Machine Learning Model: Linear Regression and Its Application for Salary Prediction

Thearith Ponn, Sithvothy Kiv, Socheat Sek

[1] UPS220974, ICT of UP, pthearith.student@puthisastra.edu.kh
[2] UPS220966, ICT of UP, ksithvothy.student@puthisastra.edu.kh
[3] Advisor, ICT of UP, ssocheat@puthisastra.edu.kh

Abstract

This research paper on machine learning algorithms aims to share the knowledge and real world experience and the idea of how to solve many problems by using statistics modelling for continuous prediction which is “Linear Regression Model ''. The Python programming, Jupyter notebook and scikit-learn library will be mainly used in this research. The simple desktop application will be developed to deploy trained machine learning models by using the Tkinter library too. This machine learning development gave the knowledge and idea in coding and analysis to develop the real algorithm to make the impact from data for real-time decision making. Moreover, this research will be the beginning point of involving to show the direction to researcher, scientist, student for machine learning model development using linear regression to solve salary prediction problem, coding the standard python code, using global well known library such as scikit-learn, Tkinter, and etc.

The knowledge is broad and no limitation but everything is not 100% perfect. The massive growth of technology and science are the necessary parts of human, economic and society. Transferring the quality education of machine learning algorithms to students in computer science and developing the machine learning technique is really important for solving the real economic and society’s challenges step by step with the quality and quantity. To be successful in this research paper, the technique of research from existing works and experience will be implemented by setting the direction and goal of developing and showing the Linear Regression model and its application for salary prediction with good accuracy.

This result of research will be developed in the form of linear regression for salary prediction by using python programming. The result of this linear regression model gave the accuracy about 90% which is a really good outcome. The initial demo development of the trained machine learning algorithm and salary prediction is just a starting point for model accuracy prediction so students and researchers are open to improving this work in the future.

Keywords: *Machine learning, Linear Regression, Prediction, Python.*

1. **Introduction**

Nowaday, the terms artificial intelligence, machine learning, and prediction are shown in many types of business, domain, industry and all around the world. In the machine learning world, there are so many algorithms for supervisor learning, unsupervised learning, simi-supervisor learning or reinforcement learning. In the research paper, a part of the supervisor machine learning algorithm will be detailed which is the linear regression model. The linear regression technique widely used to solve continuous problems such house price prediction, salary prediction and other continuous prediction. A Linear Regression model will be used to predict the salary of employed professionals based on their years of experience. Simple Linear Regression is a type of Regression algorithm that models the relationship between a dependent variable (Y) and a single independent variable (X). The relationship shown by a Linear Regression model is linear or a sloped straight line, hence it is called Linear Regression. The key point in Linear Regression is that the dependent variable must be a continuous or real value. However, the independent variable can be measured on continuous values. The equation of linear regress will be shown in the figure 1 below.

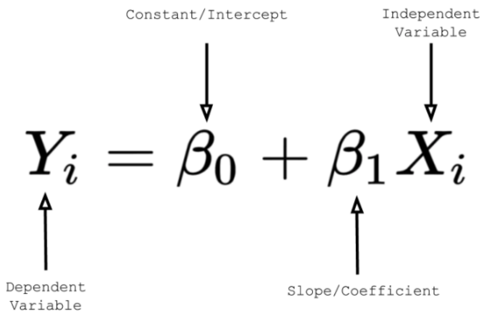


Figure 1: Linear regression equation

The linear regression equation is the way to build the model. Y is the predictor variable so when the input variable X then it will calculate the prediction value. Before achieving the equation the slope (coefficient) and intercept need to be calculated. Slope shows how the level of trend of the regression line is and it will be calculated by each data point (y2 - y1)/(x2-x1) as in figure 2. slope (coefficient) shows the point that it hits the y axis when x=0 that is why it is called y-intercept [1].

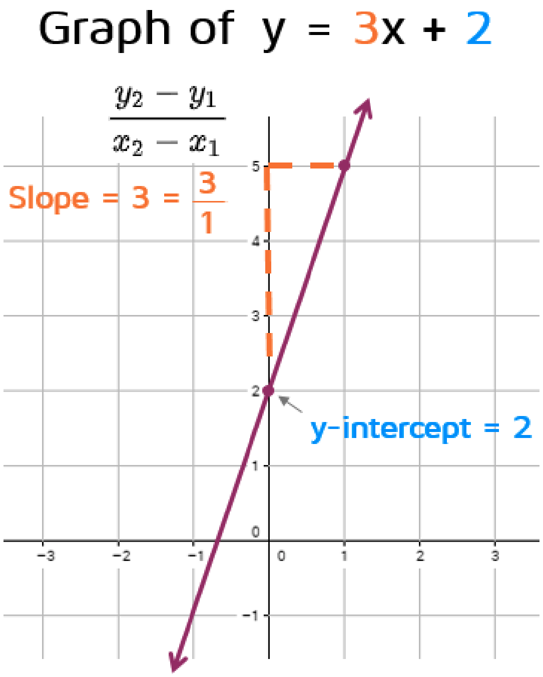


Figure 2: Example of linear regression

In figure 3 below show the type of linear association based on each data point. There are 4 types of linear association which are positive linear association, negative linear association, nonlinear association and no association. For linear regression models can be applied when associations are positive linear association and negative linear association then it produces good accuracy.

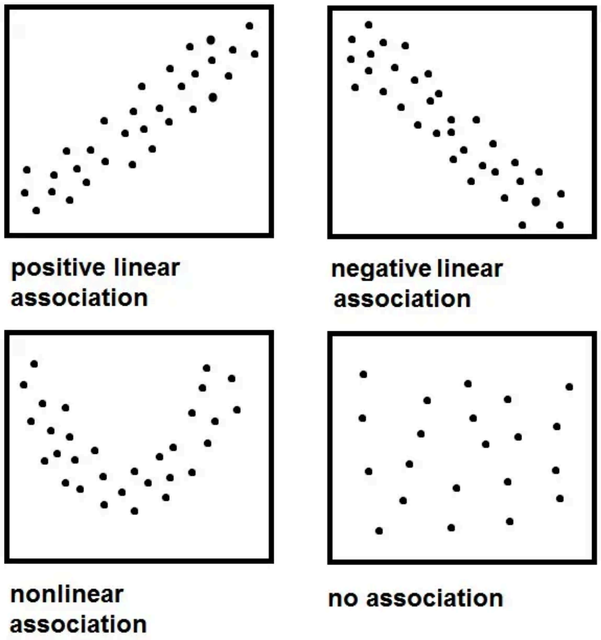


Figure 3: Type of association

Before choosing any machine learning model it is a must to create visualisation to the relation between feature (variable) to other features as shown in figure 4.

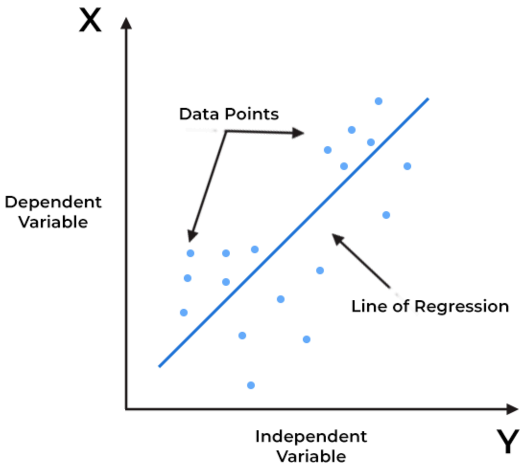


Figure 4: Linear regression

2. Literature Review

The first related work was published by Mr. Sahin Magar published in 2022 on the topic of “Loan Eligibility Prediction using Machine Learning Algorithms” [2]. In banking institutions, loan approval is a crucial process. The system approves or rejects loan applications, and loan recovery significantly impacts a bank’s financial accounts. Predicting whether a customer will repay the loan is challenging. Recently, researchers have been developing algorithms to predict loan approval. Machine Learning (ML) techniques are particularly useful for predicting outcomes in large datasets. This work uses two ML techniques, Support Vector Machine (SVM) and Random Forest (RF), to predict client loan approval.

The second research work is from Mr. Viswanatha published in 2023 on the topic of “Prediction of Loan Approval in Banks using Machine Learning Approach” [3]. With the expansion of technology and the banking industry, loan applications have increased. Banks face challenges in assessing these applications and mitigating default risks. This research proposes using machine learning models and ensemble learning to predict loan approval probabilities, improving the selection accuracy of qualified candidates. This method benefits both applicants and bank employees by reducing sanctioning time. Four algorithms - Random Forest, Naive Bayes, Decision Tree, and KNN - were used for prediction, with the Naive Bayes algorithm achieving the highest accuracy of 83.73%.

Last but not least for the related work is from Mr. Rutika Pramod Kathe published in 2021 on the topic of “Prediction Of Loan Approval Using Machine Learning Algorithm” [4]. Banks earn primarily from the interest on loans they credit. Their profit or loss largely depends on loan repayment. Predicting loan defaulters can help reduce Non-Performing Assets, making this a crucial study. Various methods exist to control loan default, with accurate predictions being vital for profit maximisation. This research uses Logistic Regression, a key predictive analytics approach, to predict loan defaulters. Data from Kaggle was used for the study. The performance of the Logistic Regression models was evaluated based on measures like sensitivity and specificity, and the results showed that different models produced different outcomes.

3. Methodology

First step the environment, program and library need to set up such as Visual studio code. It will be used to develop an algorithm and build a desktop application. Few necessary libraries need to be installed such as pandas, numpy, scikit-learn, pickle, Tkinter, and etc. Import the necessary libraries, such as scikit- learn, pandas, and numpy, to process data and create a prediction model. Transform csv file to pandas DataFrame with the salary dataset that exported from kaggle [5]. Second step, the splitting of two subsets from the preprocessed data which is the training set of 80% and a testing set of 20%. The predictive model will be trained using the training set, and its performance will be assessed using the testing set. The third step is selecting a suitable machine learning algorithm which is linear regression from scikit-learn library [6] to predict the salary based on an independent feature year_of_experience. Using the fit() method to adjust the model to the training set of data. In order to produce predictions, the model will discover patterns and relationships in the training data. The fourth step, model evaluation will be measured to know how well and accuracy of trained machine learning is. After seeing the good accuracy of the model, the final step will be developing the desktop application to deploy it. Users can input any integer number for the year of working experience then it will produce the predicting salary result. The figure 5 below shows all details.

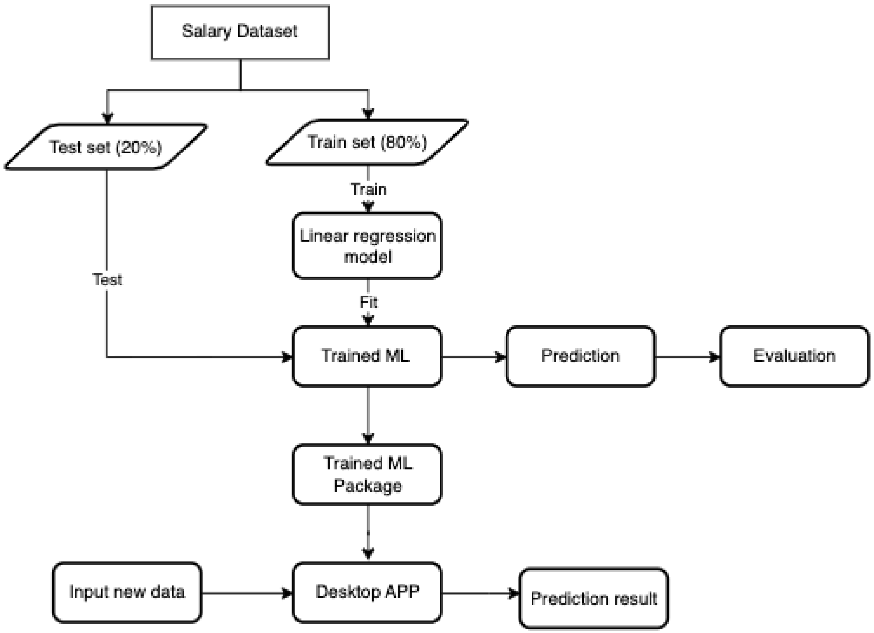
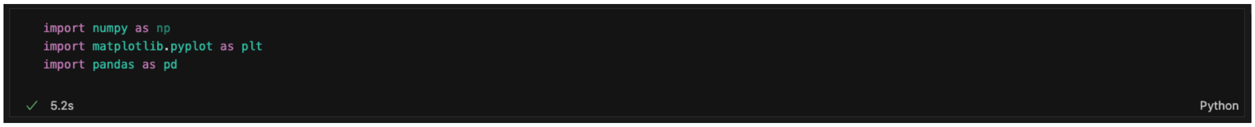


Figure 5: Development machine learning flow

4. Results and Discussion

This will show the result and step of develop model and deploy to desktop application. As mentioned above, necessary libraries need to be imported to use and here how to code it in figure 6.

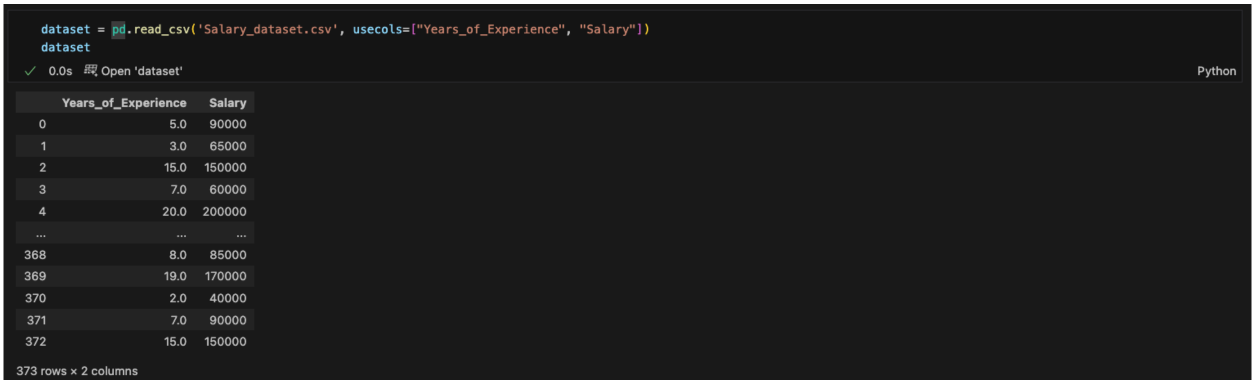


```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

✓ 5.2s Python

Figure 6: Import libraries

After that the step of importing the salary dataset in csv format to dataframe in pandas as shown in figure 7 below. It shows only year_of_esperience and salary features which are selected to build a linear regression model. It has 273 rows of data points.



```
dataset = pd.read_csv('Salary_dataset.csv', usecols=["Years_of_Experience", "Salary"])
dataset
```

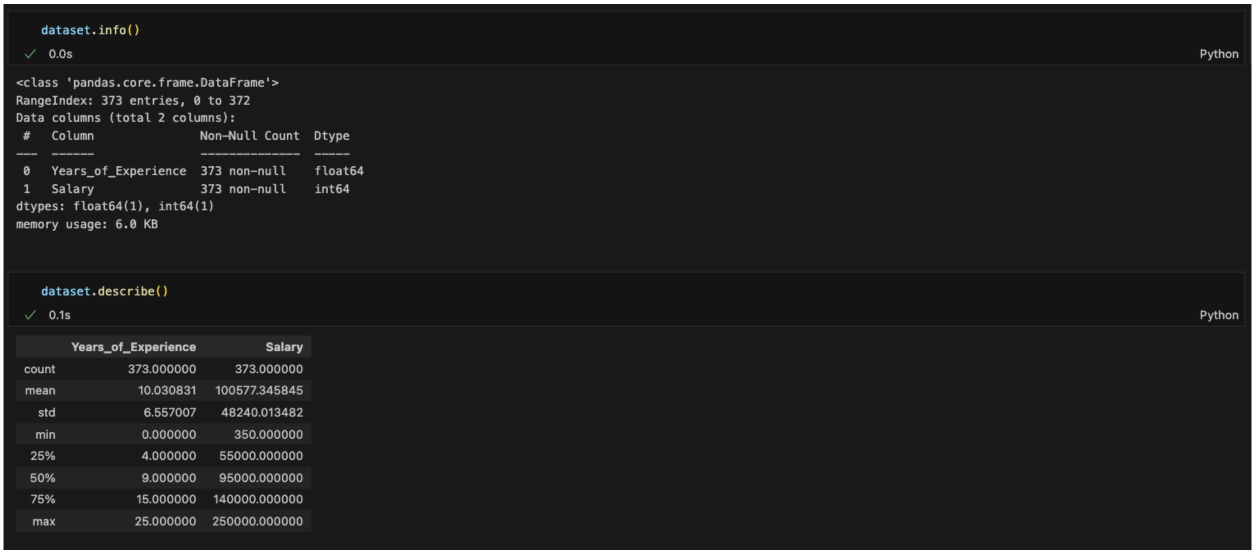
✓ 0.0s Open 'dataset' Python

```
     Years_of_Experience   Salary
0                    5.0    90000
1                    3.0    65000
2                   15.0   150000
3                    7.0    60000
4                   20.0   200000
...                  ...      ...
368                  8.0    85000
369                 19.0   170000
370                  2.0    40000
371                  7.0    90000
372                 15.0   150000

373 rows × 2 columns
```

Figure 7: Load data from csv to pandas dataframe

To measure the quality of loaded data before developing a machine learning model, the EDA (Exploratory Data Analysis) needs to be used. Figure 8 shows the code of doing EDA. Its information shows that an independent variable and a dependent variable are numerical with no missing values so the cleaning process will not be applied. Moreover, it shows the descriptive statistics of the salary dataset as well.



```
dataset.info()
✓ 0.0s                                                    Python
```

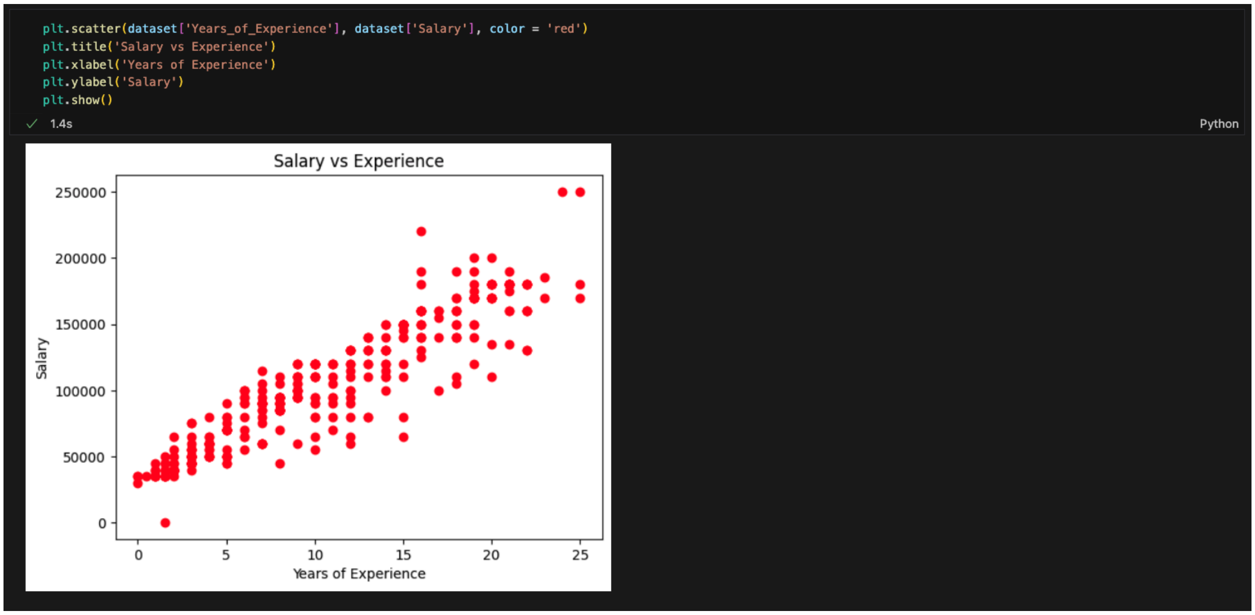
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 373 entries, 0 to 372
Data columns (total 2 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   Years_of_Experience  373 non-null    float64
 1   Salary               373 non-null    int64  
dtypes: float64(1), int64(1)
memory usage: 6.0 KB
```

```
dataset.describe()
✓ 0.1s                                                    Python
```

	Years_of_Experience	Salary
count	373.000000	373.000000
mean	10.030831	100577.345845
std	6.557007	48240.013482
min	0.000000	350.000000
25%	4.000000	55000.000000
50%	9.000000	95000.000000
75%	15.000000	140000.000000
max	25.000000	250000.000000

Figure 8: Exploratory Data Analysis process

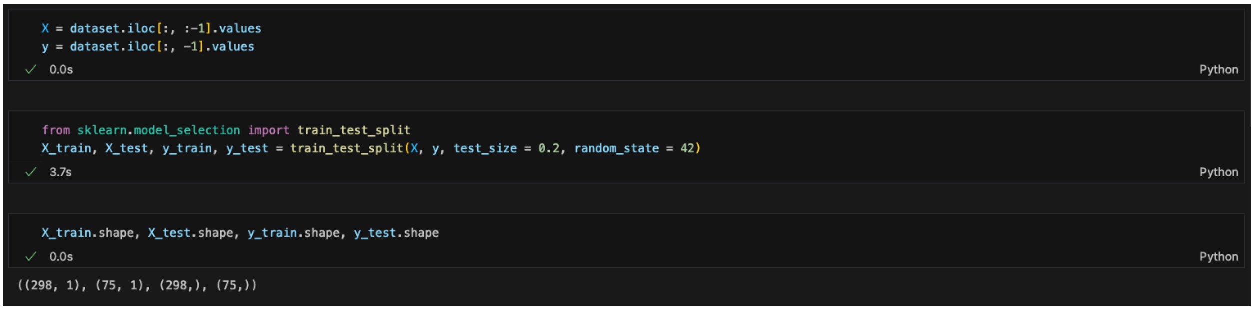
The visualisation is crucial as a picture worth a thousand words and it will show the pattern of data with relationship between features. In figure 9 shows that between independence feature year of experience and dependent feature salary has a linear relationship associated.



```
plt.scatter(dataset['Years_of_Experience'], dataset['Salary'], color = 'red')
plt.title('Salary vs Experience')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
✓ 1.4s                                                    Python
```

Figure 9: Data visualisation

After well known of dataset and it look good for developing machine learning model next step is preparing data for independence feature (year of experience) and dependence feature (salary) separately. Furthermore, training and testing of the dataset apply too. In research work, 80% of data (298 observations) is for training and 20% of data (75 observations) for testing to evaluate the accuracy. In figure 10 below shows the code to implement for these tasks.



```
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```
✓ 0.0s Python

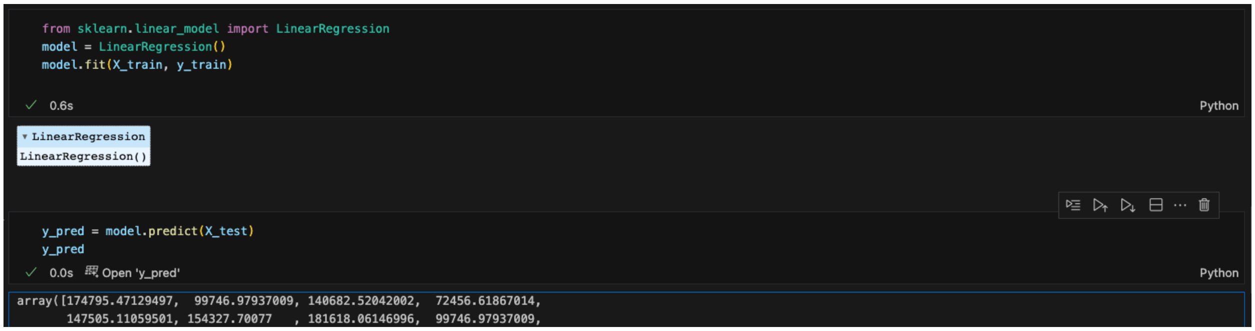
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
```
✓ 3.7s Python

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```
✓ 0.0s Python

```
((298, 1), (75, 1), (298,), (75,))
```

Figure 10: Data preparation for model

This is the time for choosing machine learning from the scikit-learn library, which is a linear regression model. The train dataset is fitted to train the model then predict on test set.



```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```
✓ 0.6s Python

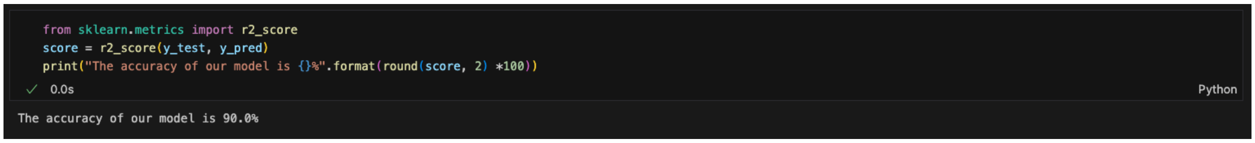
```
▾ LinearRegression
LinearRegression()
```

```
y_pred = model.predict(X_test)
y_pred
```
✓ 0.0s Open 'y_pred' Python

```
array([174795.47129497,  99746.97937009, 140682.52042002,  72456.61867014,
       147505.11059501, 154327.70077   , 181618.06146996,  99746.97937009,
```

Figure 11: Training model for linear regression and prediction

To make sure that the training model is good or bad for using, the evaluation will be applied. In figure 12 below shows the calculation result by using R square measurement. The accuracy came out with 90% which is good for initial predicting.

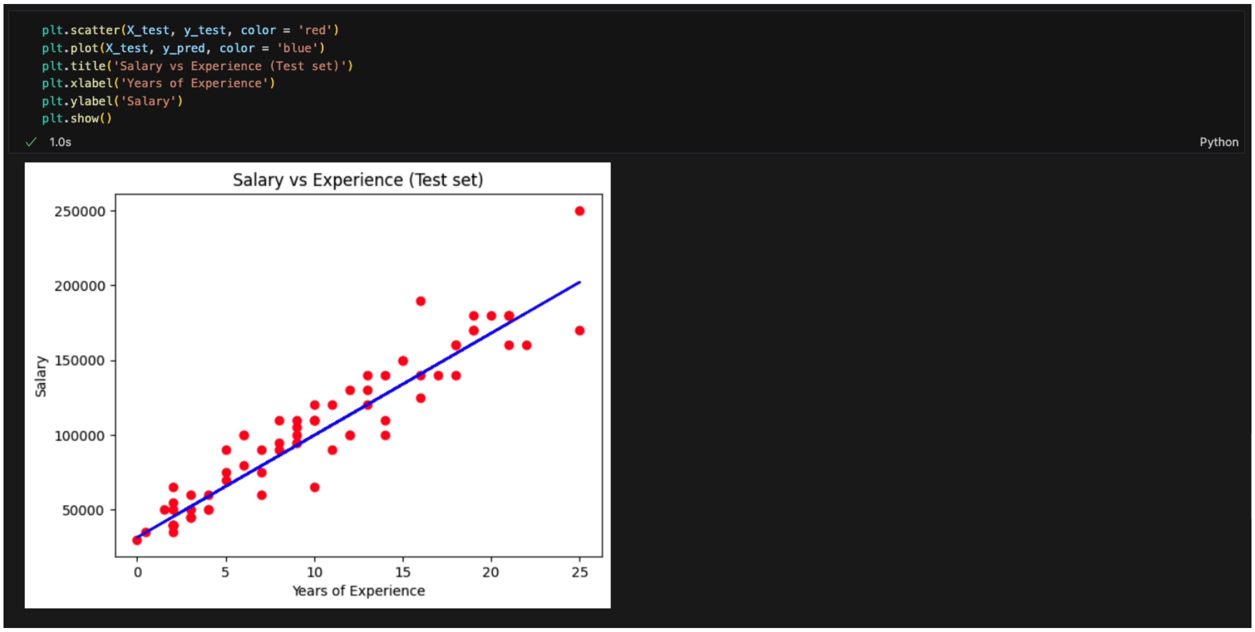


```
from sklearn.metrics import r2_score
score = r2_score(y_test, y_pred)
print("The accuracy of our model is {}%".format(round(score, 2) *100))
```
✓ 0.0s Python

```
The accuracy of our model is 90.0%
```

Figure 12: Model evaluation

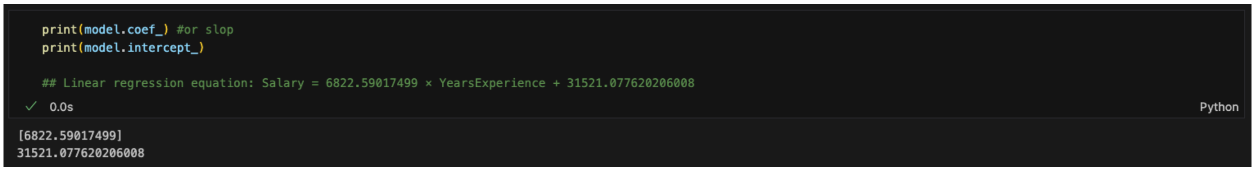
As always mentioned by many expertists, a picture is worth a thousand words so the prediction will be shown in figure 12 how well trained the model is for actual values and predicted value.



```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_test, y_pred, color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

Figure 13: Visualisation of actual and predicted values

In this figure 14 below show how to find slope and intercept for creating the linear regression equation for later use for a trained machine learning model.



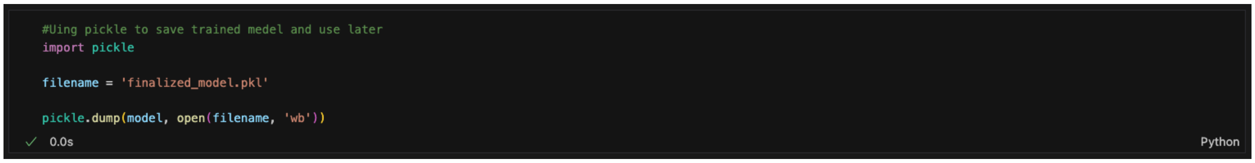
```
print(model.coef_) #or slop
print(model.intercept_)

## Linear regression equation: Salary = 6822.59017499 × YearsExperience + 31521.077620206008
```

```
[6822.59017499]
31521.077620206008
```

Figure 14: Find linear regression equation.

Before deploying the linear regression trained model to desktop application, saving trained model to a package is needed. There are 2 popular techniques which are pickle and joblib. In this research pickle is used to save a model for later use. In figure 15 below show the code how to save a trained model in pickle.



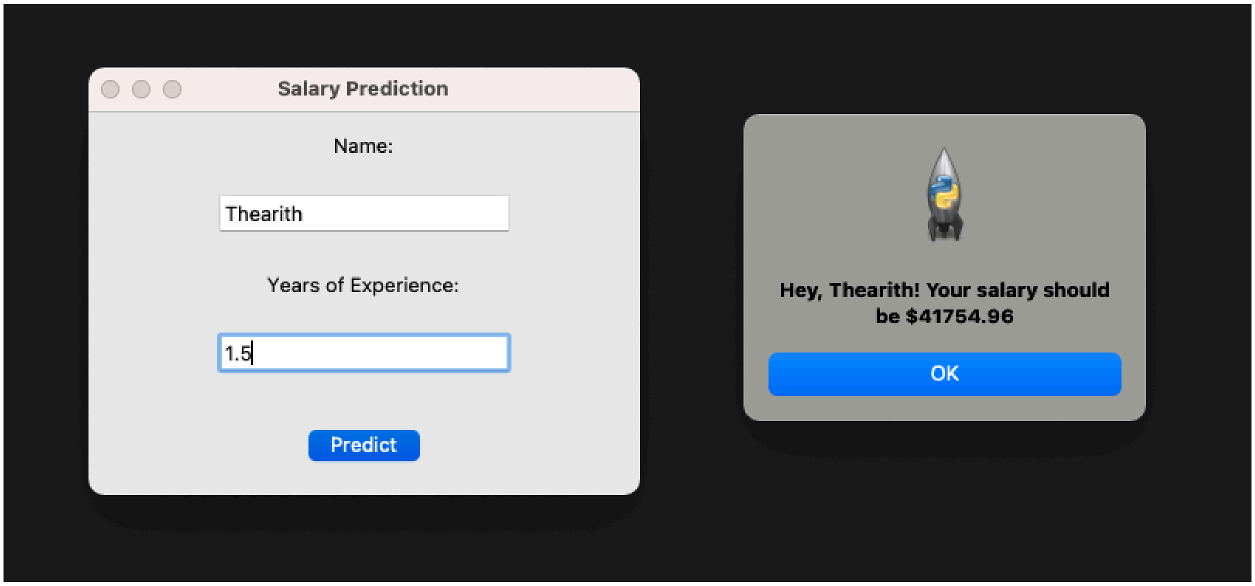
```
#Uing pickle to save trained medel and use later
import pickle

filename = 'finalized_model.pkl'

pickle.dump(model, open(filename, 'wb'))
```

Figure 15: Save Trained model in pickle

The final step is to deploy a trained machine learning model into a desktop application as shown in figure 16.



Salary Prediction

Name:

Thearith

Years of Experience:

1.5

Hey, Thearith! Your salary should be $41754.96

Figure 16: Prediction result on desktop application

5. Conclusions

In this research paper, a machine learning model for linear regression and its application for salary prediction is shown to how to develop the real world application by using python programming language, scikit-learn, tkinter and other useful libraries. This initial research by using linear regression for salary prediction shows the good accuracy with 90%. The purpose of this work is to encourage students, researchers and scientists to interest in machine learning and use it to solve many real world problems. The simple desktop application is builded to deploy that trained model so users can input any value of year of experience then predict the salary easily. The limitation of this work is just using linear regression algorithms to build the prediction model. In opportunity to use other machine learning techniques for the salary prediction such as decision tree, random forest. The new data for specific domains is good too for prediction. The last opportunity is to deploy this training to iOS and Android mobile is considered as well. Hopefully, this research is valuable for students, researchers and scientists for future research and real development.

References

[1] Linear regression equation, https://www.wikihow.com/Calculate-Slope-and-Intercepts-of-a-Line.

[2] Mr. Sahin Magar (2022), “Loan Eligibility Prediction using Machine Learning Algorithms”.

[3] Mr. Viswanatha (2023), “Prediction of Loan Approval in Banks using Machine Learning Approach”.

[4] Mr. Rutika Pramod Kathe (2021), “Prediction Of Loan Approval Using Machine Learning Algorithm”.

[5] Kaggle dataset, https://www.kaggle.com/.

[6] Scikit-learn library, https://scikit-learn.org/stable/index.html.