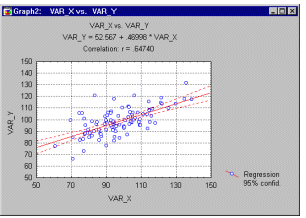
**Лек.3. Регресійний аналіз даних**

**1. Загальні відомості про РАД.  
2. Технологія регресійного аналізу.**  
  
**Загальні відомості про регресію.**  
       Загальне призначення множинної регресії полягає в аналізі зв'язку між декількома незалежними змінними (званими також **регресорам**и або **предикторами**) і залежною змінною. У природних науках  процедури множинної регресії надзвичайно широко використовуються в дослідженнях.

Загалом, множинна регресія дозволяє дослідникові поставити питання і отримати відповідь про те, "**що є кращим предиктором для**.". Наприклад, дослідник в області освіти міг би побажати дізнатися, які чинники є кращими предикторами успішного навчання. Термін **"множинна"** вказує на наявність декількох предикторов або регресорів, які використовуються в моделі.  
               На практиці, наприклад, агент з продажу нерухомості міг би вносити в кожен елемент реєстру розмір будинку в метрах кв., число спалень, середній доход населення в цьому районі відповідно до даних перепису і суб'єктивну оцінку привабливості будинку. Як тільки ця інформація зібрана для різних будинків, було б цікаво подивитися, чи пов'язані і яким чином ці характеристики будинку з ціною, по якій він був проданий. Як тільки ця інформація зібрана для різних будинків, було б цікаво подивитися, чи пов'язані і яким чином ці характеристики будинку з ціною,  якій він був проданий з даними перепису і суб'єктивну оцінку привабливості будинку. Як тільки ця інформація зібрана для різних будинків, було б цікаво подивитися, чи пов'язані і яким чином ці характеристики будинку з ціною, по якій він був проданий.  
  
**Обчислювальні аспекти**.  
Загальне обчислювальне завдання, яке вимагається вирішувати при  аналізі методом множинної регресії, полягає в підгонці прямої лінії  до деякого набору



       У **простому** випадку, коли є одна залежна і одна незалежна змінна, це можна побачити на діаграмі розсіяння.

**У багатовимірному** випадку, коли є більше за одну незалежну змінну, лінія регресії не може бути відображена в двовимірному просторі, проте вона також може бути легко оцінена.  У загальному випадку, процедури множинної регресії оцінюватимуть параметри лінійного рівняння виду :

Y = a + b1\*X1 + b2\*X2 + ... + bp\*Xp

,

де - вектор зн. функ.;

k- номер **індикатора**  (дихання, кров, бращений до лікаря і так далі);

****- вектор чинників;

**m** - розмірність вхідного вектору даних ;

**Н**- потужність навчальної вибірки;  - параметри нейромережі; ****- вагові (синаптические) коефіцієнти нейромережевої моделі.

Регресійні коефіцієнти (чи **B- коефіцієнти**) представляють незалежні вклади кожної незалежної змінної в пророцтво залежної змінної.

Іншими словами, змінна **X1**, приміром, корелює зі змінною Y після обліку  впливу усіх інших незалежних змінних.  
  
**Передбачені значення і залишки.**

Лінія регресії виражає найкраще пророцтво залежної змінної **(Y)** по незалежних змінних (**X).** Проте, природа рідко (якщо взагалі коли-небудь) буває повністю передбачуваною і зазвичай є істотний розкид спостережуваних точок відносно підігнаної прямої (як це було показано раніше на діаграмі розсіяння).  Відхилення окремої точки від лінії регресії (від передбаченого значення) називається **залишком.**

**Приклад.** Нехай розглядається реальний фізичний процес, який, з точки зору експертів, досить добре описується п'ятьма змінними у виді

**z = f (x, y, factor1, factor2),**

де z - залежна змінна, показник виходу процесу;  
 x - перша безперервна незалежна змінна;  
 y - друга безперервна незалежна змінна;  
 factor1 - перший незалежний чинник, що набуває всього два значення m і s;  
 factor2 - другий незалежний чинник, що набуває всього два значення l і d.

     Розглянемо рішення задачі загальної нелінійної регресії,  в якій незалежними змінними виступають як безперервні змінні, так і категоріальні (рівні чинників).  
      Зазвичай в таких випадках розмова зводиться до вербальної постановки  завдання на рівні "запропонувати методику побудови залежності змінною А від змінних Би, З, Д. з максимальною точністю і, бажано, якнайшвидше".  
         За всю історію спостереження за процесом, накопичився масив даних, які збережені в таблиці системи STATISTICA. Вид тієї табл. , що відповідає:

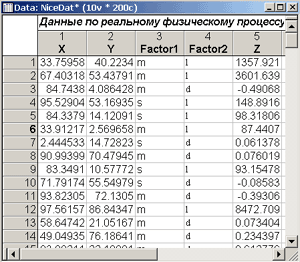


Рис.1. Таблиця початкових даних про процес.  
Метою дослідження є побудова математичної моделі фізичного процесу на  основі початкових даних, яка по заданій четвірці значень **(x, y, factor1, factor2)**  видавала б **відгук Z** з точністю не гірше 5 %.

Як вже згадувалося вище, методику рішення задачі необхідно підготувати в максимально короткий термін. Проте, тут є ряд неприємностей, які значно розтягують процедуру пошуку рішення. Найважливішим з них є формулювання гіпотези про явний вид залежності, що вивчається, яка зовсім не є очевидною.

Для ілюстрації сказаного приведемо згладжену поверхнею залежність  **z = f (x, y)** для двох рівнів змінної factor1.

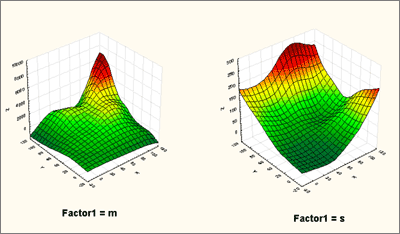


Рис.2. Вид залежності z = f (x, y) для двох рівнів factor1.  
 По виду поверхні, зображеній на рис.2, відразу, без додаткових досліджень, про явний вид залежності сказати щонебудь складно. Тим більше, слід згадати, що ми не врахували ще один чинник. Загалом, рішення подібної задачі методами нелінійного оцінювання може розтягнутися дуже надовго, а може так ні до чого і не привести. У таких критичних ситуаціях, коли відомо, що

- залежність між змінними є;  
- залежність безперечно нелінійна;  
- про явний вид залежності сказати що-небудь складно  
застосовуються нейромережеві алгоритми. Розглянемо спосіб рішення цієї задачі в модулі STATISTICA Neural Networks.

**1. Технологія регресійного аналізу.**Дуже поширеною є наступна теза: "нейронні мережі є універсальною структурою, що дозволяє реалізувати будь-який алгоритм". Спробуємо побудувати нейронну мережу, яка б "уловила" пропоновану залежність відразу (мається на увазі, без попереднього, розвідувального аналізу). Тим більше що експерти, що спостерігають за цим фізичним процесом, з повною упевненістю стверджують, що залежність є. Здійснюватимемо пошук мережі, на вхід якої подаються чотири чинники (x, y, factor1, factor2), а на виході знімаються значення z.

Одним з найважливіших питань, досі, не вирішених сучасною наукою, є питання про структуру нейронний мережі, яка була б здатна до відтворення шуканої багатовимірної нелінійної залежності. Та дійсно, теорема Колмогорова про повноту, доведена їм ще 1957 року, стверджує, що нейронна мережа здатна відтворити будь яку (**дуже важливо - безперервну**) функцію.

У 1988 році, ряд авторів узагальнили теорему Колмогорова і показали, що **будь-яка безперервна функція може бути апроксимована тришаровою  нейронною мережею з одним прихованим шаром і алгоритмом зворотного поширення помилки з будь-якою мірою точності"** ….

         Таким чином, в нашому випадку позитивним аспектом є знання того, що мережа має бути **тришаровою**, але зновутаки в розпорядженні немає правил,  що встановлюють залежність між "будь-якою мірою точності" і кількістю нейронів на проміжному, так званому  прихованому шарі.  
Резюмуючи усе вищесказане, відмітимо, що універсальних правил, що говорять про те, якої топології нейронної мережі варто дотримуватися для вирішення того або іншого завдання немає. Тому, потрібна розумна процедура пошуку потрібної мережі.

Модуль Neural Networks системи STATISTICA включає унікальну процедуру,  організуючу пошук потрібної конфігурації мережі. Цей інструмент називається Intelligent Problem Solver. Скористаємося цим інструментом і здійснимо пошук нейронної мережі, яка буде здатна до рішення нашої задачі.  
Для запуску модуля Neural Networks скористаємося однойменною командою основного меню системи STATISTICA - Statistics. Ця команда приведе до виклику стартової панелі модуля STATISTICA Neural Networks (SNN).

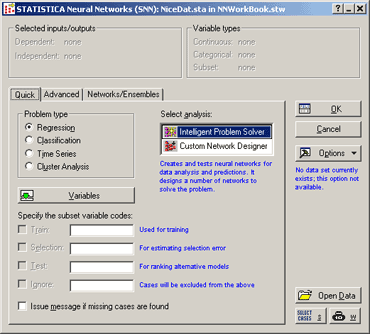


Рис.3. Стартова панель модуля SNN.

У вкладці Quick - Швидкий цього діалогового вікна доступні три, найчастіше використовувані опції. У розділі Problem Type - Клас завдань пропонується здійснити вибір того класу завдань, з яким ми зіткнулися. Нашою метою є побудова багатовимірної залежності або, іншими словами, багатовимірній регресії. Значить, в розділі Problem Type - Клас завдань слід вказати Regression - **Регресія.**

Для того, щоб вибрати змінні, скористаємося кнопкою Variables. При натисненні цієї кнопки з'являється діалогове вікно Select input (independent), output (dependent) and **selector** **variables** - Вкажіть вхідні (незалежні), вихідні (залежні) і групуючі змінні. У цьому діалоговому вікні необхідно задати три списки змінних. Continuous outputs - що Безперервні, що виходять, в нашому випадку, - це змінна z. Continuous inputs - що Безперервні, що входять, в нашому прикладі, - це змінні x і y. А так само Categorical inputs - що Категоріальні, що входять, у нас це змінні **Factor1 і Factor2**. Розділ Subset variable - Розбиття на підмножини є необов'язковим для заповнення і служить для вибору змінній, в якій містяться коди для розбиття даних на повчальне контрольне і тестове множ.

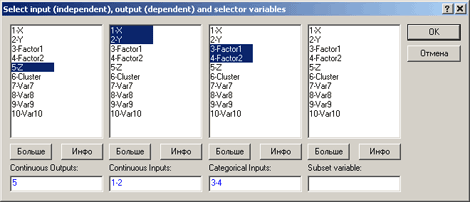


Рис.4. Вибір змінних для Аналізу.

У розділі Select analysis - Вибір аналізу доступні дві опції Intelligent Problem Solver і Custom Network Designer. На цьому етапі нам потрібно першу опцію, яка встановлюється за умовчанням. Усі необхідні нам параметри вибрані. Для продовження Аналізу натиснемо кнопку OK.  
На наступному кроці з'являється діалогове вікно налаштування процедури Intelligent Problem Solver. Це вікно містить велику кількість опцій, розподілених в різних вкладках. На цьому етапі нам знадобиться вкладка Quick - Швидкий і її розділ Optimization Time - Час оптимізації.

Цей розділ містить групу опцій, що відповідають за час виконання алгоритму пошуку нейронної мережі. Є дві можливості:  
1. задати кількість мереж, які необхідно протестувати (з'ясувати чи підходять вони для вирішення сформульованого завдання);  
2. вручну задати час виконання алгоритму. Для цього необхідно скористатися опцією Hours/Minutes - Годинник/Хвилини.  
Скористаємося першою опцією. У розділі Optimization Time - Час оптимізації в розділі Networks tested - Кількість тестованих мереж вкажемо 100. Усі інші параметри залишаються не змінними.

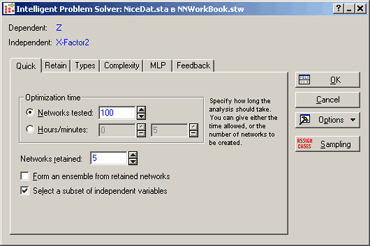


Рис.5. Завдання кількості тестованих мереж.

Для запуску процедури пошуку мереж натиснемо ОКИ.  
Стан алгоритму пошуку відображається в діалоговому вікні IPS Training In Progress - Процес пошуку мережі, см Рис.6.

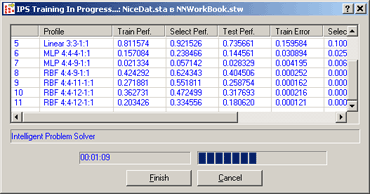


Рис.6. Процес пошуку мережі.

Під час роботи алгоритму пошуку відповідної нейронної мережі в цьому діалоговому вікні наводиться інформація про час виконання алгоритму, а так само про розглянуті нейронні мережі. Мета алгоритму пошуку - перебір ряду нейромережевих конфігурацій і вибір найкращої з точки зору мінімуму помилки на виході мережі і максимуму її продуктивності. Відповідно, для здійснення процедури порівняння мереж, мережі необхідно навчати, розраховувати їх помилки і продуктивності і ці показники, потім, порівнювати. В результаті, кожна навчена і перевірена нейронна конфігурація описується рядком в інформаційному полі діалогового вікна. Найбільш важливими показниками тут є Profile - Тип мережі, Train (Select, Test) Performance - Продуктивність мережі на повчальній (контрольному, тестовому) множині і Train Error – *Помилка навчання*

   У розділі ***Profile - Тип мережі*** описується топологія  нейронної мережі, тобто клас мереж, до якого вона належить (персептрон, мережа RBF і так далі), кількість вхідних і вихідних змінних, кількість прихованих шарів і число елементів на кожному прихованому шарі.  
Для навчання алгоритм пошуку мережі розбиває (за умовчанням) усю безліч  спостережень на *Training - Повчальне, Selection - Контрольне і Test – Тестове  множини.*Кожна з цих множин несе свою важливу функцію.  
На повчальній множині відбувається безпосереднє навчання мережі, тобто зміна вагових коефіцієнтів кожного з нейронів пропорційно помилці на виході. Відповідно усі спостереження з цієї множини багаторазово беруть участь в процедурі зміни вагових коефіцієнтів навчаної мережі.

Спостереження контрольної великої кількості в процедурі зміни вагів нейронів не беруть участь. Основна функція цих даних в постійному контролі здатності мережі до узагальнення даних, на яких вона не навчалася. Така процедура називається крос - перевіркою. На кожному кроці алгоритму навчання розраховується помилка для усього набору спостережень з контрольної множини і порівнюється з помилкою на повчальній множині. Природно, що ці помилки відрізнятимуться. Як правило, помилка на контрольній множині перевищує помилку на повчальній множині. Проте, важливий не сам факт відмінності, а спостережувана  тенденція похибок.

Дійсно, алгоритм навчання націлений на мінімізацію помилки на виході мережі. Відповідно, помилка на повчальній множині просто зобов'язана в середньому зменшуватися. Але ніхто не обіцяє спаду помилки на перевірочній множині. Отже, якщо спостерігається картина зростання помилки на контрольній множині, тоді як вона зменшується на повчальній множині, то цього говорить про те, що мережа "зазубила" усі пред'явлені їй спостереження і не здатна до узагальнення.

Такий стан називається ***перенавчанням****.* Бажано перенавчання уникати. Алгоритм Intelligent Problem Solver самостійно відстежує перенавчання і при завершенні навчання повертає мережу в найкращий стан (Retain Best Network - Відновити найкращу мережу).  
Тестова множина не бере участь в навчанні взагалі. Воно використовується після завершення навчання для розрахунку продуктивності отриманої мережі і її помилки на даних, про які "їй взагалі нічого невідомо". Хорошою мережею можна вважати ту мережу, у якої помилка однаково мала на усіх трьох підмножинах.

Під продуктивністю мережі в завданні регресії розуміється відношення стандартного відхилення помилок мережі до стандартного відхилення початкових  даних (SD - ratio).  Емпіричне правило свідчить, що **якщо SD - ratio не перевищує значення 0.2,**  мережа підібрана добре. Продуктивність розраховується для кожної з трьох підмножин. Бажано, щоб розкид значень продуктивності на  кожній з підмножин був невеликим.

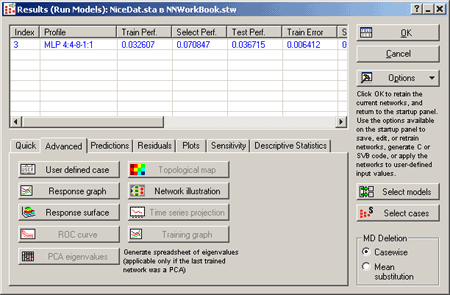


Рис.7. Діалогове вікно результатів пошуку мережі.  
У результаті, нами була відібрана мережа - тришаровий персептрон з 8-у нейронами на прихованому шарі, продуктивністю 0.07. Для того, щоб вибрати потрібну мережу зі списку, пропонованого Intelligent Problem Solver, необхідно скористатися кнопкою Select Models - Вибрати моделі діалогового вікна Results - Результати. У діалоговому вікні, що з'явилося, необхідно підсвітити потрібну нейронну мережу і натиснути ОК. Для того, щоб подивитися ілюстрацію вибраної мережі, скористаємося кнопкою Network Illustration вкладки Advanced - Додатково діалогового вікна результатів.

Графічною ілюстрацією якості роботи нейронної мережі є графік залежності спостережуваних значень вихідний змінної z від передбачених значень. Для побудови такого графіку необхідно скористатися кнопкою - Графік X від Y, розташованою у вкладці Plots, - Графіки. При цьому, заздалегідь вказавши в якості X - axis - Observed - Спостережувані, Y - axis - Predicted - Передбачені. Результат побудови графіку приведений на Рис.9.

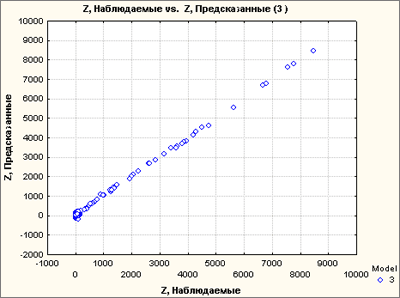


Рис.8. Графік залежності передбачених значень Z від спостережуваних Z.

Якщо модель побудована добре, то точки цього графіку повинні розташовуватися як можна ближче до прямої, що лежить під кутом 45 градусів до осей координат. На перший погляд, так воно і відбувається. Значить, побудована нами нейромережева модель добре узгоджується з даними. Для того, щоб остаточно переконатися в правильності зробленого висновку, розглянемо чисельні результати.

У вкладці Predictions - Прогнози діалогового вікна результатів натиснемо кнопку Predictions - Прогнози для перегляду результатів роботи нейронної мережі на усьому наборі даних. Досить збіглого перегляду отриманої таблиці (см Рис.9), щоб переконатися в тому, що не все так добре, як хотілося б.

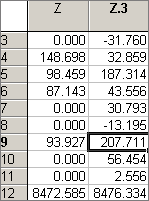


Рис.9.Таблиця результатів прогону усього набору даних через нейронну мережу. У лівій колонці таблиці відображаються***істинні*** значення. У правій -**відгу*к***  нашої нейронної мережі на відповідну четвірку незалежних змінних. У цій області помилки колосальні. У області значень z, що не перевищують 100, результати так само сильно суперечливі. Мережа більша - менш непогано проявляє себе в області великих значень z, близько 1000. Підсумок: необхідної погрішності  5% усіх даних досягти не вдалося.

Необхідно вживати заходи по зниженню помилок.

**ЛІТЕРАТУРА**  
 1. Хайкин С. Нейронні мережі: повний курс, 2-е видання. Пер. з англ. - М.:  Видавничий дім "Вільямс", 2006 р. - 1104с.  
 2. Боровиков В.П. STATISTICA Neural Networks - Технічний опис. М.: Світ, 1999. - 239 с.