Звіт з теми:

“Прогнозування розміру доходів”

Виконали студенти групи МІ-4 Омельяненко Андрій, Стецик Олексій

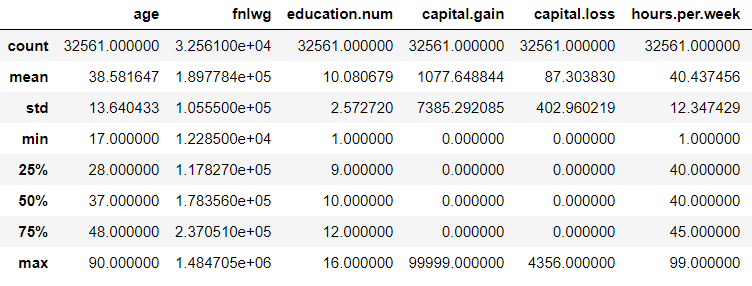
**Аналіз та формалізація задачі**

Задача полягає в аналізі та класифікації заробітної плати дорослих в США, використовуючи UCI Machine Learning Repository Adult data set(<http://archive.ics.uci.edu/ml/datasets/Adult>).

В нашому розпорядженні є дані про 32560 людей. Кожна людина характеризується 14 атрибутами(8 номінальних та 6 числових) та бінарним значенням target(0 – людина отримує менше 50000 $/year, 1 -не менше).

Числові атрибути: вік, фінальний коефіцієнт(~кількість людей з такими характеристиками в країні), найвищий рівень освіти(відсортований), отриманий капітал, втрачений капітал, кількість робочих годин на тиждень.

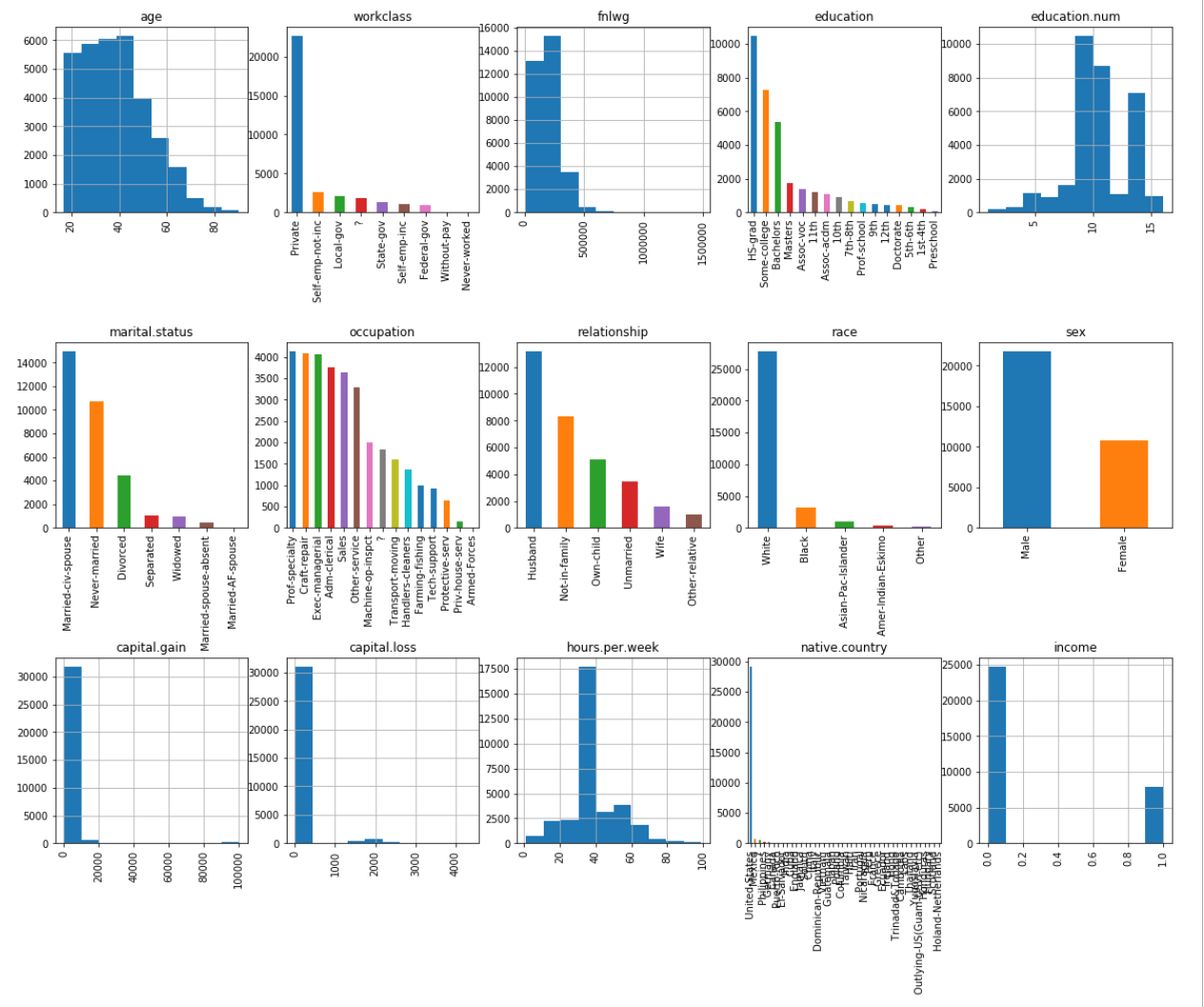
Статистика по числових атрибутах:

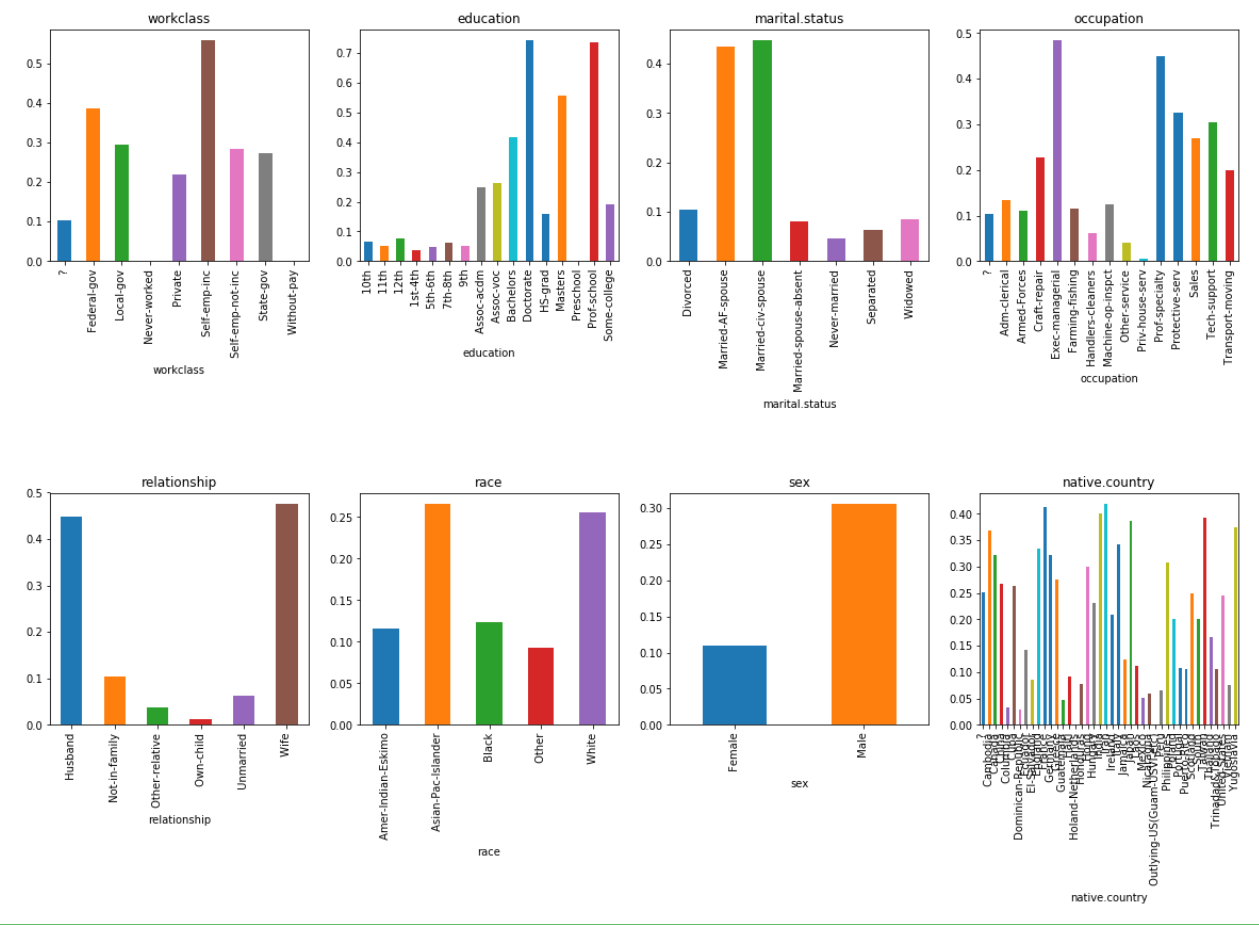


Номінальні атрибути: робочий клас(описує тип роботи(самостійний, державний, приватний і тд), найвищий рівень освіти, сімейний статус(одружений, неодружений і тд), вид діяльності(сільське господарство, менеджмент і тд), відносини(неодружений, дружина і тд), раса, стать, батьківщина.

Статистика по номінальних атрибутах:

* Освіта: 32% закінчили школу, 22% були студентами коледжу і 16.5% мають ступень Бакалавра.
* Сімейний стан: 46% одружені, 33% ніколи не були одруженими, 14% розлучені.
* Відносини: 40% одружені чоловіки, 26% не в сім’ї.
* Раса: 86% білі, 10% чорні.
* Стать: 67% чоловіки, 33% жінки.
* Батьківщина: 90% США, 2% невідомо

Розглянемо розподіл кожної функції, щоб ми могли краще розуміти, що відбувається в наших даних. Зображаємо кількість рядків кожного типу для кожного атрибуту.

Також розглянемо який відсоток людей для кожного значення кожного атрибута має >50000$/year(0.3 ~ 30%). 

Наступним кроком розглянемо кореляцію між різними атрибутами. Рахувати кореляцію між номінальними атрибутами неможливо, тому ми їх закодуємо. Це робиться з-за допомогою LabelEncoder з sickit-learn. Для кращого результату ми для кожного атрибуту сортуємо можливі значення в порядку зростання в залежності від % людей що мають більше 50000$/year.

Приклад до workclass, аналогічно робиться для всіх атрибутів:

workclass

Never-worked 0.000000

Without-pay 0.000000

? 0.104031

Private 0.218673

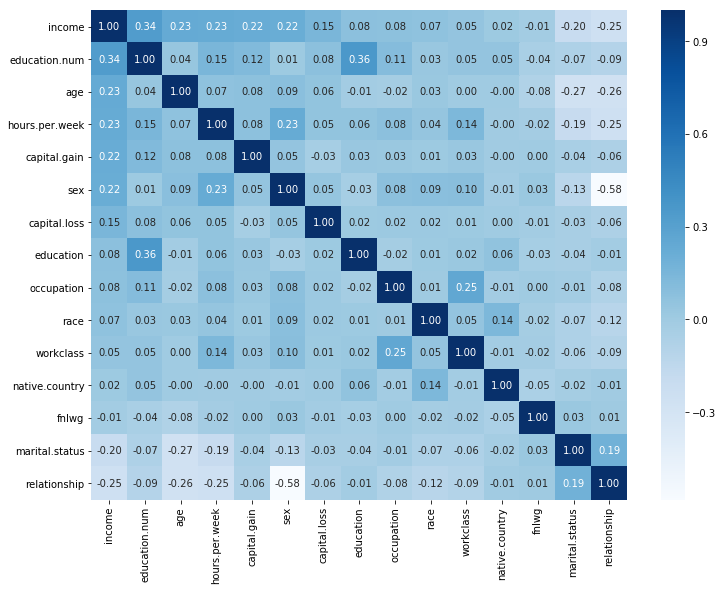
State-gov 0.271957

Self-emp-not-inc 0.284927

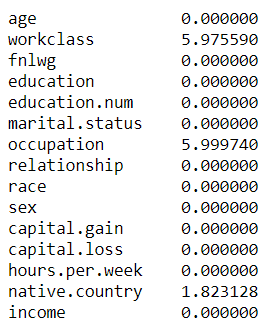
Local-gov 0.294792

Federal-gov 0.386458

Self-emp-inc 0.557348

Матриця кореляції між кожною парою атрибутів. 

**Попередня обробка даних**

В першу чергу перевіримо якісь наших даних: 

Отже, є 3 атрибути з невідомими/відсутніми значеннями. Робочий клас(6%), батьківщина(2%) та вид діяльності(6%). Робочий клас в 70% випадків є приватним сектором, тому можна для кожного з цих 6% приписати приватний сектор. Аналогічно, батьківщиною в 90% випадків є США, тому можна приписати США для всіх невідомих значень. Отже, для 2х атрибутів невідомі значення замінились медіанним значенням на вибірці. На жаль, для виду діяльності не можна провести підчистку аналогічним чином, оскільки цей атрибут не має переважаючої характеристики та може бути дуже важливим. Оскільки вхідних невідомих даних цього атрибуту всього 6% то їх можна просто видалити.

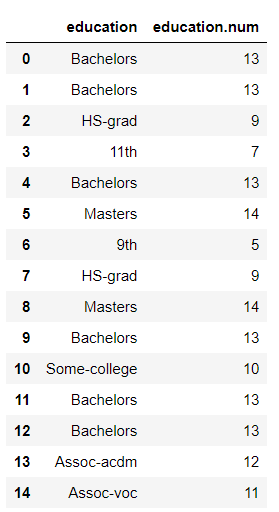
Крім цього дані містять 47 рядків-дублікатів, їх можна просто видалити.

На жаль, існує дисбаланс у пропорції атрибуту-таргету, маємо 75% <= 50K та 25% >50K, але цього достатньо для побудови достатньо точних класифікаторів.

Для використання моделей машинного навчання необхідно щоб кожен атрибут був числовим, тому переведемо *income* та *sex* в бінарні числа наступним чином:

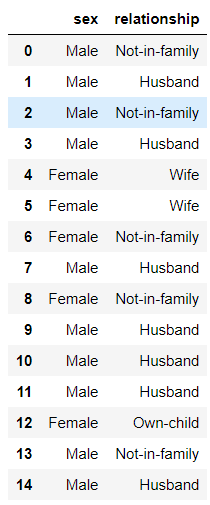
* data['income'] = data['income'].map({' <=50K': 0, ' >50K': 1}).astype(int)#(Якщо <=50000$/year то 0, інакше 1)
* data['sex'] = data['sex'].map({' Female': 0, ' Male': 1}).astype(int)#(Якщо жінка то 0, інакше 1)

Можна побачити велику додатну кореляцію між education та education.num, дивимось що означають ці стовпці:

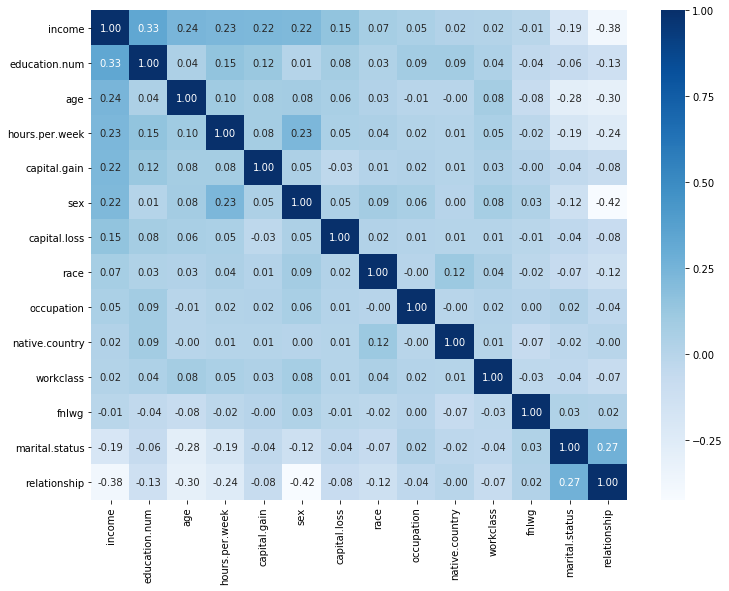


По суті це одне й теж, але в education.num ми маємо властивість чисел між собою: більше число – більший освітній рівень, тому видалимо рядкове позначення, а числове залишимо, адже їхня властивість може бути корисною для моделей.

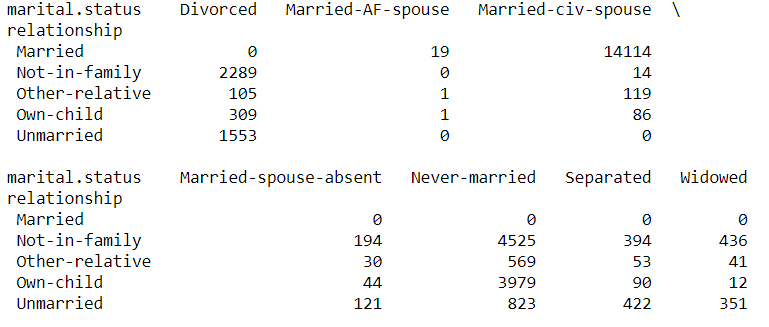
Побачимо також велику від’ємну кореляцію між відносинами та статтю, поглянемо чому так:



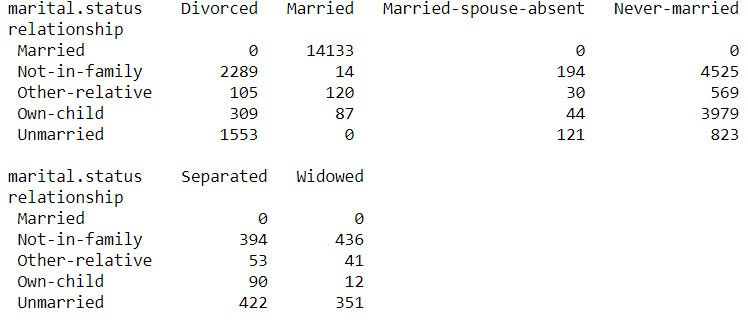
Зрозуміло що ця кореляція з’являється між Male-Husband та Female-Wife, тому виконаємо наступну підміну: Husband, Wife -> Married. Таблиця кореляції змінилась: кореляція між *sex* та *relationship* зменшилась.



Але кореляція між *relationship* та *marital.status* збільшилась, подивимось чому:

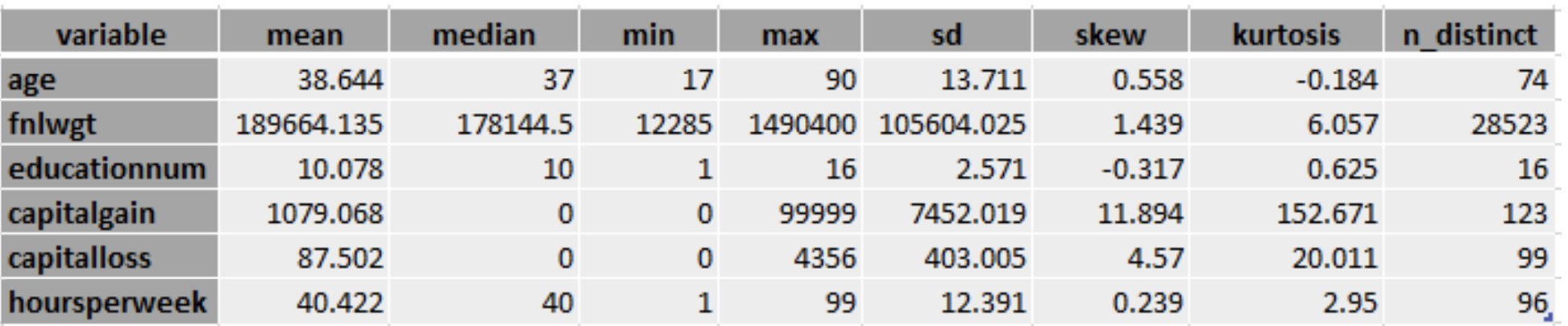


Бачимо що Married-AF-spouse майже завжди Married, тому просто заміняємо Married-AF-Spouse, Married-civ-spouse на Married, отримуємо:



Бачимо що є люди які не одружені в одному атрибуті, але одружені по-іншому атрибуту. Цим і подібними неточностями можна знехтувати, оскільки це присутнє в невеликій кількості рядків.

Після реалізації багатьох machine learning моделей, ми порівняли їхні результати на тестовій і тренувальній множині, щоб побудувати модель яка працює найкраще для завдання класифікації на train i test вибірці. У всьому процесі, ми користувалися минулими роботами, які використовували даний датасет.



.Детальний аналіз дати був проведений з допомогою jupyter notebook.

Посилання на сам notebook: <https://github.com/Omelianenko/adult_salary_prediction/blob/master/adult_prediction.ipynb>

Використання різних моделей.

Логістична регресія. Логістична регресія не враховує зв’язок між різними фічами і тому її використання дає менший результат чим результат інших моделей. Як тільки ми використали нейрону мережу і побачили збільшення в точності, ми зрозуміли що є зв’язок між різними фічами.

Наївний Баєс. Зрозуміло, що наївний Баєс дав погану точність оскільки було використано припущення про незалежність між фічами.

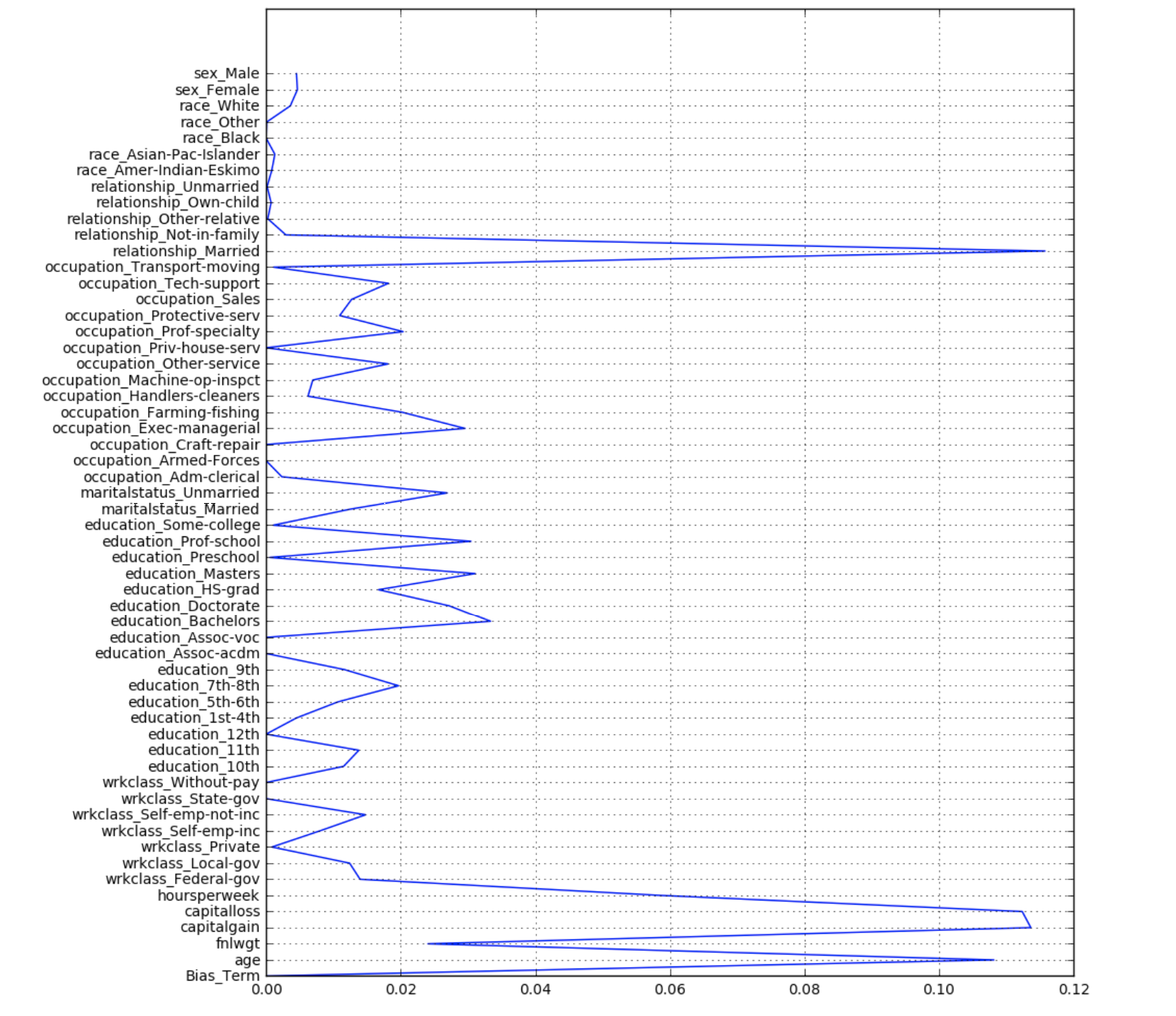
Нейроні мережі. Нейроні мережі додали зв’язок якого не було в логістичній регресії і тому отримали кращий результат. Після того як ми перепробували велику кількість прихованих шарів, ми знайшли що число схованих шарів 5 дає нам найкращу оцінку. При зростанні кількості схованих шарів, зростання далі точності йде дуже повільно.

SVM. Свм-ка в загальному працює добре з нелінійними даними. Ми зауважили нелінійність в деяких наших змінних такия як capial gain/loss і звідси ми зробили висновок що доцільно попробувати застосувати SVM-ку.

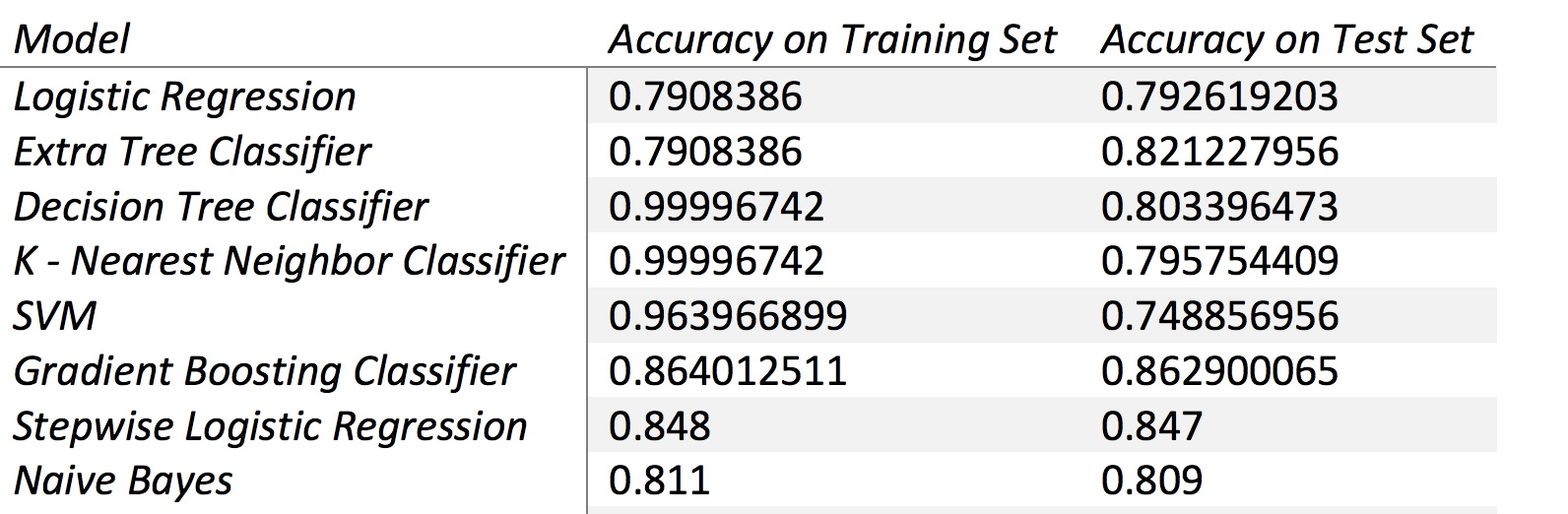
K найближчих сусудів. Використовуючи KNN класифікатор для нашого завдання, ми серйозно покращили точність, але нам потрібно було шукати точне значення к. Ми знайшли що к = 1 найкраще працює для трейн сету і к = 24 працює найкраще для тест сету.

Gradient Boosting Classifier. Також попробували застосувати дану модель. Як виявилося вона спрацювала найкраще. Також вона допомогла нам краще зрозуміти, що найбільше впливає на нашу класифікацію і як виявилося це такі характеристики як Married, capital gain, hours per week і age.

Власне ось графік того, що ми отримали:



Представимо також точність яка була досягнута:



**Висновок**. Ми заоохочені нашими результатами. Ми отримали хороші результати з штучними нейроними мережами з 5 схованими шарами, і дуже хороший результат для Gradient Boost classifier. Подальші вдосконалення можна робити підкручуючи параметри, або декомпозицію одної фічі в декілька(наприклад бінарних).