平成29年度
卒業論文

ウェブページアクセスにおける様々な機械学習モデルを用いたリスティング広告の評価

佐藤 芙柚

(学籍番号：14232036)

九州工業大学　情報工学部
電子情報工学科

小田部研究室
平成30年 2月 16日

目 次

第 1 章　序論 **1**
1.1 はじめに 1
1.2 機械学習 1
1.2.1 教師あり学習 1
1.2.2 教師なし学習 4
1.2.3 ニューラルネットワーク 4
1.3 リスティング広告 7
1.4 使用言語・アプリケーション 8
1.4.1 R 言語 8
1.4.2 Neural Network Console 8
1.4.3 Mathematica 8
1.5 研究背景 8
1.6 本研究の目的 9

第 2 章　研究方法 **11**
2.1 R 言語による重回帰を用いたモデル 14
2.2 Neural Network Console を用いた方法 14
2.2.1 リカレントニューラルネットワーク 15
2.2.2 LSTM ネットワーク 16
2.3 Mathematica 17

第 3 章　結果 **18**
3.1 R 言語による重回帰を用いたモデルの結果 18
3.2 Neural Network Console 20
3.2.1 リカレントニューラルネットワーク 20
3.2.2 LSTM ニューラルネットワーク 22
3.3 Mathematica 25
3.3.1 ランダムフォレスト 25
3.3.2 ニューラルネットワーク 26
3.4 各モデルの精度評価 28

第 4 章　リスティング広告の評価 **29**

第 5 章　まとめ **32**

謝辞 **33**

参考文献 **34**

表 目 次

3.1 相対誤差まとめ 28

図 目 次

1.1 機械学習の流れ 2
1.2 ランダムフォレストのモデル 3
1.3 単純なモデル 4
1.4 階層型ニューラルネットワークモデル 5
1.5 RNN（Recurrent Neural Network）のネットワーク構造 6
1.6 LSTM（Long Short Term Memory）のネットワーク構造 6
1.7 IoM のリスティング広告 7
1.8 IoM のホームページトップ 9

2.1 精度評価に用いるデータ 12
2.2 予測に用いる学習データと評価データ 13
2.3 R の使用データ 14
2.4 AIC による回帰式の決定のためのコード 14
2.5 NNC の使用データ（分割データファイル）の内容 15
2.6 NNC の使用データ（学習ファイル）の内容 15
2.7 リカレントニューラルネットワークのネットワーク構成 16
2.8 LSTM ネットワークのネットワーク構成 17
2.9 Mathematica における機械学習のコード 17

3.1 step 関数を用いて決定した回帰式 18
3.2 精度評価のための重回帰式による実測データと予測データの比較グラフ 19
3.3 リスティング広告評価のための重回帰式による実測データと予測データの比較グラフ 20
3.4 精度評価のための NNC（RNN）の学習曲線 21
3.5 精度評価のための NNC の実測データと予測データの比較グラフ . . 21
3.6 リスティング広告評価のための NNC（RNN）の実測データと予測データの比較グラフ 22
3.7 精度評価のための NNC（LSTM）の学習曲線 23
3.8 精度評価のための NNC の実測データと予測データの比較グラフ . . 23
3.9 リスティング広告評価のための NNC の実測データと予測データの比較グラフ 24

3.10 精度評価のための Mathematica ランダムフォレストの実測データと予測データの比較グラフ 25
3.11 リスティング広告評価のための Mathematica ランダムフォレストの実測データと予測データの比較グラフ 26
3.12 精度評価のための Mathematica ニューラルネットワークの実測データと予測データの比較グラフ 27
3.13 リスティング広告評価のための Mathematica ニューラルネットワークの実測データと予測データの比較グラフ 27

4.1 全モデルの予測結果とリスティング広告使用時の実測値 29
4.2 セッション数と売上推移 30

第1章　序論

1.1　はじめに

インターネットが普及している現代において, IT 技術というものは切っても切れない存在である. そんな中で, AI（人工知能）は革新的に技術が進化している. AI の中のひとつである機械学習は, 1959 年にアメリカの Arthur Lee Samuel がボードゲームのチェッカーにおいて, より強いコンピュータを開発するために過去データに基づく予測エンジンを作ったことが始まりとされている. そして 2016 年に人工知能である「Alpha Go」が世界トップレベルの囲碁棋士に勝利したことをきっかけに脚光を浴びた. 他にも, google 翻訳でニューラルネットワークを使用することで精度を大きく改善することに成功したりと私たちの身の回りにも進出してきている. ビジネスへの応用も進んでおり, 製造, 流通, 運輸・輸送, 医療, 金融, エネルギー・住宅, 教育, 人材, 建設, 公共といった様々な分野や業界でも積極的に取り入れていく動きになってきている [1]

1.2　機械学習

機械学習の基本的な定義は,「アルゴリズムとして明示的に解放が与えられてないタスクに対して, そのタスクを遂行するためのモデルを, 学習データから構築すること」[2] である. その基本的な流れを図 1.1 に示す. 図 1.1 のように, まず実世界のデータを整え, 数値やラベルの集合データに加工する. その後, 学習データを入力として与え, 機械はそれに対してどのようなアルゴリズムを適用すべきか考える. その学習の結果得られたモデルを, 新たな入力データに対してどのように適用するか考え, 結果を出力する. また, 機械学習は教師あり学習と教師なし学習, そして両方を含むニューラルネットワークがある.

1.2.1　教師あり学習

教師あり学習の学習データは, 式 (1.1) のような入力データに対応するベクトル $\boldsymbol{x_i}$ と正解の情報 y_i のペアで構成される. つまり, 正解のある学習データを用いる.

$$\{(\boldsymbol{x_i}, y_i)\}, \quad i = 1, \cdots, N \tag{1.1}$$

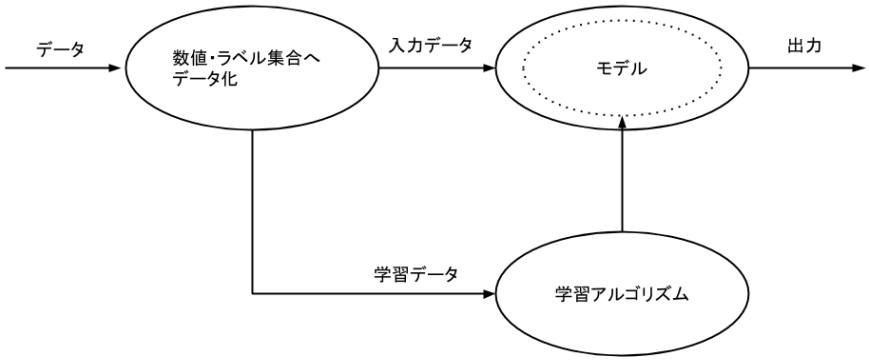

図 1.1: 機械学習の流れ. まず学習データが入力されると数値データやラベルの集合にデータ化され, 学習アルゴリズムに沿ってモデルが作成される. 次に入力されるデータにこのモデルを適用することで出力結果が得られる.

ここで, N は学習データの総数, 添字 i は学習データの中の i 番目であることを示す. また, 入力ベクトル $\boldsymbol{x_i}$ は, 次元数 d の固定長ベクトルであるとする.

$$\boldsymbol{x_i} = (x_{i1}, \cdots, x_{id}) \tag{1.2}$$

入力ベクトルを, 例えば（性別, 身長, 体重）とする. 教師あり学習は, このような入力ベクトルと正解情報, 例えばテストの点数をペアとして学習データとする. この学習データから, 入力 x を正解 y にする関数 c を学習する. そしてその関数 c を用いて, 未知のデータに対し識別や回帰を行い出力することを目的とする.

回帰

回帰は, 入力から予測される出力値を求める手法のことである. 学習データの説明変数 x_i（入力ベクトル）を用いて, 目的変数 y（正解の情報）にできるだけ誤差がないように関数を決定する. 特に, 説明変数が複数あるものを重回帰といい, 式 (1.3) で表される. b_i を偏回帰係数という.

$$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_i x_i \quad i = 1, \cdots, N \tag{1.3}$$

また, データの総数を N, i 番目のサンプル予測値を f_j, 実測値を y_i とする. 回帰式の精度を表す指標として標準偏差 (RMSE) があり, 式（1.4）に示す.

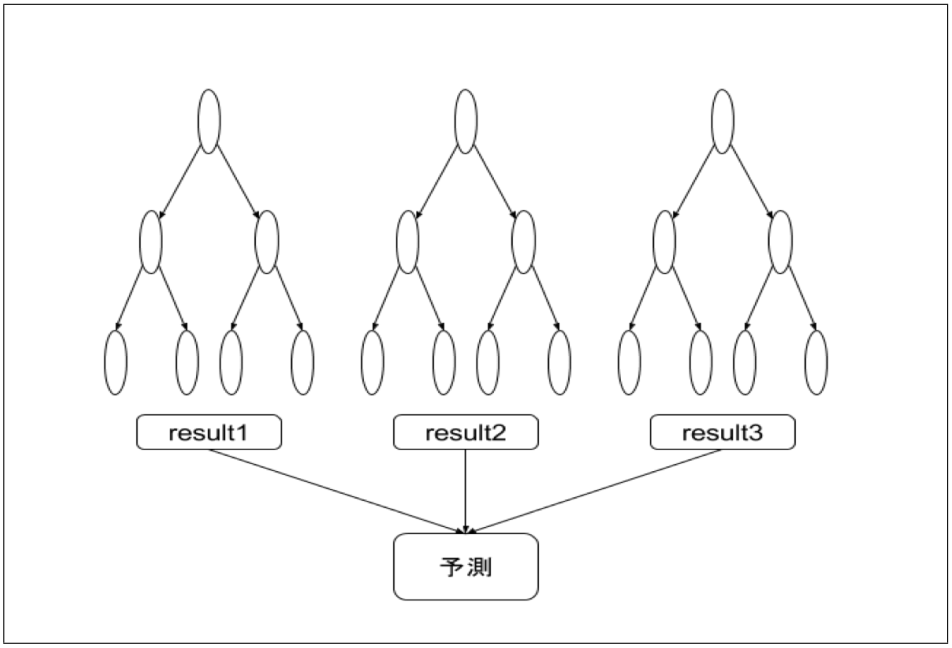

図 1.2: ランダムフォレストのモデル. 複数の決定木を持つランダムフォレストは, 各決定木で条件分岐を繰り返し, それぞれの結果から一つの出力を求める構造になっている.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - f_i)^2} \tag{1.4}$$

最後に, 回帰式モデルの適切さを表す AIC（赤池情報量基準）を式（1.5）に示す. この AIC は, 日本の数理統計学者である赤池弘次が発表した. 最大尤度の対数がデータへの適合度合いを, モデルの自由パラメータの個数がモデルの複雑さを表している. AIC は, 小さいほどその回帰式が良いモデルであることを示す.

$$\mathrm{AIC} = -2\log(\text{最大尤度}) + 2 \times \text{モデルの自由パラメータの数} \tag{1.5}$$

ランダムフォレスト

ランダムフォレストは, 図 1.2 のように決定木を複数持つ構造をしている. 決定木学習はある入力値に対して条件分岐を繰り返し結果を得る手法のことである. この決定木を異なる重みや異なるサンプルで複数生成し, それぞれの結果を考慮して最終的な出力を得るという流れになっている.

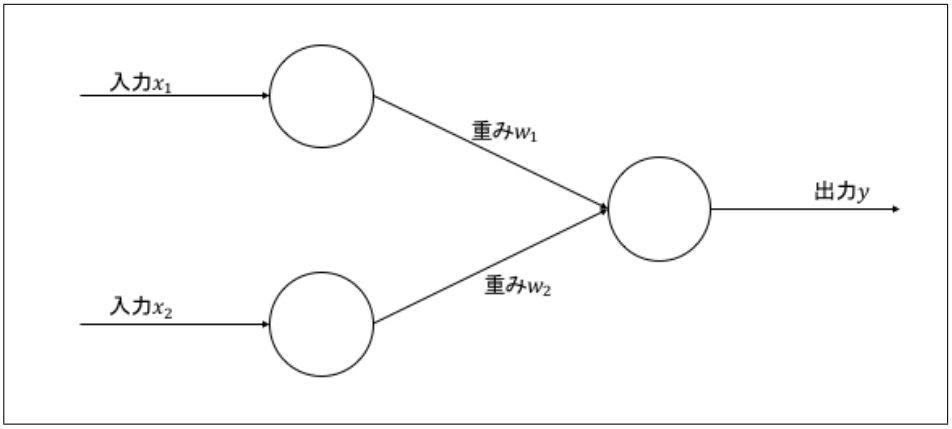

図 1.3: 単純なモデル. 情報の入り口となるニューロンには閾値はなく, そのまま入力される. その後, ニューロン間の結合の強さを示す重み w と乗算される. つぎのニューロンでは式（1.7）に従い, 0 もしくは 1 が出力されるという仕組みになっている.

1.2.2 教師なし学習

教師なし学習は, 教師あり学習と違い学習データのみで正解が与えられていない. 教師なし学習では, 入力データの規則性を学習することを目的とする. 本論文では, 教師なし学習は用いないため詳しい説明は省略する.

1.2.3 ニューラルネットワーク

ニューラルネットワークは, 人間の脳の構造をしていることが大きな特徴としてあげられる. 人間の脳は, ニューロンと言われる神経細胞の集合体でできており, このニューロン同士が電気信号で情報伝達を行っている. 単純なモデルを図 1.3 に示す. 入力を x_1, x_2 とし, ニューロン同士の結合度合いを示す重みを w_1, w_2 とする. この 2 つのニューロンから伝えられる電気信号の総量は式（1.6）で表される. 2 つのニューロンからの電気信号を受け取ったニューロンは, 式（1.7）に従い出力値を決定する. なお, θ は閾値である. このモデルを応用することで複雑なニューラルネットワークの構成も可能となる. [3]

$$w_1x_1 + w_2x_2 \tag{1.6}$$

$$y = \begin{cases} 1 & (w_1x_1 + w_2x_2 \geq \theta) \\ 0 & (w_1x_1 + w_2x_2 < \theta) \end{cases} \tag{1.7}$$

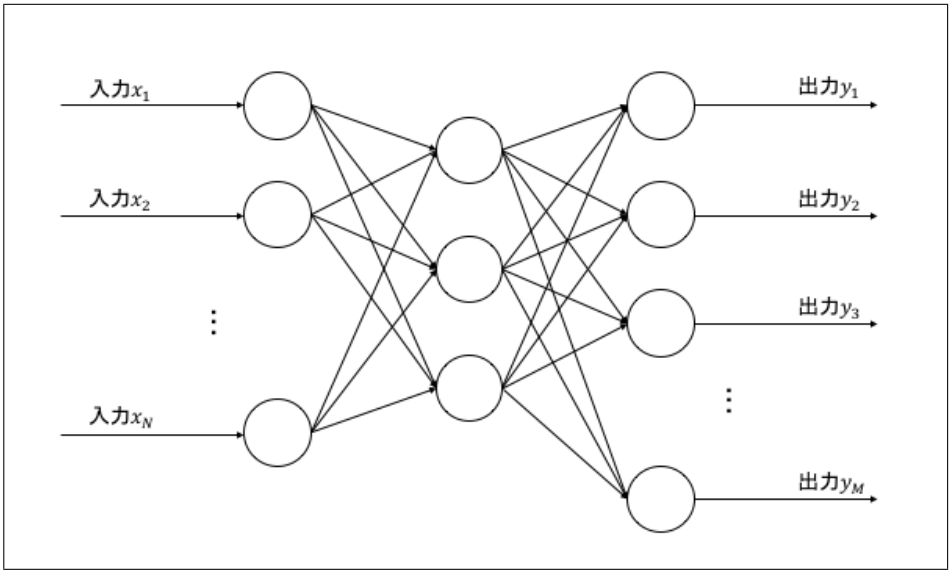

図 1.4: 階層型ニューラルネットワーク. 図 1.3 を組み合わせたモデルである. 左から入力層, 中間層, 出力層といい, 中間層を複雑により深く設計することで出力の精度が向上する.

図 1.3 を複数組み合わせた階層型ニューラルネットワークを図 1.4 に示す. 真ん中の中間層をより充実させることでより正確な出力値を出力するモデルを作ることができる. しかし, 傾向を捉えすぎてしまう過学習も考慮する必要がある.

LSTM ネットワーク

LSTM（Long Short Term Memory）ネットワークは, 時間依存関係を学習することができる RNN（Recurrent Neural Network）の一種である. 従来のニューラルネットワークでは, 時系列データを正しく学習することができなかった. しかしそれを実現するために RNN（図 1.5）が誕生した. 入力値に対し, その前のデータ（h_{i-1}）も考慮しながら出力値を吟味するという構造になっている. しかし, この RNN では短期間の時系列データを扱うことはできるが, 長期的な依存を学習することが難しいという欠点がある. それを補ったものが LSTM（図 1.6）である. 前のデータを考慮するノード h に加え, 記憶を考慮するノード c を追加することで長期的な時系列データを扱うことを可能とした.

Affine 層

Affine 層は, すべての入力値からすべての出力ニューロンへの結合を持つ全結合層であり, 式 (1.8) で表される. なお, y を出力, x を入力, w を重み, b をバイアス項

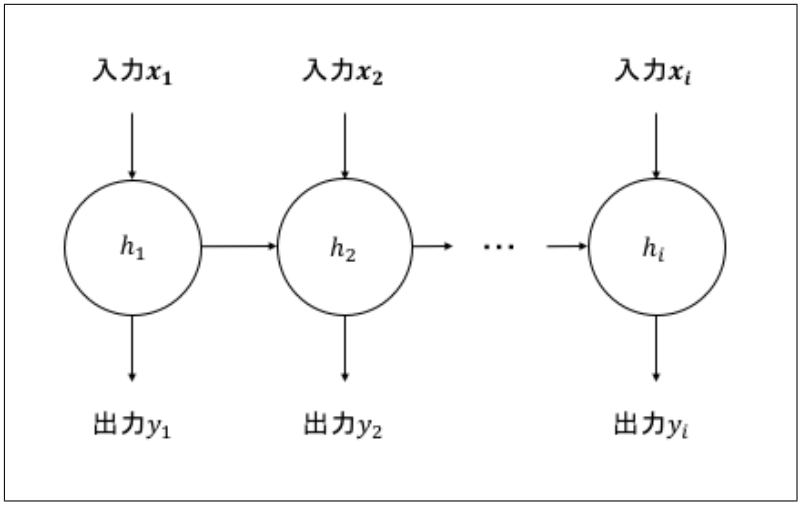

図 1.5: RNN（Recurrent Neural Network）のネットワーク構造. 時系列の入力データを扱うためのモデルであり, ひとつ前のデータ h を引き継ぐことで時間の流れを捉えることができるようになっている. しかし, 長期的な依存関係を扱うことが難しいという問題点がある.

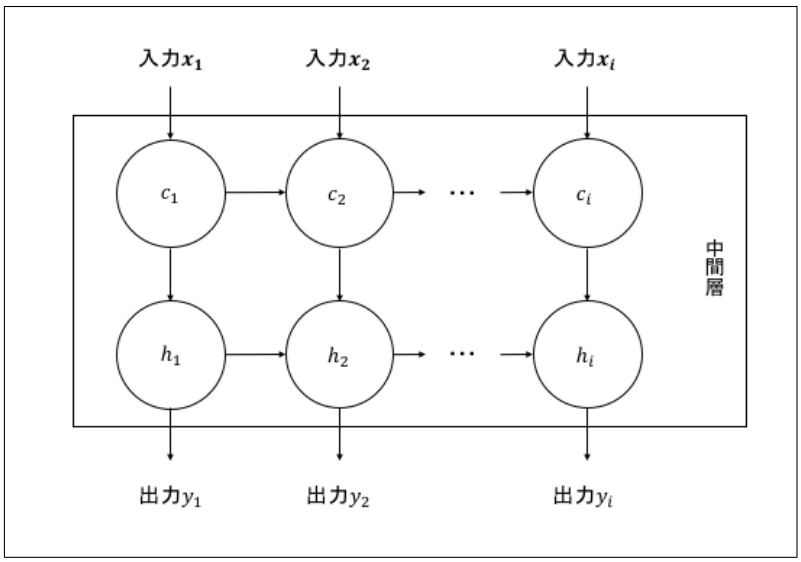

図 1.6: LSTM（Long Short Term Memory）のネットワーク構造. RNN では扱うことが難しい長期的な事案依存関係を含むデータの学習を可能としたモデルである. ひとつ前のデータ h の情報だけでなく, 随時蓄積されてきた記憶を受け持つノード c を付加したものである.

とする.

$$y = wx + b \tag{1.8}$$

Sigmoid 層

Sigmoid 層は, 0.0　1.0 の値を出力する出力層である.

SquaredError

データセットとの変数の二乗誤差を最小化する出力層である. 連続値を出力するネットワークを構成する際に使用する.

1.3　リスティング広告

世の中にはチラシや看板, テレビ CM といった様々なマーケティング方法があるが, ウェブマーケティングにおいて効果があるとされている手法に「リスティング広告（検索連動型広告)」がある. リスティング広告は, Google や Yahoo!といった検索エンジンでユーザーが検索した際に, その検索ワードに連動して表示される広告のことである. キーワード単位で入札し, あるキーワードに対する価格が高いほど, そのキーワードが入力された時の出現率が高いというものである. その分野に興味や関心のあるユーザーに対して広告を出すことができるため, 効率が良い広告方法としても知られている. リスティング広告の料金体制は, クリックされた数×そのキーワードの入札価格が実際の広告料金となる. (式 (1.9)) 人気キーワードは最低入札価格が高くなるという傾向がある. 入札価格やキーワードの指定ができ, 実際の訪問者数などを見ながら調整できるため比較的自由度の高い広告となっている. リスティング広告の一例を図 1.7 に示す.

$$\text{広告費用} = \text{入札価格} \times \text{クリック数} \tag{1.9}$$

図 1.7: IoM のリスティング広告. 「部品加工」「見積り」というような IoM に関係のあるキーワードを検索ウィンドウで入力し検索をすると, この広告が表示される可能性がある.

1.4 使用言語・アプリケーション

1.4.1 R 言語

R 言語は, ニュージーランドの Ross Ihaka と Robert Clifford Gentleman によりつくられた統計解析向けのプログラミング言語である. R 言語は統計処理のライブラリがとても豊富で, 総合開発環境の RStudio を利用すると便利だ.

1.4.2 Neural Network Console

Neural Network Console（以下, NNC）は 2017 年 11 月に SONY がリリースをしたディープラーニングツールである. ここでディープラーニングとは機械学習をより発展させたもので, ニューラルネットワークを使用している.

1.4.3 Mathematica

Mathematica とは, 様々な技術計算を可能とする数式処理システムである. グラフィック能力にも長けており, また, 会話形式で答えを得ることができるのも魅力のひとつである. バージョン 11 では機械学習機能も拡張され, 常に発展を続けている.

1.5 研究背景

2017 年 4 月より, ご縁があって北九州の岡野バルブ製造株式会社（以下、岡野バルブ）に長期インターン生として勤めた. 岡野バルブはバルブの製造やメンテナンスを主事業とし, 現在は新規事業の展開にも挑戦している. その新規事業の 1 つとして, 2015 年から IoM（Internet of Manufacturing）事業が始動した. 岡野バルブの属する製造業界は, 他業界に比べて IT 化が進んでおらず, 見積りの依頼から納品まで FAX やメールを通して行う. この方法ではとても手間がかかるため, インターネットを通して受発注を行うことができるのではないかというアイデアから IoM が生まれた. インターネットを介すことで今までは出会うことのなかったような日本各地のものづくり企業と発注企業を結びつけることができるというメリットもある. 図 1.8 に IoM のホームページトップを示す. ホームページも 2017 年にリニューアルし, ユーザーが一目でわかり易いようなホームページ作りに力を入れている.

1.6　本研究の目的

2016 年に「Alphe Go」が世界トップレベるの囲碁棋士に勝利したことがきっかけに人工知能が脚光を浴びた．その中の機械学習もまた日に日に進歩を見せ, 今やネット通販の販売促進や迷惑メールチェックといった様々な場面で使われている．また, ビジネスにおいても IT 技術が広く使われており, そのひとつとしてホームページのセッション数（ホームページへのユーザー数の訪問回数）を増やすためにリスティング広告（検索連動型広告）が営業ツールとして用いられている．このリスティング広告は検索エンジンの入力ウィンドウに入力されたキーワードに応じて広告を出すネット広告のひとつである．リスティング広告の利用は非常に簡単で, 広告がクリックされた時のみ費用が発生するため使い勝手が良い一方で, リスティング広告を使用することできちんと利益として還元されているかの評価を行うことは難しいという問題点がある．

岡野バルブ製造株式会社の新規事業である IoM 事業部では, ものづくりをインターネットで実現するサービスを展開している．インターネットを利用するビジネスとなるため日本全国の様々な企業とマッチングする可能性は増えるが, 検索エンジンでユーザーが入力したキーワードで IoM のホームページが上位に出てこないと IoM の存在を知ってもらう機会を失うことになる．そこで IoM では, SEO 対策と並行して図 1.7 のようなリスティング広告を出している．ユーザーが入力す

図 1.8: IoM のホームページトップ．IoM では, ものづくりの受発注をネットで行うためのサービスを提供している．2017 年にリニューアルし, 気軽に問い合わせが可能なチャットも備え付けてある．この画面がトップページとなっており, ここに顧客を誘導ための対策としてリステイング広告を開始した．

るキーワード（ニーズ）に合わせて広告が出現するため, IoM のようなサービスに興味のあるユーザーをホームページに誘導することができる. つまり, リスティング広告はこのサービスに興味のあるユーザーのセッション数の増加を狙うことができる広告である. このようにリスティング広告は, ネットビジネスを展開する上で使い勝手が良いツールのひとつである. しかし, リスティング広告がどれくらいセッション数に影響を与えているか可視化することは難しい. そこで, 機械学習の様々なモデルを使いながらリスティング広告によってホームページのセッション数がいかに増加したかを数値化し, さらにはかかった費用と売上推移からリスティング広告の有効性を評価する.

第2章　研究方法

ここでは, 1) で各手法のモデルの精度を評価するために広告不使用の場合のデータを使ってモデルを構築し, 予測する. その結果と実際のデータの相対誤差をとることでモデルの評価を行う. 2) ではリスティング広告の評価をするためにリスティング広告をしなかった場合の予測を行い, 実データと比較してセッション数がどのくらい増加したかを調査する.

1). 精度評価
まずはじめに各モデルの精度評価を行う. この精度評価には, リスティング広告を使用していなかった期間のデータ 23 週分を用いる. 図 2.1 に示すように, 23 週のデータを学習データ（2016/10/3 - 2016/02/27）と評価データ（2017/02/28 - 2017/03/12）の 2 つに分ける. 学習データを用いてそれぞれの手法ごとの学習ありゴリズムに基づきモデルを作り, 評価データの期間のセッション数の予測を行う. 予測したデータと評価データのセッション数を比較することにより, そのモデルの精度評価を行う.

評価の方法としては, 式（2.1）に示す相対誤差用いる. この値が小さいほど, 実測値とのズレが少なく精度の良いモデルであると判断することができる.

$$\text{相対誤差} = \left|\frac{\text{実測値} - \text{予測値}}{\text{実測値}}\right| \tag{2.1}$$

2). 予測
次に , それぞれの手法で決定したモデルを使い, 精度評価で使用した期間（2016/10/03-2017/03/02）を学習データとし, その先の 4 週分のセッション数を予測する. リスティング広告を行った実際のセッション数データと, ここで結果として出てくるデータ（リスィテング広告を行っていなかった場合のセッション数の予測値）を比較することでリスティング広告によりどれくらいセッション数が増加したかを評価する.

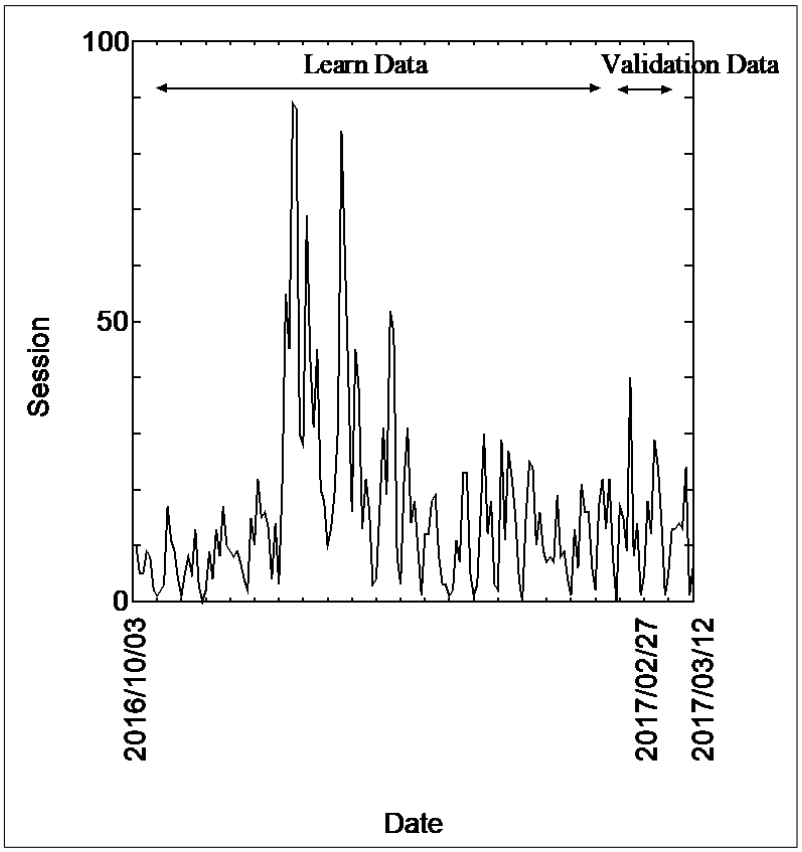

図 2.1: 精度評価に用いるデータ. 学習データ (Learn Data) として, 2016/10/03-2017/02/27 を使用し, 各手法に従って学習させる. そして, 各手法の精度を評価するために, 図中の評価データ（Validation Data）の期間 2017/02/27-2017/03/12 を予測して実際のデータとのズレを比較する.

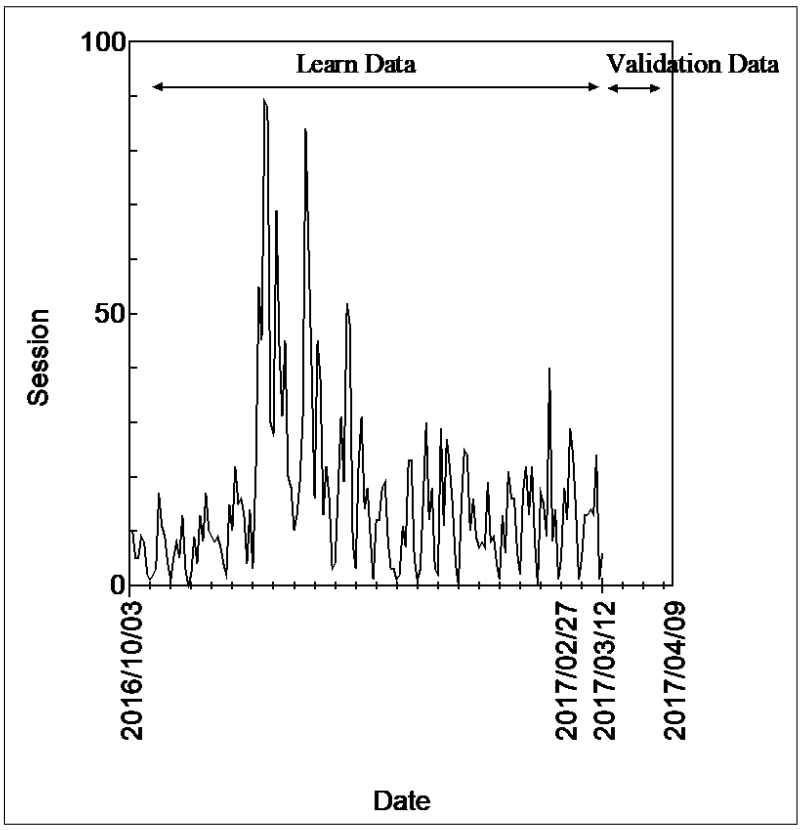

図 2.2: 予測に用いる学習データと評価データ．2016/10/03-2017/03/12 のデータを学習データとして, 広告をしなかった場合の実測データがない期間（2017/0313-2017/04/09）のセッション数を予測する．これとリスィテング広告を使用したセッション数の実測値を比較することでリスィテング広告がセッション数に与える影響を調査する

2.1 R 言語による重回帰を用いたモデル

データセットとしては, 日付や曜日情報を含んだ図 2.3 のようなものを用意する. データの 1 列目に日付, 2 – 8 列目に曜日データを格納している. また 9 列目には求める値の正解値であるセッション数がインプットされている.

また, 用いたコードを図 2.4 に示す. ファイルをインプットし, 日付データを数値化し以下を実行させる. ここで, lm 関数は回帰を行う関数であり, step(lm) をすると説明変数を増減させながら最も AIC の良いモデルを探し出す. これにより回帰式を決定し, これにデータを代入することで予測を行う.

date	mon	tue	wed	thu	fri	sat	sun	session
2016/10/3	1	0	0	0	0	0	0	10
2016/10/4	0	1	0	0	0	0	0	5
2016/10/5	0	0	1	0	0	0	0	5
2016/10/6	0	0	0	1	0	0	0	9
2016/10/7	0	0	0	0	1	0	0	8

図 2.3: R の使用データ. 1 列目に日付, 2 – 8 列に曜日データ, そして学習データにのみ 9 列目にセッション数を格納してある. 1 – 8 列目を学習データの入力, 9 列目を答えとしてモデルを作らせる.

```
data= read.csv("学習用R_.csv", header = TRUE) データ読み込み#
data$date <- as.Date(data$date) 日付型へ#
data$date <- as.numeric(data$date) 実数型へ#

lm.data <- lm(session ~ date + sun + mon + tue + wed + thu + fri + sat,
data = data)
step(lm.data) #調査AIC
```

図 2.4: AIC による回帰式の決定のためのコード. データを読み込み, 必要箇所の数値変換, そして lm 関数により回帰式を作成する. それを step 関数の引数として入れることで, もっとも AIC がよくなる回帰式を探してくれるコードである.

2.2 Neural Network Console を用いた方法

まず, input として与える学習データについて説明を行う. 学習データの形式は, 図 2.5 のようなセッション数を縦に 1 週間分 (7 つ) 並べたものを 1 つの excel ファイルに準備する. これを学習させる週の分作成する. それとは別に, 学習データの excel ファイルがある場所を示すファイルを作成する. (図 2.6) 一列目に各ファイルをフルパスで指定し, 二列目は今回は用いないので 0 で埋める. なお, 一列目はヘッダーとなっており任意入力で良い. データのファイル分割やファイルパスの取得等には excel マクロを使用した.

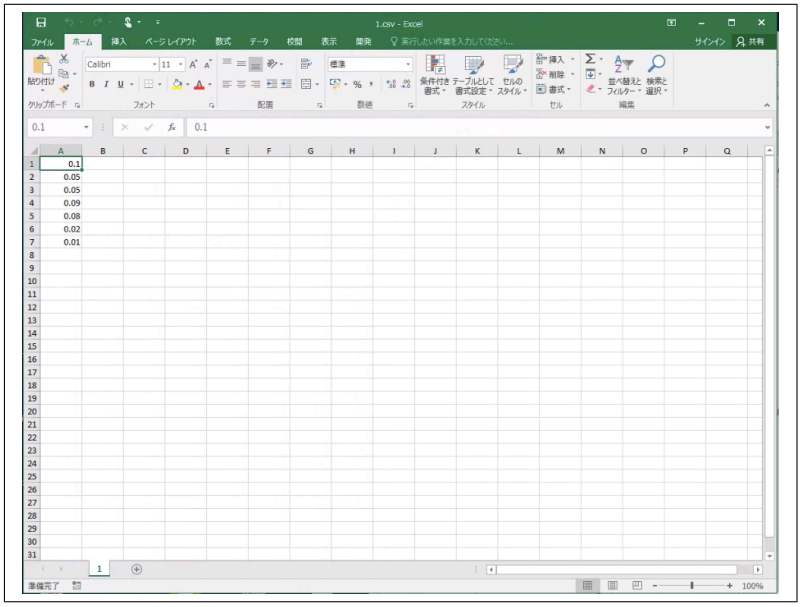

図 2.5: NNC の使用データ (分割データファイル) の内容. 列方向にセッション数を一週間分ずつ格納する. なお, NNC では値の範囲として −1.0-1.0 である必要があるため, 100 で除算している.

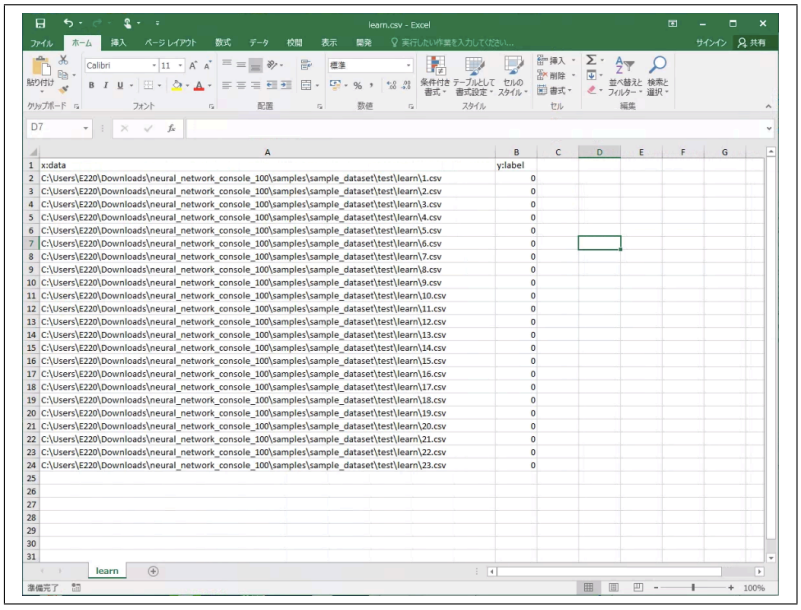

図 2.6: NNC の使用データ (学習ファイル) の内容. 1 列目に分割したデータファイルの場所を示すファイルパスを入力する. 2 列目は今回使用しないので, 0 を入力した.

2.2.1 リカレントニューラルネットワーク

まず, Neural Network Console で作成したリカレントネットワークを図 2.7 に示す. リカレントニューラルネットワーク（以後, RNN）の特徴としては, Delay 層で

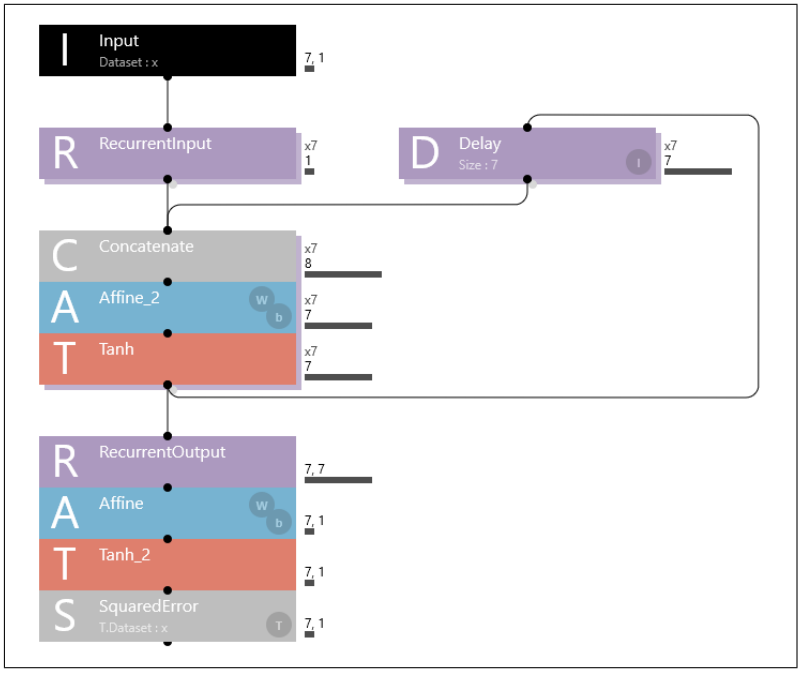

図 2.7: リカレントニューラルネットワークのネットワーク構成．時間依存性を学習するために, Delay レイヤで過去のデータを保持しているところがこのネットワークのポイントである．

1 つ前の時間のデータを受け継ぐことができるという点である．その他には, 全結合を行う Affine レイヤ, 畳み込みを行う Tanh レイヤを使った．しかし, この RNN では, 長期的な時間依存を学習することができないという欠点がある．そのため, 次の手法として LSTM（Long-Short Term Memory）ネットワークを作成した．

2.2.2 LSTM ネットワーク

Neural Network Console で作成した LSTM ネットワークを図 2.8 に示す．入力層として Input レイヤ, 中間層として時系列データを扱うことのできる LSTM レイヤ, 全結合を行う Affine レイヤ, 出力層として Sigmoid レイヤと SquaredError レイヤを用いた．Neural Network Console では, LSTM レイヤとしてモデル化されているが, 実際には RNN より複雑なネットワーク構成をしている．このネットワークに準備した学習用データをインプットして予測を行う．

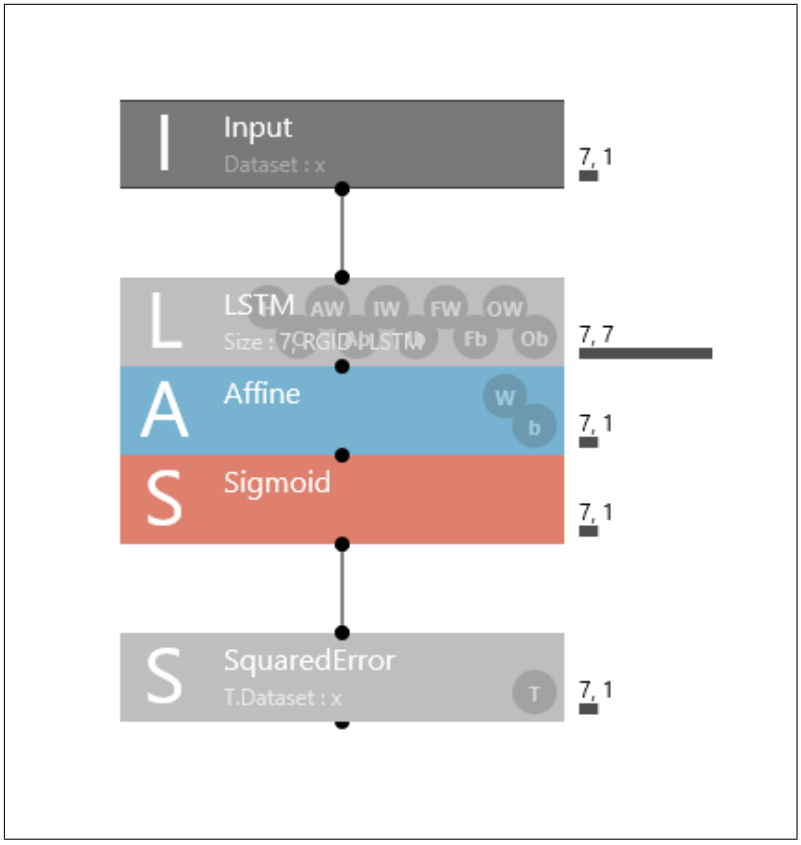

図 2.8: LSTM ネットワークのネットワーク構成. 長期時間依存関係を学習できるように LSTM レイヤを用いた.

2.3 Mathematica

Mathematica での予測に用いるデータは, 図 2.3 と同内容の物を用いる. Mathematica では, 手法を簡単に指定して機械学習できるため, 今回はニューラルネットワークとランダムフォレストの 2 つの手法で行った.

```
p1 = Predict[ learn -> result, Method -> "NeuralNetwork"]
p2 = Predict[ learn -> redult, Method -> "RandomForest"]
```

図 2.9: Mathematica における機械学習のコード. 入力 − > 出力, Method で使用する手法を指定する.

第3章　結果

1) に機械学習の各モデル評価を行った結果を, 2) にセッション数の予測を行いリスティング広告の効果を調査した結果を示す.

3.1　R 言語による重回帰を用いたモデルの結果

図 2.4 のコードを用いて AIC が最もよくなる回帰式を導出した結果を図 3.1 に示す. 図 3.1 の通り, 回帰式は式（3.1）に決定した. 日付データに左右されない曜日のみ考慮する式となった. ただし, mon, tue, wed, thu, fri はそれぞれ月火水木金を示す. また, $b_0 = 7.500$, $b_1 = 9.548$, $b_2 = 10.93$, $b_3 = 12.31$, $b_4 = 16.17$, $b_5 = 11.93$ である.

$$\text{session} = b_0 + b_1\text{mon} + b_2\text{tue} + b_3\text{wed} + b_4\text{thu} + b_5\text{fri} \tag{3.1}$$

Df（自由度）はある変数において自由な値をとることのできるデータ数, Sum of sq（変動）は観測値から平均値を引いた偏差の 2 乗和, RSS（残差平方和）は誤差の 2 乗和を示す.

```
Step:  AIC=817.13
session ~ mon + tue + wed + thu + fri

       Df Sum of Sq   RSS    AIC
<none>              35156 817.13
- mon   1    1276.2 36432 820.38
- tue   1    1672.1 36828 821.97
- fri   1    1992.1 37148 823.24
- wed   1    2121.3 37277 823.75
- thu   1    3659.1 38815 829.69
```

図 3.1: step 関数を用いて決定した回帰式. step 関数を用いると, 最初に指定した式から説明変数を削ったり増やしたりしながら AIC の値がもっとも良いものを探索する.

1). 精度調査の結果

決定した重回帰式である式 3.1 を用いて予測したものと実際のデータをグラフにまとめたものを図 3.2 に示す. 式（3.1）と図 3.2 を見ると, 上記で述べたように時間の流れ（日付データ）を考慮せず, 曜日によって決まった値をとるような結果になったが曜日による変動は捉えることができている. また, 各日付ごとの相対誤差の平均をとると, 54.06%となった.

2). 予測結果

式（3.1）を用いて, 2017/03/13 – 2017/04/09 の 1 か月分のセッション数を予測した. その結果と実データをグラフにまとめたものを図 3.3 に示す. 黒実線のデータは, リスティング広告を行った時のセッション数で, 赤実線のデータは, リスティング広告を行ってなかった場合のセッション数の予測値を示す. リスティング広告を導入することで, 1 日あたり約 19.77 セッション増加して

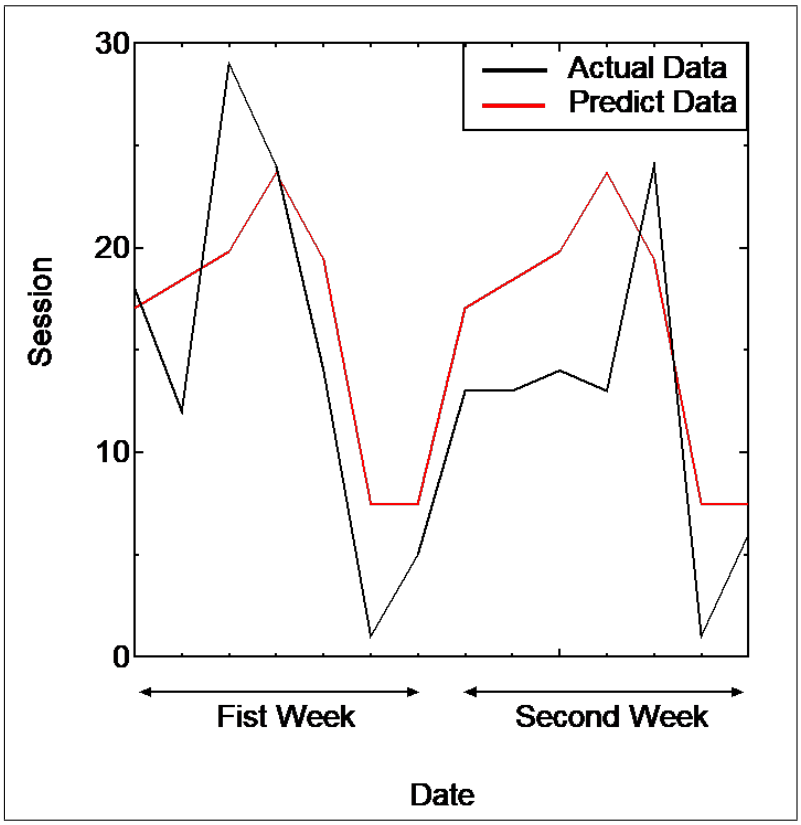

図 3.2: 重回帰式による実測データと予測データの比較グラフ. 式（3.1）を用いて, 2 週間分のセッション数予測を行い実際のデータと比較した. 図を見てわかるように曜日によってとる値が決まっており, 単調な結果となったが曜日による変動は再現していることがわかる.

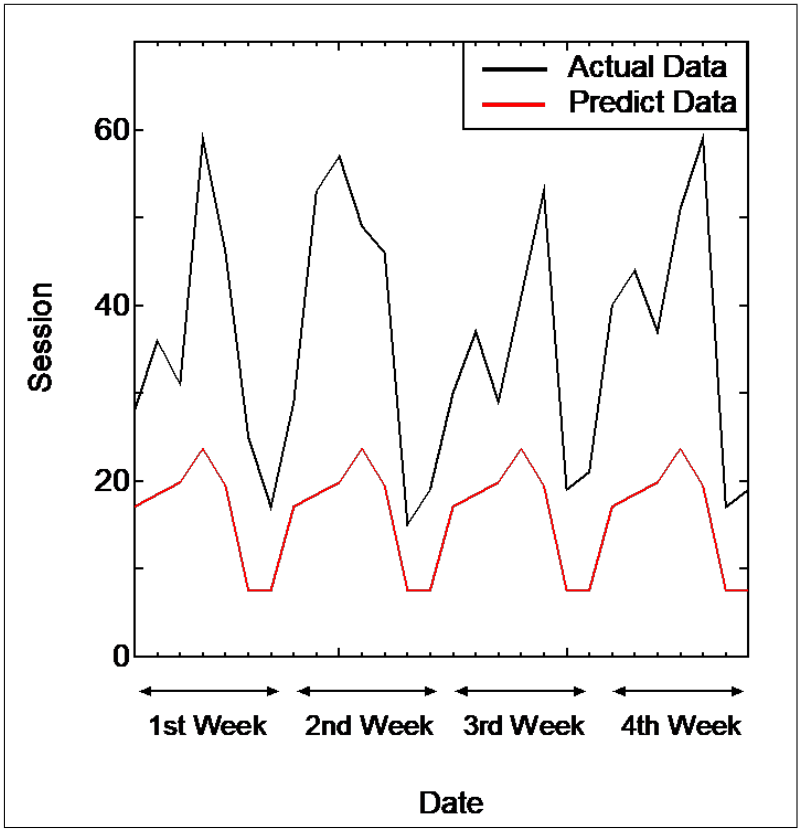

図 3.3: リスティング広告評価のための重回帰式による実測データと予測データの比較グラフ. 黒は実際のデータで赤は予測データとなる. 広告を行った場合は, 広告を行わなかった場合と比較して, セッション数を増加させる働きがあることが明確に見える.

いるという結果になった.

3.2 Neural Network Console

3.2.1 リカレントニューラルネットワーク

1). 精度調査の結果
図 2.7 のネットワークを作成し, 精度評価を行うための学習データセットをインプット, そして学習を行った. すると, 図 3.4 のような学習曲線が描かれる. 図 3.4 のように, Epoch（世代）が増加するにつれて学習エラーが減少するような曲線になっているため, 適切な学習が行われていることがわかる.

次に, このモデルの精度評価のための 2 週間分の予測を行った結果と実測データをグラフにまとめたものを図 3.5 に示す. 各日付ごとの相対誤差の平均をとると, 53.15%となった.

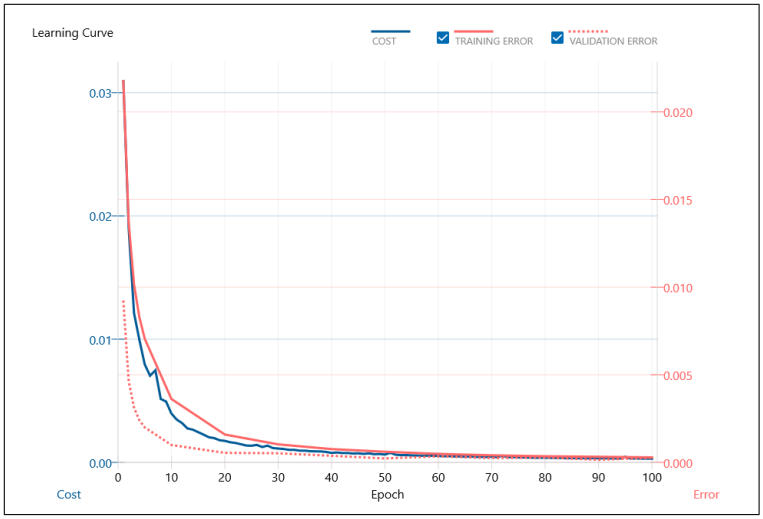

図 3.4: 精度評価のための NNC（RNN）の学習曲線. Epoch（世代）が増加するにつれて学習エラーは減少していることがわかるため, 適切に学習が行われている.

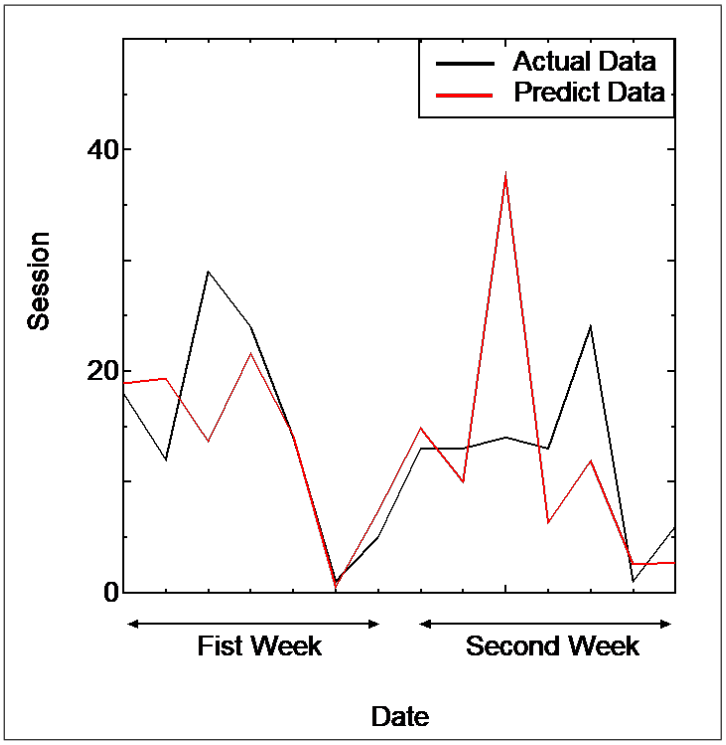

図 3.5: 精度評価のための NNC の実測データと予測データの比較グラフ. 実測値と大きく外れるデータもあることが見て取れる. しかし, 日付データを考慮して予測できるというこのモデルの強みもある.

2). 予測結果

図 2.7 のネットワークで 2017/03/13 – 2017/04/09 の 1 か月分のセッション数を予測した. その結果であるリスティング広告を利用していない場合の予測値とリスティング広告導入後実データをグラフにまとめたものを図 3.6 に示す. この手法で予測した結果により, リスティング広告による効果は, 1 日あたり 20.21 セッション増加したことがわかった.

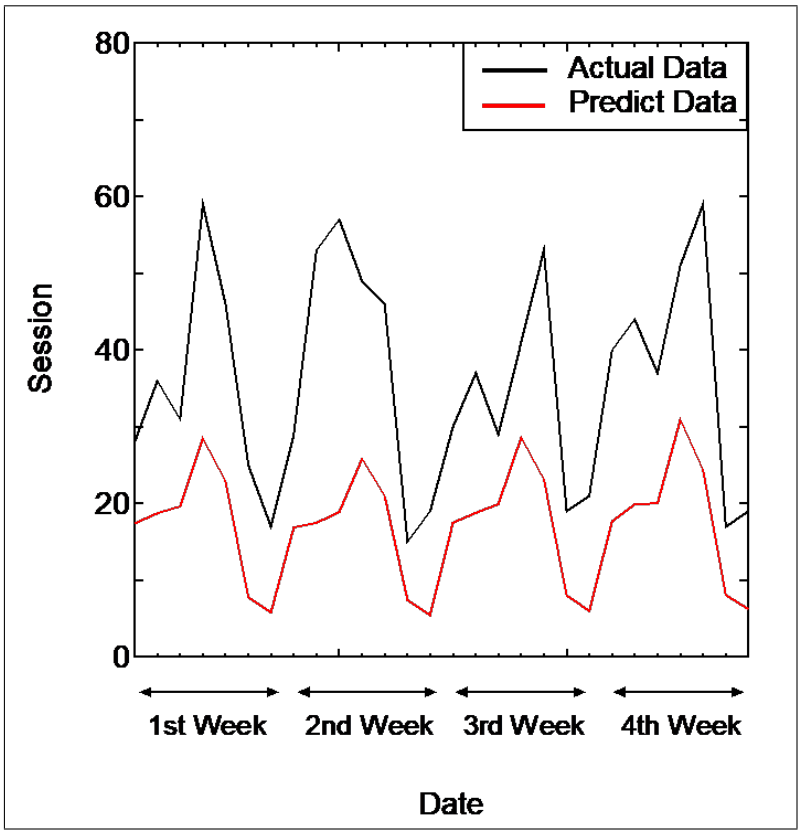

図 3.6: リスティング広告評価のための NNC の実測データと予測データの比較グラフ. 黒実線は, リスティング広告導入後の実際のセッション数を示し, 赤実線は, リスティング広告を行わなかった場合の予測セッション数を示す. 図からリスティング広告の効果は明確に表れていることがわかる.

3.2.2 LSTM ニューラルネットワーク

1). 精度調査の結果

図 2.8 のネットワークを作成し, 学習を行った. すると, 図 3.7 のような学習曲線が描かれる. RNN と同様に図 3.7 のように, Epoch（世代）が増加するにつれて学習エラーが減少するような曲線になっているため, 適切な学習が行われていることがわかる.

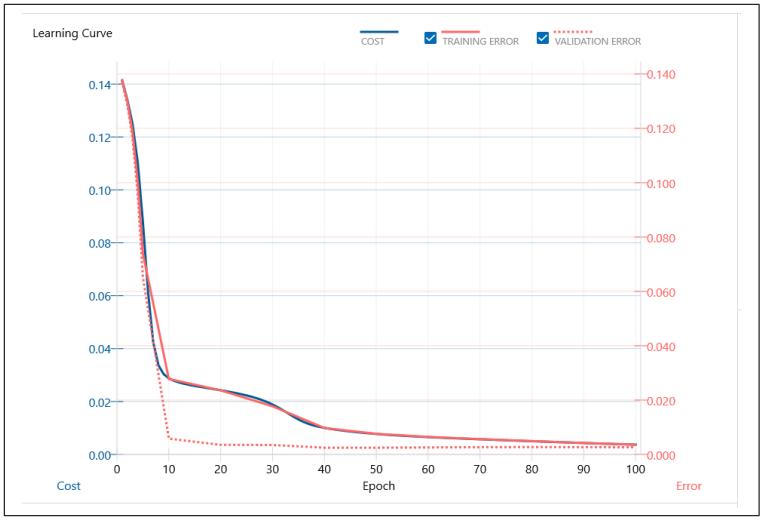

図 3.7: 精度評価のための NNC（LSTM）の学習曲線. Epoch（世代）が増加するにつれて学習エラーは減少していることがわかるため, 適切に学習が行われている.

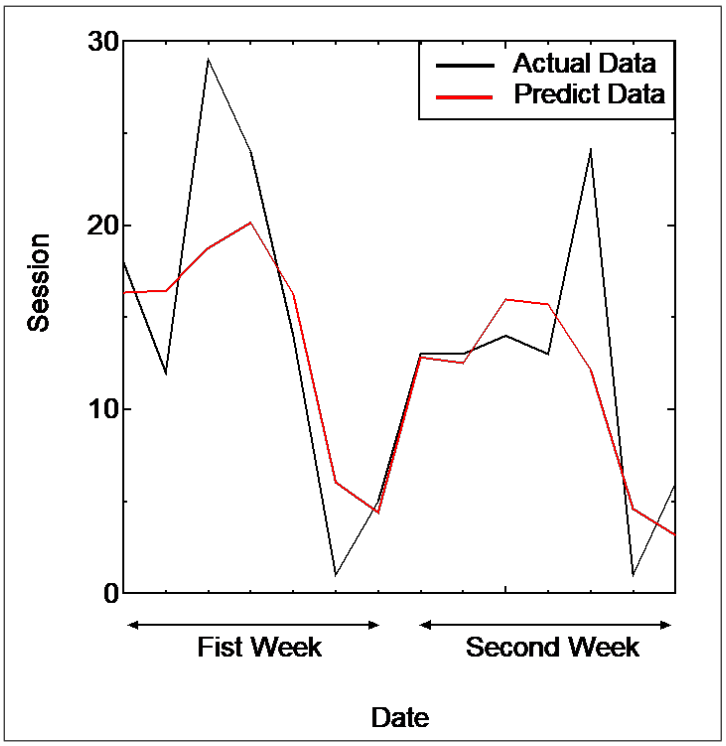

図 3.8: 精度評価のための NNC の実測データと予測データの比較グラフ. 実測値と大きく外れるデータもあることが見て取れる. しかし, 日付データを考慮して予測できるというこのモデルの強みもある.

次に, このモデルの精度評価のための 2 週間分の予測を行った結果と実測データをグラフにまとめたものを図 3.8 に示す. 各日付ごとの相対誤差の平均をとると, 80.37%となった. 今回. 精度としては RNN モデルに比べ劣っているが, LSTM ネットワークを利用したため, 曜日だけでなく日付の変動によるセッション数の変動も考慮したモデルとなっている.

2). 予測結果

図 2.8 のネットワークで 2017/03/13 – 2017/04/09 の 1 か月分のセッション数を予測した. その結果であるリスティング広告を利用していない場合の予測値とリスティング広告導入後実データをグラフにまとめたものを図 3.9 に示す. この手法で予測した結果により, リスティング広告による効果は, 1 日あたり 19.58 セッション増加したことがわかった. また, 日にちが経つにつれて 1 週間ごとにセッション数が増加傾向にあることもわかる.

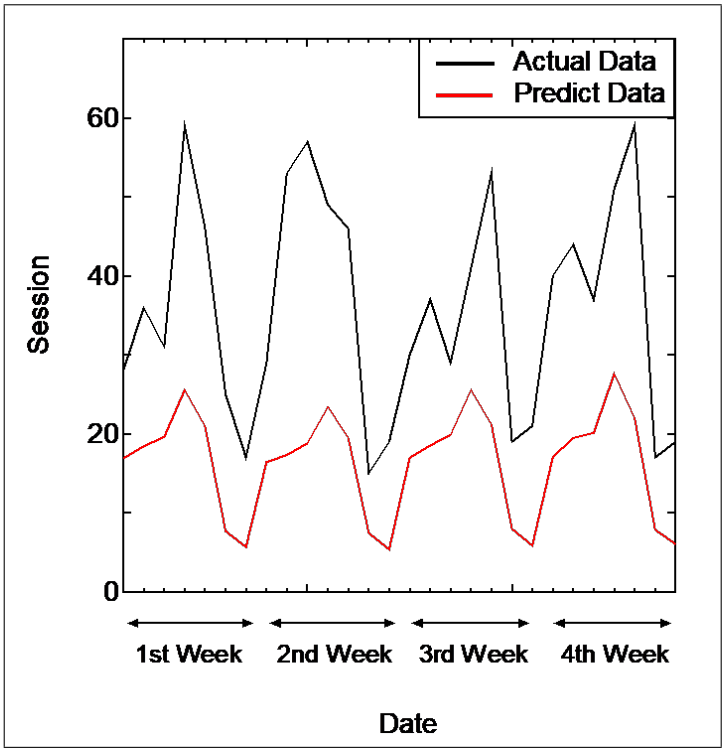

図 3.9: リスティング広告評価のための NNC の実測データと予測データの比較グラフ. 黒実線は, リスティング広告導入後の実際のセッション数を示し, 赤実線は, リスティング広告を行わなかった場合の予測セッション数を示す. 図からリスティング広告の効果は明確に表れていることがわかる.

3.3 Mathematica

3.3.1 ランダムフォレスト

1). 精度調査の結果
Mathematica の Method でランダムフォレストを呼び出し, 精度評価を行なった結果を図 3.10 に示す. 結果は, 重回帰を行なった時と同様に曜日によってとる値が同値になった. なお, 相対誤差は 148.34%となり, その他のすべての手法と見比べてもいい結果にはならなかった.

2). 予測結果
図 3.11 に 1 か月分のセッション数を予測したものと実際のデータをグラフにまとめたものを示す. この手法で予測を行うと, リスティング広告導入後は 1 日あたり約 18.14 セッション増加していることがわかった.

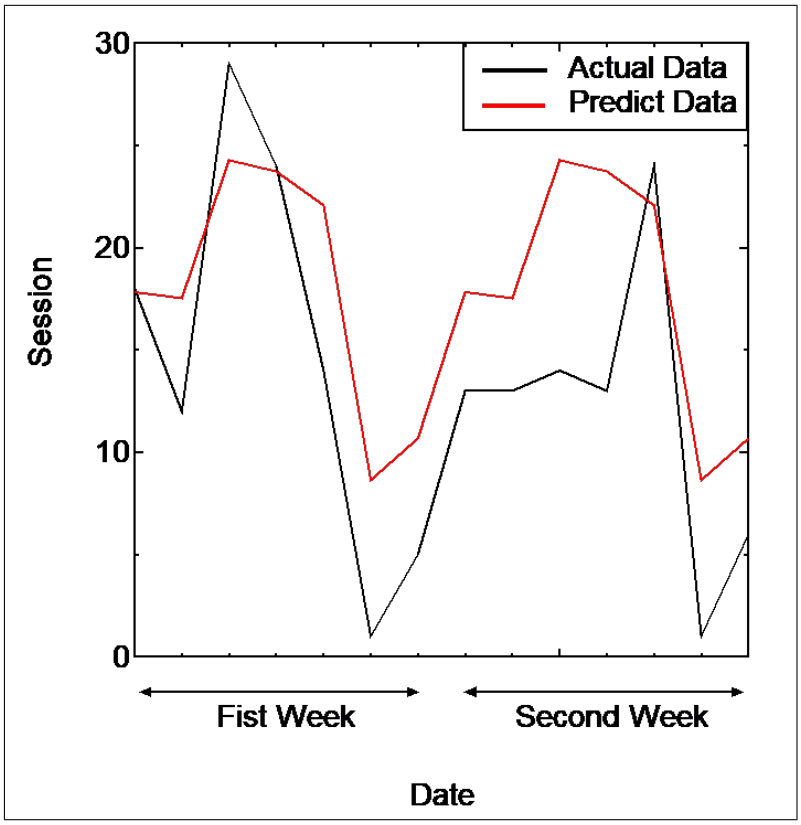

図 3.10: 精度評価のための Mathematica ランダムフォレストの実測データと予測データの比較グラフ. 曜日によってとる値が一定になるという結果になった.

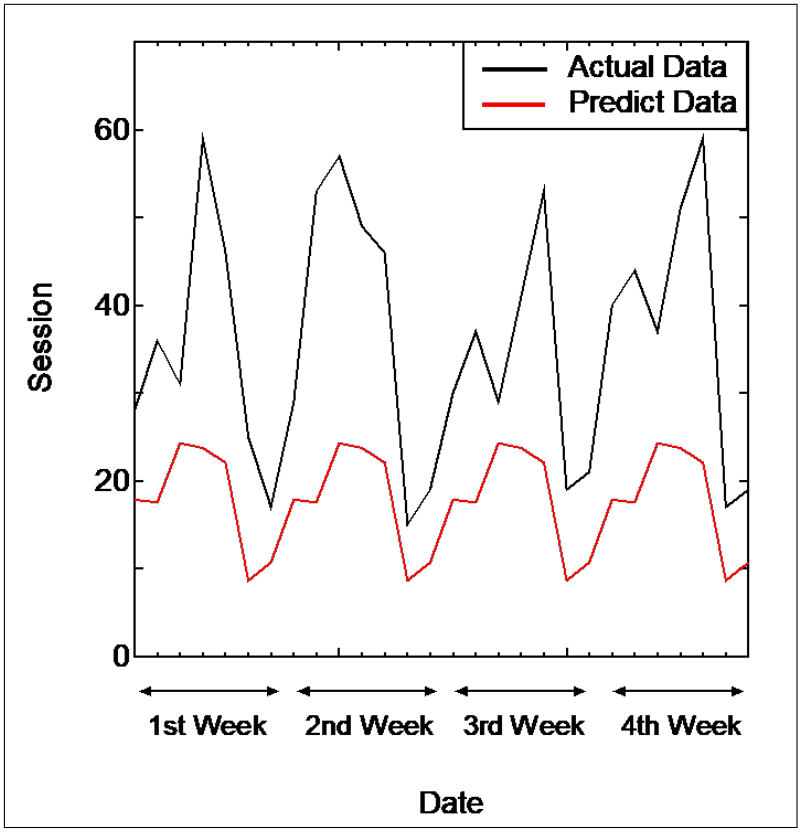

図 3.11: リスティング広告評価のための Mathematica の実測データと予測データの比較グラフ. 黒実線がリスティング広告を使った実際のセッション数データを示し, 赤実線はリスティング広告を使わなかった場合のセッション数予測値を示す. 図より, リスティング広告の効果によりセッション数が増加していることが見て取れる.

3.3.2 ニューラルネットワーク

1). 精度調査の結果
Mathematica の Method でニューラルネットワークを呼び出し, 精度評価を行なった結果を図 3.12 に示す. この手法による相対誤差は 270.30%となった. また, 図 3.12 を見ると, 全体的に平均値を中心としセッション数が曜日によって変動しているが, ばらつきの少ない値をとるようなモデルになっているように見える.

2). 予測結果
図 3.13 に 1 か月分のセッション数を予測した結果と実際のデータをグラフにまとめたものを示す. 1 週ごとにセッション数が増加していることから長期的な時間依存性も考慮したモデルとなっていることがわかる. またこの手法で予測を行うと, リスティング広告導入により 1 日あたり約 5.99 セッション

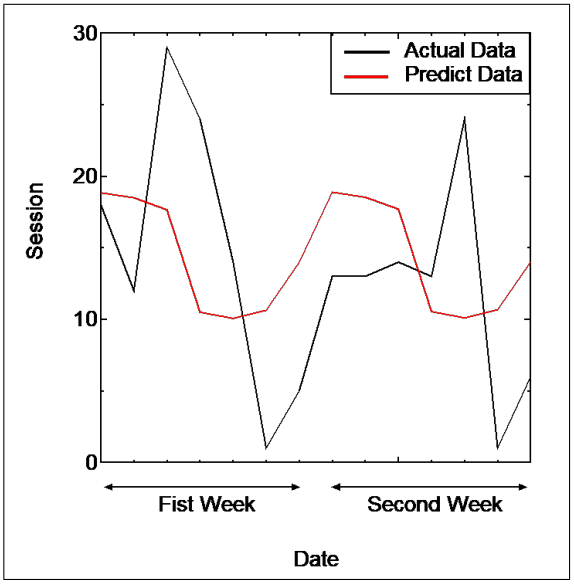

図 3.12: 精度評価のための Mathematica ニューラルネットワークの実測データと予測データの比較グラフ. 全体的に平均値を中心にセッション数が上下するというモデルになっている.

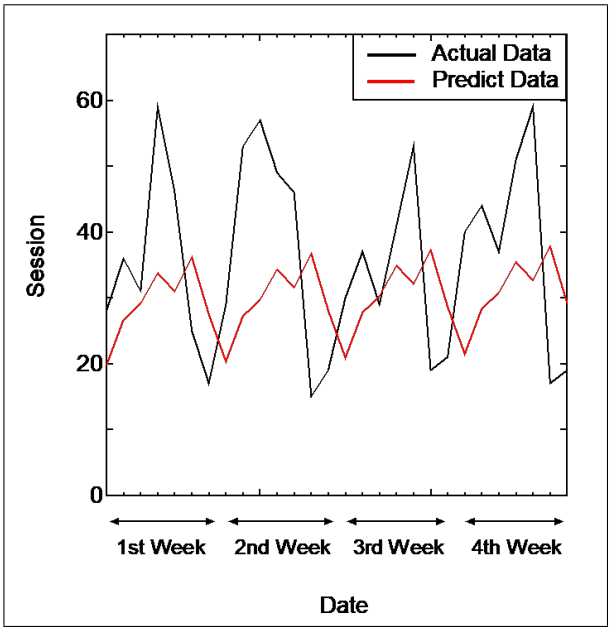

図 3.13: リスティング広告評価のための Mathematica ニューラルネットワークの実測データと予測データの比較グラフ. 黒実線がリスティング広告を使用した際の実際のセッション数で, 赤実線はリスティング広告を使用しなかった場合のセッション数の予測値を示す.

増加しているという結果になった. 広告の導入によりセッション数が増加することはわかるが, 他の手法に比べて増加数が少ない結果となった.

3.4 各モデルの精度評価

ここまで, 各モデルについての精度評価と予測を行ってきたが, ここでは各モデルの相対誤差について論じる. 各モデルの相対誤差をまとめたものを表 3.1 に示す. なお, 表の上段は各モデルの小番号を指す.

表 3.1: 相対誤差まとめ

	2.1	2.2.1	2.2.2	2.3.1	2.3.2
相対誤差	0.541	0.804	0.427	1.48	2.70

表の通り, NeuralNetworkConsole で用いた RNN が最も良い精度となった. RNN モデルは, 時系列データを扱うことができるネットワークなため, その部分においての学習能力に長けていたと考えられる. その点 LSTM で精度が落ちたのは, 今回用いたデータのデータ数が少なく, 長期依存性の少ないデータであったためこのような結果になったと考えられる.

また, Mathematica では, 複雑な設定をせずにアルゴリズムのみの指令で学習を行った. そのため, NNC のように細かく設計をされておらず, あまり良い結果が得られなかった. 今回指定したアルゴリズムの選択もあまり適切ではなかったことも考えられる.

今回扱ったセッション数のような, 長い目で見ると増加するであろう数値に関するデータに適切なモデルは時系列データを扱うことのできる RNN モデルであった. サイトのセッション数は, ユーザーへ周知が進むにつれ増加していくことを予測することは人間には容易であるが, 機械で学習するのは難しいことであるとわかった. この精度評価を通して, このように相対誤差の差が出たのは, データに適したアルゴリズムやモデルを決めてあげないと適切に学習することができないということがわかる.

第4章　リスティング広告の評価

本論文では, 様々なモデルを用いてリスティング広告を行っていなかった場合のセッション数の予測を行ってきた. すべてのモデルによるリスティング広告をしなかった場合の予測値とリスティング広告をした時の実測値をグラフにまとめたものを図 4.1 に示す.

図 4.1 の黒線はリスティング広告を使用した時の実測データを示し, その他はリスティング広告を使用しなかった場合の予測データを示す. 実データと予測データを比較すると, 1 日あたり約 5.99〜19.77 セッション増加する. ホームページのセッション数を増やすという意図でのリスティング広告は有効であると考える. また,

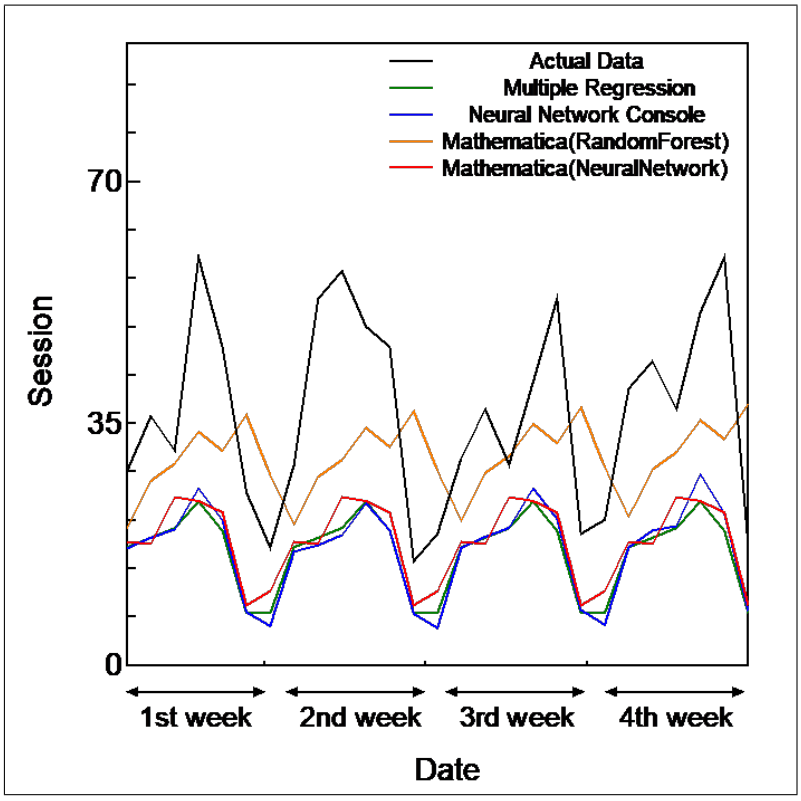

図 4.1: 全モデルの予測結果とリスティング広告使用時の実測値. 黒線はリスティング広告を使用した時の実測データを示し, その他はリスティング広告を使用しなかった場合の予測データである.

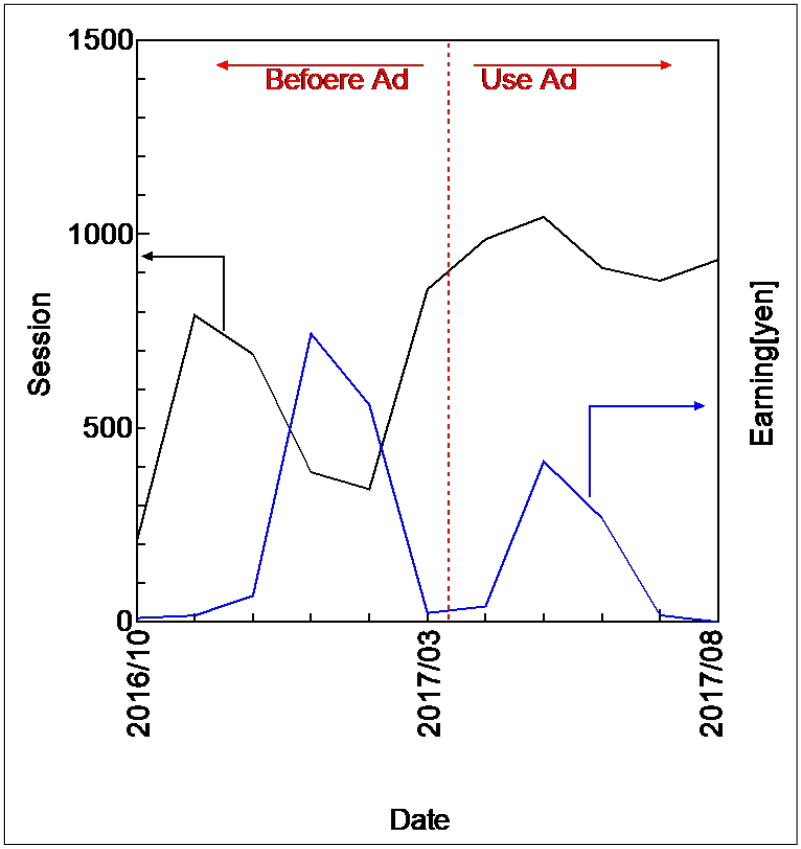

図 4.2: セッション数と売上推移.　黒線がセッション数の推移を示し, 青線が売上の推移を示す. セッション数の増加と売上が必ずしも結びついてはいないということがわかる.

今回のホームページは製造業（特に金属加工）という狭い業界での広告であった. そのため, より幅広く知られているキーワードを含むホームページや web サイトであればよりセッション数の増加が見込める.

次に, 平日と休日に分けてセッション数の増加を調査したところ, 平日では平均 21.3 セッション, 休日では平均 12.2 セッション増加するという結果になった. 今回岡野バルブで用いたリスティング広告のターゲットは, 製造業, もしくはものづくりを必要としている企業（個人）であったため, 営業日であると考えられる平日の方が休日に比べて増加しやすい傾向となった. 顧客となるユーザーが現れるであろう日にちに合わせてリスティング広告を出すと, より効果的に潜在顧客を取り込むことができると考える.

最後に, セッション数と売上の関係を示すグラフを図 4.2 に示す.

2016 年 11 月でセッション数が増えているのがわかる. これは, 2 年に 1 度行われ

る JIMTOF というイベントに IoM が出展したため急増している. その後, 一度減少し, リスティング広告を使用後には安定して毎月 800 セッションを超えていることがわかる. 一方, 売上推移には波があり, セッション数と売上に相関は少ないように思える. また, 広告費用として月額 67,000 円, 運用費（企業への委託費）として月額 20,000 円が発生している. これらのことを考慮すると, リスティング広告によりセッション数は増加しているが利益へと結びついていない可能性が大いにあると考える. そのため, 広告により誘導された潜在顧客をいかに優良顧客へと導いていくかが重要となる. その手法としては, 顧客にとってわかり易いホームページ構成であったり, サービスの質が鍵になってくると考える.

第5章　まとめ

近年, IT 技術の発展によりインターネットを営業ツールとして用いる企業が増えてきており検索エンジンではリスティング広告をよく目にするようになった. リスティング広告は, 通常, セッション数（訪問者数）を増やすことで顧客となりうる分母を拡大するために用いられるネット広告のひとつである. しかし, 重複した期間でリスティング広告を使用した場合としていない場合を比べることができないため実際にどのくらいリスティング広告の影響でセッション数が増加したのかを数値化するのは難しいという問題があった.

そこで本研究では, リスティング広告の評価を行うために機械学習のモデルを用いて将来値を予測してきた. 手法としては, R 言語を用いた重回帰による予測, SONY のディープラーニングツールによる Neural Network Console を用いた RNN モデルと LSTM モデル, Mathematica を用いた RandomForest モデルと NeuralNetwork モデルの計 5 パターンで行った.

リスティング広告を用いなかった場合の予測の結果とリスティング広告を用いた場合の実際のデータを比較すると, 1 日あたり約 5.99〜19.77 セッション増加することがわかった. また, 曜日によってセッション数の増加数にも差が出ており, 休日より平日の方が増加しやすい傾向にあった. しかし, リスティング広告を導入する環境やターゲットの環境により, リスティング広告がセッション数に与える影響には差が出たり, 曜日変動も異なってくると考えられる.

また, 売上との関係についても調査を行ったが, 今回直接的な結びつきは見られなかった. リスティング広告を用いるともちろん費用が発生するため, 広告により誘導された顧客をいかに優良顧客へと取り込んでいくかが重要となる. そのためには, 顧客にとってわかり易いホームページ構成であったり, サービスの質が鍵になっていくと考える.

謝辞

本研究を進めるにあたり, 国立大学法人 九州工業大学 大学院情報工学研究院 電子情報工学研究系 エレクトロニクス分野 教授 小田部 荘司先生から熱心なご指導をいただきました, 深く御礼申し上げます. 小田部先生には, 私のインターンシップ活動を快諾していただき, さらには専門外であるにも関わらず様々なアイデアをいただきました.

国立大学法人 九州工業大学 大学院情報工学研究院 電子情報工学研究系 エレクトロニクス分野 准教授 木内 勝先生には, ゼミを通して超電導の基礎知識から社会に出てからの振る舞い等, 多岐にわたりご教示いただきましたことを深く御礼申し上げます.

岡野バルブ製造株式会社の皆様には, 右も左もわからない私を快く受け入れていただき, さらには利益を生み出す仕組みや実際のプロジェクトの流れまで多くのことを学ぶ機会を与えていただきました. 特に IoM 事業部の皆様には助言や褒め言葉も沢山いただき前向きに取り組むことができました. 深く御礼申し上げます.

最後に, 小田部・木内研究室の先輩, そして同期の方々は個性あふれるメンバーが揃っており, 非常に楽しい研究室生活を送ることができました. ありがとうございました.

参考文献

[1] 日経ビックデータ編 (2017) グーグルに学ぶディープラーニング, p12 – 23, 日経 BP 社

[2] 荒木雅弘 (2014) フリーソフトではじめる機械学習入門, 森北出版

[3] 巣籠悠輔 (2017) 詳解 ディープラーニング, 株式会社マイナビ, p69 – 73