Міністерство освіти і науки України
НТУУ «Київський політехнічний інститут»
Фізико-технічний інститут

Спеціальні розділи програмування

Курсова робота

MLP для класифікації зарплати навчений на даних перепису населення

Виконав:
Студент 2 курсу ФТІ
Групи ФІ-92
Поночевний Назар Юрійович

Перевірив:

_____

Прийняв:

_____

1 Мета роботи

Отримати досвід використання основних методів та засобів аналізу та візуалізації даних у середовищі розробки IPython Notebook [1] на реальних наборах даних.

2 Завдання

На підставі даних перепису спрогнозувати, чи перевищуватиме дохід особи $50 тис. на рік.

3 Датасет

У роботі було використано Census Income (Adult) датасет. [4] В ньому є 2 файли (.data, .test), проте вони не збалансовані, тому файли були зконкатеновані для самостійного балансування. Було отримано (48841 рядків x 15 колонок) dataframe з різними типами даних та відсутніми значеннями.

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
0	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
1	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
2	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
3	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
4	37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White	Female	0	0	40	United-States	<=50K

Рис. 1: Датасет (перші 5 рядків)

4 Очищення даних

Перш за все, замінюємо відсутні дані та балансуємо набір даних, видаляючи деякі рядки з більшості ('<= 50K'). Ця процедура зменшила dataframe до (11687 рядків x 15 стовпців). Після цього виконуємо нормалізацію, стандартизацію та OneHot-кодування даних за допомогою цієї карти (карта була побудована з використанням гістограм кожного стовпця кадру даних):

```
norm_map = {'age': 'standartization', 'workclass': 'onehot',
'fnlwgt': 'normalization', 'education': 'onehot',
'education-num': 'standartization', 'marital-status': 'onehot',
'occupation': 'onehot', 'relationship': 'onehot',
'race': 'onehot', 'sex': 'onehot',
'capital-gain': 'normalization',
'capital-loss': 'normalization', 'salary': 'onehot',
'hours-per-week': 'standartization', 'native-country': 'onehot'}
```

5 Візуалізація даних

Розглянемо кореляційні зв'язки Пірсона: [2]

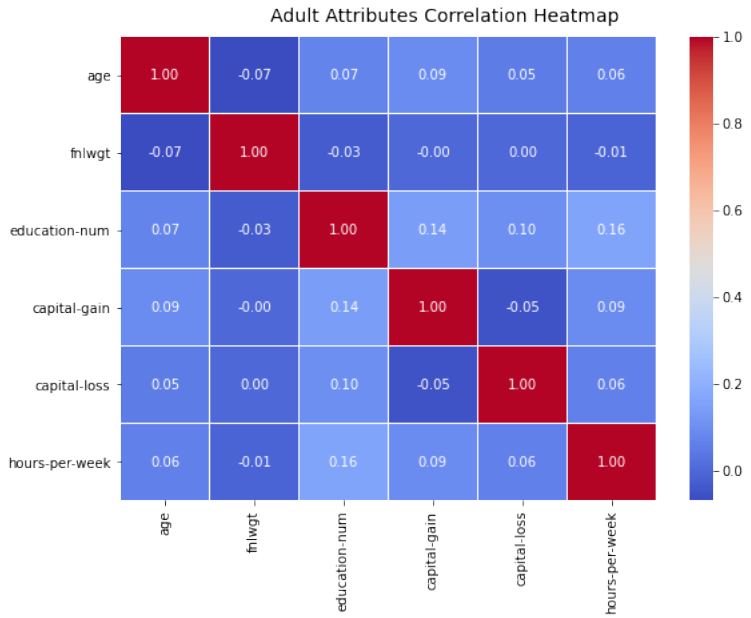

Рис. 2: Теплова карта кореляції

Існує невелика кореляція, тож, подивимось, як категоричні дані будуть відокремлювати значення "Зарплата":

Розглянемо як залежать один від одного інші категоріальні змінні:

Також побудуємо графік щоб побачити розподіл за змінною "Вік":

Розглянемо як залежать 4 змінні одночасно:

6 Класифікація

Побудуємо моделі машинного навчання

1. Перш за все розділяємо дані

```
X shape: (23374, 104)
Y shape: (23374, 2)
Training X shape: (15193, 104)
Training Y shape: (15193, 2)
Test X shape: (8181, 104)
Test Y shape: (8181, 2)
```

2. Створюємо моделі

```
# kNN
knn = KNeighborsClassifier()
knn_model = MultiOutputClassifier(estimator=knn)
```

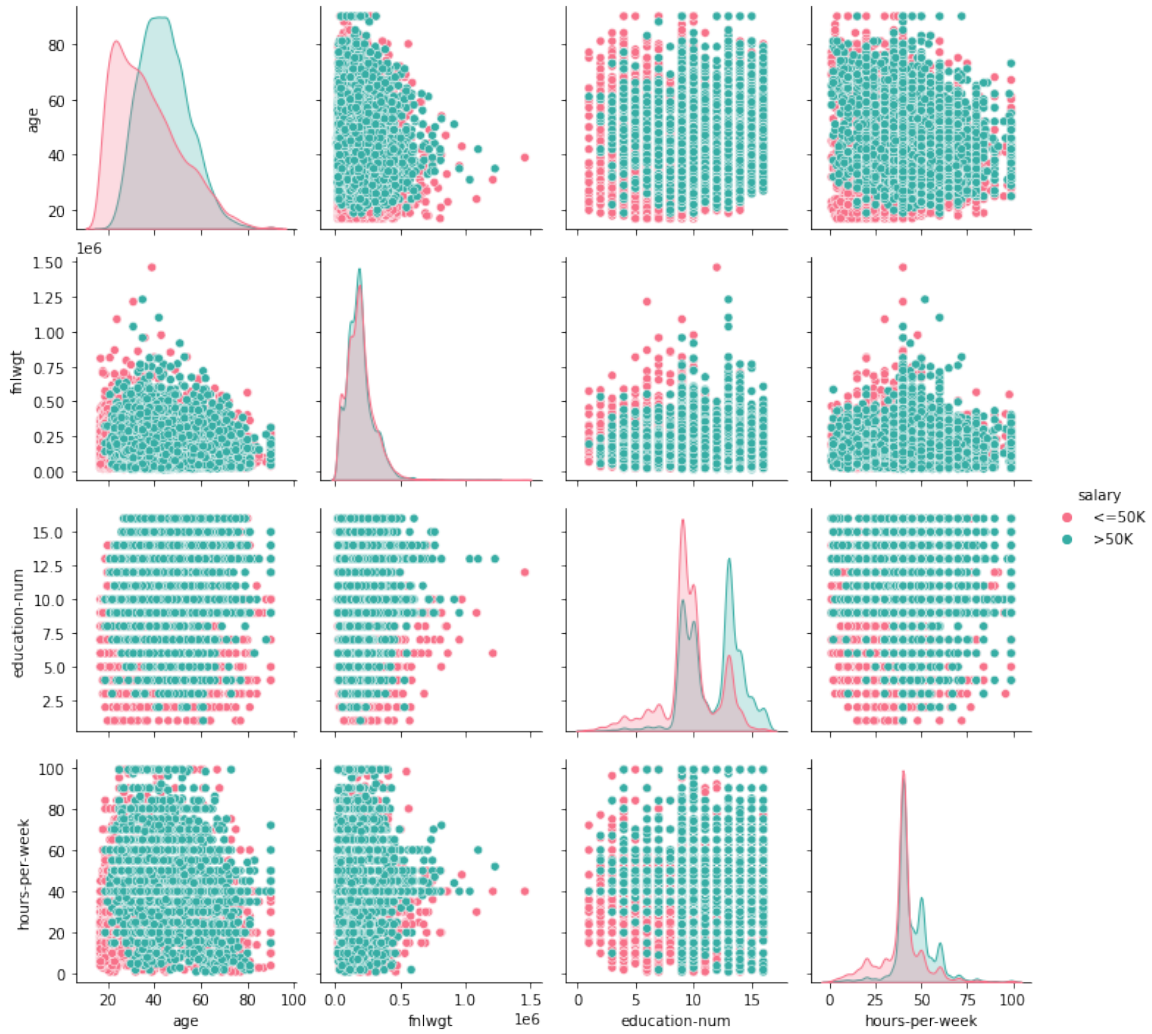

Рис. 3: Кореляції з категоричними ознаками

```
# Naive Bayes
nb = GaussianNB()
nb_model = MultiOutputClassifier(estimator=nb)

# SVM
svm = SVC(kernel='rbf', C=1e3, gamma=0.1)
svm_model = MultiOutputClassifier(estimator=svm)

# DecisionTree
dtree = DecisionTreeClassifier()
dtree_model = MultiOutputClassifier(estimator=dtree)

# RF
rf = RandomForestClassifier(n_estimators=10)
rf_model = MultiOutputClassifier(estimator=rf)
```

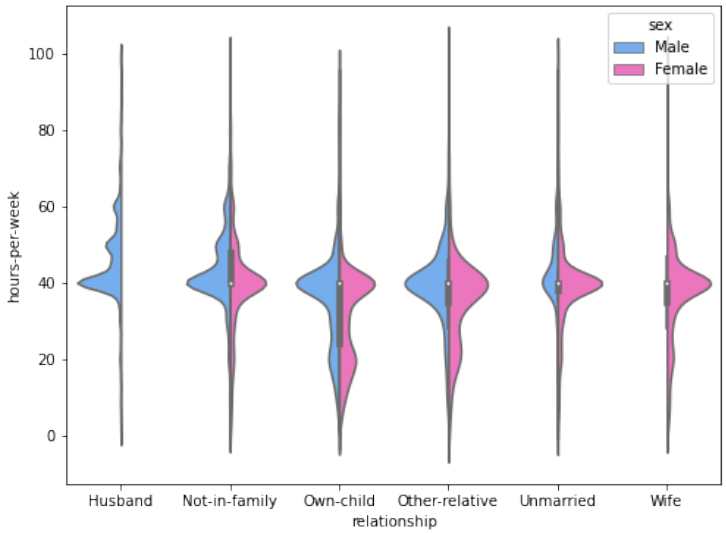

Рис. 4: Кореляції з іншими категоричними ознаками

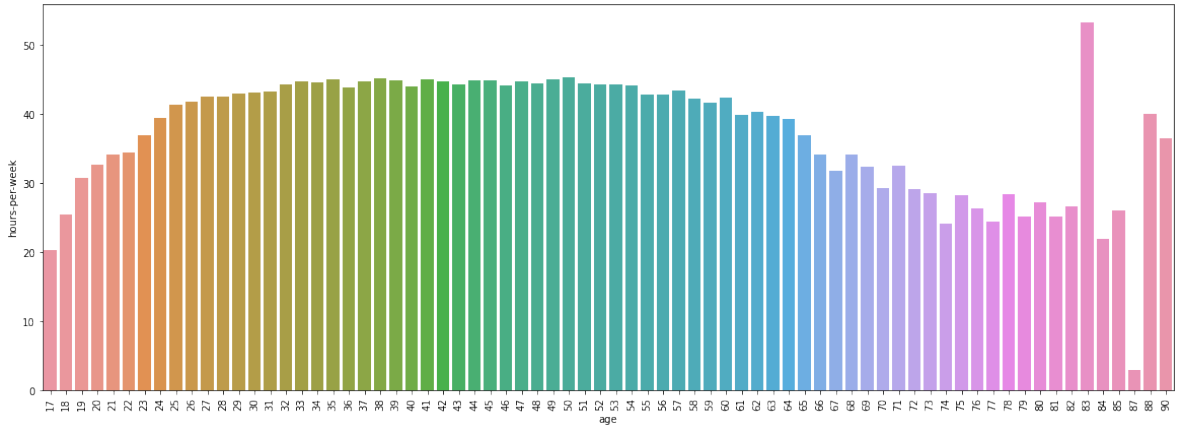

Рис. 5: Розподіл за змінною "Вік"

```
# MLP
x_shape, y_shape = X.shape[1], Y.shape[1]
mean_shape = (x_shape + y_shape) // 2

mlp_model = Sequential()
mlp_model.add(Dense(x_shape, input_shape=(x_shape,),
activation='relu'))
mlp_model.add(Dense(mean_shape, activation='relu'))
mlp_model.add(Dense(y_shape, activation='softmax'))
es = EarlyStopping(monitor='val_accuracy', verbose=1,
patience=5)
mlp_model.compile(loss='categorical_crossentropy',
optimizer='adam', metrics=['accuracy'])
```

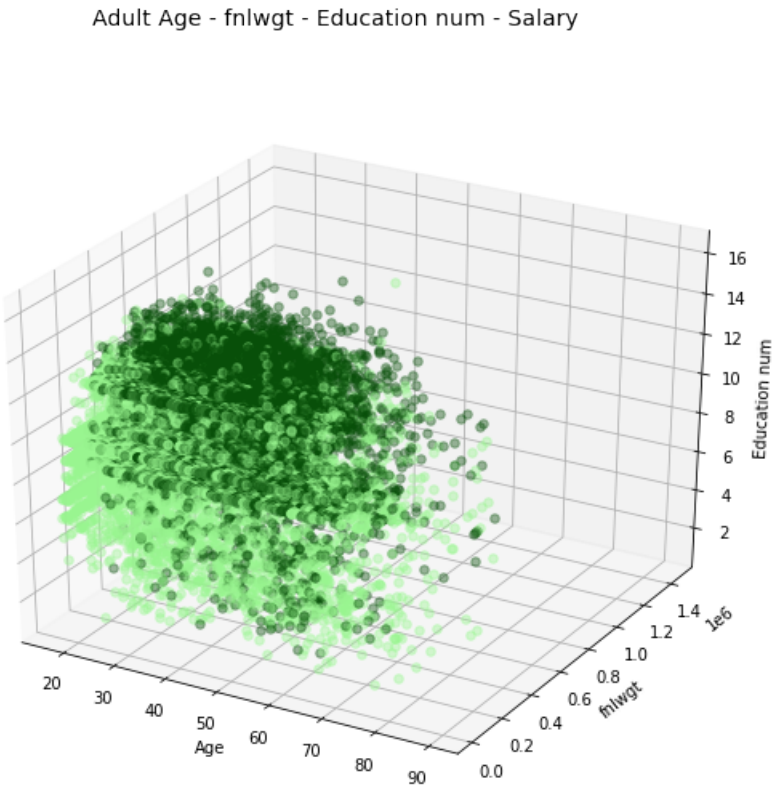

Рис. 6: Залежність зарплатні від віку, років навчання та вагового коефіцієнту

3. Навчаємо та оцінюємо

```
kNN Test accuracy: 0.790
Naive Bayes Test accuracy: 0.699
SVM Test accuracy: 0.784
DecisionTree Test accuracy: 0.744
RF Test accuracy: 0.745
MLP Test accuracy: 0.815
```

4. Дивимося матрицю Confusion

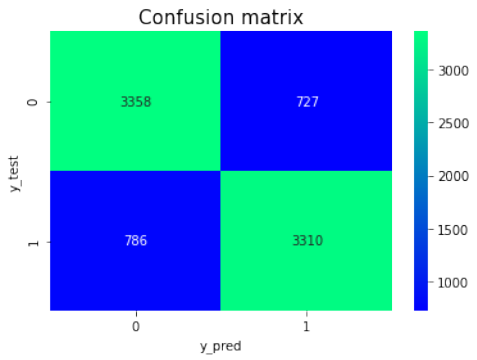

Рис. 7: Матриця Confusion

5. Дивимося важливість ознак

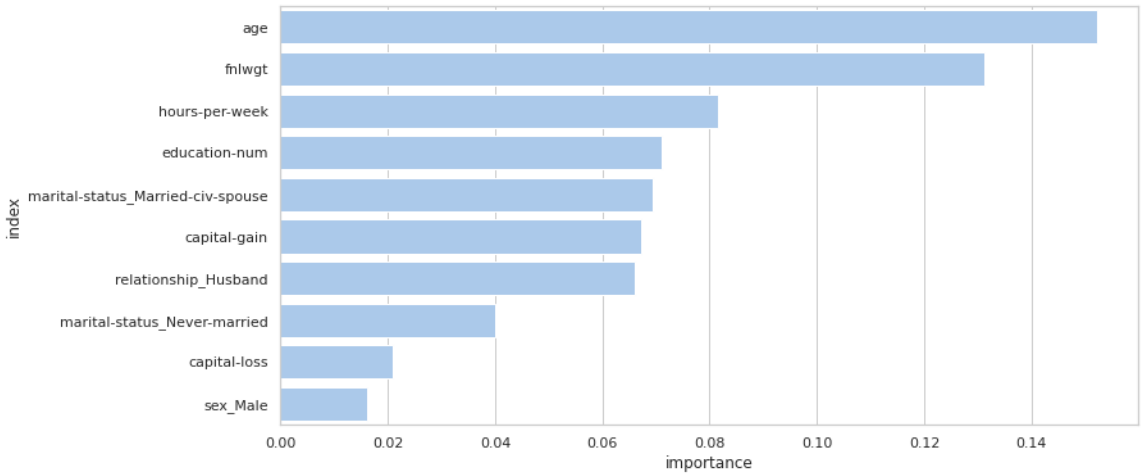

Рис. 8: Важливість ознак

7 Побудова більш простої моделі

Тепер нам відомі найважливіші особливості (вік, ваговий коефіцієнт, години на тиждень, номер освіти, сімейний стан, стосунки) та модель з найбільшою точністю (MLP), тому побудуємо спрощену модель MLP, але з тією ж акуратністю на тестових даних.

1. По новому розділяємо дані

```
X shape: (23374, 17)
Y shape: (23374,)
Training X shape: (15193, 17)
Training Y shape: (15193,)
Test X shape: (8181, 17)
Test Y shape: (8181,)
```

2. Створюємо нову модель

Model: "sequential_1"

Layer (**type**)	Output Shape	Param #
dense_3 (Dense)	(None, 17)	306
dense_4 (Dense)	(None, 9)	162
dense_5 (Dense)	(None, 1)	10

Total params: 478

```
Trainable params: 478
Non-trainable params: 0
_________________________________________________________________
```

3. Навчаємо та оцінюємо

```
MLP Test accuracy: 0.806
```

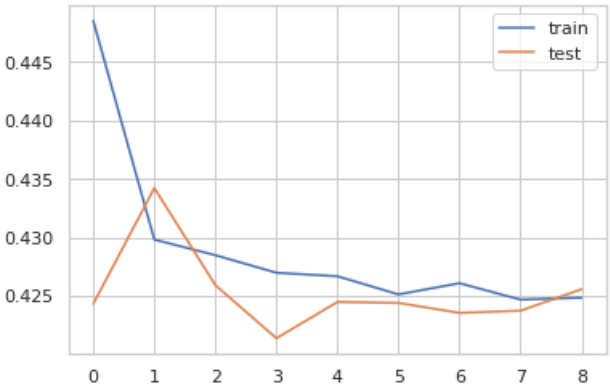

Рис. 9: Навчання нової моделі

8 Висновок

За результатами роботи було отримано досвід використання основних методів та засобів аналізу та візуалізації даних у середовищі розробки IPython Notebook на реальних наборах даних і побудовано ефективну модель багатошарового персептрону, яка на підставі даних перепису прогнозує, чи перевищуватиме дохід особи $50 тис. на рік. Ваги нейронної мережі були опубліковані на GitHub-репозиторії для публічного використання. [3]

Література

[1] Anaconda Inc. Anaconda website.

[2] Albert Sanchez Lafuente. Complete guide to data visualization with python. *Towards Data Science*, 2020.

[3] Nazar Ponochevnyi. Trained-mlp-for-census-income-classification repository.

[4] Machine Learning Repository. Census income data set.