like the state of the country's economy, growth rate inflation, etc.
- There are also other factors that are more specific to a stock like profit margin, debt to equity issues, sales of a company, etc.

Case Study: Stock Price Prediction

So, we are given the data for stock market price for ABC company, along with Nifty and Sensex (market indices). We are also given the data of dividend announcement and a sentiment index.

Date	Price	ABC	Sensex	Dividend Announced	Sentiment	Nifty
03-01-2007	718.15	0.079925	0.073772	0	0.048936	0.095816
04-01-2007	712.9	–0.00731	0.021562	0	–0.05504	0.009706
05-01-2007	730	0.023987	–0.02441	0	0.019135	–0.03221
06-01-2007	788.35	0.079932	0.012046	0	0.080355	0.011205
07-01-2007	851.4	0.079977	–0.0013	0	0.094038	–0.0004
10-01-2007	919.5	0.079986	0.019191	1	0.015229	0.030168
11-01-2007	880	–0.04296	–0.04025	0	–0.07217	–0.04966
12-01-2007	893.75	0.015625	0.036799	0	0.01396	0.020999
13-01-2007	875	–0.02098	–0.00845	0	0.057518	–0.01164
14-01-2007	891	0.018286	0.004858	1	0.008828	0.020714
17-01-2007	819.75	–0.07997	–0.01228	0	–0.12395	–0.00962
……	……	……	……	……	……	……
……	……	……	……	……	……	……

Case Study: Stock Price Prediction

- Consider a portfolio manager who has built a model for a particular stock
- The manager wants to predict whether in the next period the ABC stock price returns for this stock will go up or down
- The data starts from 2007 and goes till 2019, so we have approximately 13 years of data
- We have daily returns of ABC or a change in the price of ABC in column B. Next, we have a daily return on Sensex in column C and a daily return on Nifty in column D.

Case Study: Stock Price Prediction

- Sensex and Nifty are the two main stock indices used in India
- They are benchmark Indian stock market indices that represent the weighted average of the largest Indian companies
- So, Sensex represents average of 30 largest and most actively traded Indian companies
- Similarly, Nifty represents a weighted average of 50 largest Indian companies

Summary

The following tasks need to be performed

- Create a dummy variable that is 1 when stock prices go up and create a dummy variable that is “0” when stock prices go down
- Segregate the data into test and train datasets
- Train and build the model using simple logit/probit classification algorithms using market index as the independent variable, and up/down dummy as the dependent variable

Summary

The following tasks need to be performed

- Evaluate the in-sample performance and out-of-sample performance of the model
- Compute the marginal effects of the independent variable
- Visualize the performance of these models using the ROCR curve
- Examine the classification accuracy of the model and compare it with a similar linear probability model
-

Data Input and Exploration

Data Input and Exploration

- In this video, we will start with the implementation of the classification algorithms using ABC Case study Data
- First, we will set the working directory, then we will read the data
- Lastly, we will create the binary response variable: ‘1’ for positive returns and ‘0’ for negative returns

Summary

- We started our analysis with setting the working directory
- Next, we loaded the relevant package libraries
- Then we read the data from the working directory
- Lastly we created a new ‘updown’ binary response variable, which is ‘1’ when returns are positive and ‘0’ when returns are negative

Creation of Test and Train Datasets

Creation of Test and Train Datasets

- In this video, we will create the test and train sample datasets
- Then we will examine the distribution of our binary response variable in 1's and 0's

Summary

- First, we filtered the observations after 2006 and cleaned our data
- Next, we randomly selected 80% observations as training dataset and remaining 20% as test dataset
- Lastly, we tested the proportion of 1's and 0's in the parent dataset, test dataset, and train dataset
- The distribution of 1's and 0's is fairly similar for all the three datasets

Training the Linear Probability Model (LPM) Algorithm

Training the LPM Algorithm

- In this video, we will train an LPM algorithm with the training dataset
- Next, we will compute the classification/confusion matrix
- Final, using the classification/confusion matrix, we will compute various performance measures, i.e., accuracy, specificity, and sensitivity

Summary

- We trained an LPM algorithm using the training dataset
- Using the fitted results, we converted them into 1’s and 0’s using thresholding values of 0.4, 0.6, and 0.8
- Lastly, using the classification/confusion matrix, we computed three performance parameters, namely, accuracy, specificity, and sensitivity

Training the Logit/Probit Algorithms

Training the Logit/Probit Algorithms

- In this video, we will train the Logit/Probit classification algorithms using the training dataset
- Next, we will compute the in-sample performance evaluation measures
- We will also compute the marginal effects of the independent variable on the dependent variable
- Lastly, we will evaluate and compare the performance of these algorithms on parameters, namely, accuracy, specificity, and sensitivity

Summary

- We trained our classification algorithms using the training dataset
- Next, we computed the Pseudo R-square measure and also computed the marginal effects
- Lastly, we evaluated the performance of these algorithms on three parameters of sensitivity, specificity, and accuracy, using classification matrix at threshold values of 0.4, 0.6, and 0.8
- The performance of all the algorithms appear to be close to each other; this is ascribed to the fairly symmetric distribution of 1's and 0's in the training dataset

Visualizing the Performance

Visualizing the Performance

- In this video, we will compare the performance of the three trained classification algorithms (linear, logit, and probit objects) using correlation measure and through visualization

Summary

- We computed the correlation across the fitted values for the three classification algorithms (linear, logit, and probit)
- The correlations appear to be very high
- Next, we visualized the performance of the algorithms on parameters of accuracy, sensitivity, and specificity for the three threshold values of 0.4, 0.6, and 0.8
- While the performance of these algorithms appear to be close, logit model appears to offer the best fit, followed by the probit, and then the linear model

Receiver Operating Characteristic (ROC) Curve

ROC Curve

- In this video, we will compare the performance of the three trained classification algorithms (linear, logit, and probit objects) with the help of ROC curve

Summary

- We plotted the ROC curve and examined the performance of the three trained classification algorithms
- Area under the curve (AUC) appears to be identical for all the three algorithms; this is ascribed to the extremely high correlation in the fitted objects of these models

Defining the Objective Performance Function

Defining the Objective Performance Function

- In this video, we will develop a simple machine learning system that will help the computer learn how to select the best classification algorithm across a class of algorithms
- We will create a suitable user defined performance function to analyze the performance of these algorithms

Summary

- We created an optimization function, which included the arguments, namely, fitted values, actual values, and simulated threshold values
- These values are employed to compute accuracy, sensitivity, and specificity parameters through classification matrix
- The final performance object is a simple average of these three parameters (i.e., accuracy, sensitivity, and specificity)
-

Creating Performance Objects

Creating Performance Objects

- In the previous video, we defined our performance objective function; in this video we will simulate 1000 threshold values and calculate the performance object values for all the three classification algorithms using these threshold values

Summary

- We created three performance objects for the three classification algorithms, namely logit, probit, and linear
- We simulated 1000 performance object values using our performance objective function for all the three algorithms (linear, logit, and probit)

In-sample Performance Evaluation

In-sample Performance Evaluation

- In the previous video, we computed 1000 performance object values for the three classification algorithms
- In this video we will compare the performance of these three classification algorithms through visualization

Summary

- We plotted 1000 performance object values for our three classification algorithms, namely, linear, logit, and probit
- We found that for most of the threshold values, the logit model algorithm works best, closely followed by probit model algorithm, and lastly the linear model algorithm
- Lastly, we extracted the best fit model and the corresponding threshold value

Out-of-Sample Prediction

Out-of-Sample Prediction

- In this video, we will start with out-of-sample prediction
- We will use the trained algorithms for our linear, logit, and probit models and predict using test data set
- Lastly, we will compute the correlations across the predicted values between the three algorithms

Summary

- We performed the prediction on the test data using our trained algorithms for linear, logit, and probit models
- We found that the correlation across the predicted values are very high; in fact the correlation between logit and probit predicted values are 99%, and the correlation with linear model predicted values are more than 90%
- This is ascribed to the fact that correlations across predicted objects are very high, and the distribution of 1’s and 0’s is highly symmetric in our test and training datasets

Out-of-Sample Prediction: ROC Curve

Out-of-Sample Prediction: ROC Curve

- In the previous video, we performed prediction with trained algorithms, using the test datasets
- In this, video, we will visualize and compare the performance of the three trained algorithms, using ROC curve and also compute area under the ROC curve

Summary

- We plotted ROC curves for all the three classification algorithms for linear, logit, and probit models
- The performances as per the ROC curve are quite similar with identical area under the curve (ROC)
- This is ascribed to the high correlation across fitted objects and symmetric nature of 1’s and 0’s in our test and training datasets
- In the next video, we will simulate 1000 threshold values and compute the performance object values

Out-of-Sample Prediction: Performance object

Out-of-Sample Prediction: Performance object

- We have already set-up a performance object, which is the average of three parameters: accuracy, sensitivity, and specificity
- Using our predicted values for all the three algorithms, we will compute the performance object values for the 1000 simulated threshold values

Summary

- In this video, we computed the values of our performance object using 1000 simulated threshold values for all the three algorithms, i.e., linear, logit, and probit
- In the next video, using these values of the performance object, we will visualize and compare the out-of-sample performance of the three algorithms

Out-of-Sample Prediction: Performance Evaluation and Visualization

Out-of-Sample Prediction: Performance Evaluation and Visualization

- In the previous video, we have simulated 1000 performance object values using our trained algorithms with the test data
- In this video, using these performance object values, we will visualize and compare the performance of the three trained algorithms

Summary

- To summarize, we plotted our simulated performance object values
- For most of the threshold region, the logit model offers the best prediction, closely followed by the probit and linear models
- We also extracted the details corresponding to the best performance object value, including its threshold level

Summary and Concluding Remarks

Summary and Concluding Remarks

- ABC stock price up/down movements are modelled using logit/probit classification algorithms
- The model is trained using the training dataset and is examined on various measures of model performance evaluation
- Fitted modelled is examined visually as well

Summary and Concluding Remarks

- The model is tested using test dataset and various measures of out of sample fit are examined
- Marginal effects of these independent variables are computed
- The performance of this model is compared with a similar linear probability model

Thanks!