Regression : Software Developer Salary Prediction Using Sklearn Python

Michael Leonardo, Agustinus Ardian Cakra Widiastara, Gabriel Jehuda

Computer Science Department, School of Computer Science, Bina Nusantara University
Jakarta, Indonesia 11480

***Abstrak*—Software Developer merupakan salah satu profesi yang sedang hangat dibicarakan di era perkembangan teknologi yang sangat pesat ini. Proyek ini bertujuan untuk membuat sebuah model Machine Learning yang memiliki tujuan utama untuk memprediksi gaji seorang Software Developer berdasarkan variabel-variabel tertentu. Variabel yang digunakan pada proyek ini adalah lokasi pekerjaan, status pendidikan, serta jumlah tahun pengalaman kerja. Model yang digunakan untuk memprediksi gaji adalah Decision Tree Regression Model.**

***Kata kunci*—Regression, Software Developer Salary, Machine Learning**

I. Introduction

Seiring dengan berkembangnya teknologi, semakin banyak pula profesi-profesi baru yang lahir. Tidak sedikit pula dari profesi-profesi baru tersebut memiliki angka bayaran yang cukup tinggi. Software Developer merupakan salah satu profesi yang sedang marak dibicarakan di dunia teknologi ini. Semakin banyak individu yang ingin terlibat dalam industri dunia teknologi, semakin besar pula lapangan pekerjaan yang diciptakan oleh tren tersebut. Banyak pelaku industri teknologi membutuhkan individu-individu yang mampu bekerja sebagai Software Developer sehingga profesi tersebut menjadi pembicaraan yang hangat di kalangan komunitas programmer. Belum lagi bayaran yang ditawarkan relatif cukup besar dibanding dengan profesi-profesi lainnya.

Namun seperti yang diketahui, gaji seseorang tidak hanya ditentukan oleh profesi yang dimilikinya, namun juga terdapat banyak faktor lain yang mempengaruhi besaran gaji seseorang. Diantaranya adalah lokasi pekerjaan, pengalaman kerja, serta status pendidikan menjadi parameter yang menentukan jumlah gaji seseorang. Maka, proyek ini bertujuan untuk membuat sebuah model prediksi gaji seorang Software Developer, menggunakan beberapa variabel-variabel penentu.

II. Methodology

A. Memilih Tools

i. Python

Python adalah bahasa pemrograman interpretatif yang dapat digunakan di berbagai platform dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode dan merupakan salah satu bahasa populer yang berkaitan dengan Data Science, Machine Learning, dan Internet of Things (IoT).[1]

ii. Scikit-Learn

Scikit-Learn atau sklearn adalah modul untuk bahasa pemrograman python yang dibangun diatas NumPy, SciPy, dan matplotlib, fungsinya dapat membantu melakukan processing data ataupun melakukan training data untuk kebutuhan machine-learning.[2]

iii. Streamlit

Streamlit merupakan framework berbasis Python yang bersifat open source dan dibuat lebih interaktif khususnya dalam membangun sebuah aplikasi website di bidang data science.[3]

iv. Matplotlib

Matplotlib adalah suatu library atau package yang paling populer di bahasa python untuk melakukan visualisasi data

seperti membuat plot grafik untuk satu sumbu atau lebih.[4]

v. Pandas

Pandas adalah sebuah library di Python yang berlisensi BSD dan open source yang menyediakan struktur data dan analisis data yang mudah digunakan. Pandas biasa digunakan untuk membuat tabel, mengubah dimensi data, mengecek data, dan lain sebagainya.[5]

vi. Numpy

Numpy merupakan salah satu library pada Python yang berfungsi melakukan proses komputasi numerik.[6]

vii. Decision Tree Regressor

Regression Tree merupakan model menghasilkan data kontinu atau masih berhubungan. Hasil dari regresi model ini bisa dianggap sebagai bilangan nyata.[7]

viii. Linear Regression

Linear Regression (Regresi Linear) adalah suatu regresi linear yang digunakan untuk mengestimasi atau memprediksi hubungan antara dua variabel dalam penelitian kuantitatif. Dimana regresi linear ini mampu membuat satu asumsi tambahan yang mengkorelasikan antara variabel independen dan dependen melalui garis yang paling sesuai dari titik data garis lurus.[8]

B. Memilih Dataset

Dataset yang digunakan harus memiliki beberapa variabel pilihan kami agar dapat menciptakan model prediksi gaji yang cukup baik. Variabel-variabel pilihan kami diantaranya adalah lokasi pekerjaan, status pendidikan, serta jumlah tahun pengalaman kerja.

Dataset yang ditawarkan oleh survey website Stack Overflow menawarkan banyak variabel yang dapat digunakan untuk memodelkan model prediksi gaji Software Developer. Maka, proyek ini menggunakan dataset "Stack Overflow Annual Developer Survey" (https://info.stackoverflowsolutions.com/rs/719-EMH-566/images/stack-overflow-developer-survey-2021.zip)[9].

C. Data Cleaning

Dataset yang diperoleh tidak dapat secara langsung digunakan untuk memodelkan model prediksi gaji tersebut. Diperlukan beberapa tahap pre-processing yang harus dilakukan agar dataset dapat membuahkan model prediksi yang optimal dan mendekati akurat.

Yang pertama dilakukan adalah melihat secara umum bagaimana bentuk dataset yang akan digunakan untuk proyek pembuatan model prediksi gaji. Hal tersebut dapat dilakukan dengan melihat beberapa baris pertama dari dataset serta memunculkan semua kolom yang terdapat di dataset.

```
df.head()
```

Potongan kode diatas dapat memunculkan lima baris pertama dari dataset lengkap dengan semua kolom yang terdapat di dataset. Setelah diteliti, dataset memiliki beberapa kolom yaitu:

1. *ResponseId*
2. *MainBranch*
3. *Employment*
4. *Country*
5. *US_State*
6. *UK_Country*
7. *EdLevel*
8. *Age1stCode*
9. *LearnCode*
10. *YearsCode*
11. *Age*
12. *Gender*

13. *Trans*

14. Dan lain-lain

Dari banyaknya variabel di atas, sesuai dengan variabel pilihan yang sebelumnya sudah ditentukan maka kolom-kolom yang akan digunakan untuk memodelkan model prediksi gaji adalah *Country, EdLevel, Employment, YearsCodePro*, dan *ConvertedCompYearly* sebagai jumlah gaji developer per tahun. Cuplikan kode di bawah digunakan untuk mengambil variabel-variabel tersebut serta mengubah nama variabel *ConvertedCompYearly* menjadi *Salary*,

```
df =
df[["Country","EdLevel","YearsCodePro","Employment"
,"ConvertedCompYearly"]]

df = df.rename({"ConvertedCompYearly": "Salary"},
axis=1)
```

Setelah mendapatkan variabel-variabel yang diinginkan, maka tahap berikutnya adalah melakukan *Data Cleaning* agar dataset menjadi optimal.

i. *Handling Null Values*

Pada proyek ini nilai kosong atau *NULL values* akan dihandle dengan cara di *drop* atau dihilangkan barisnya dari dataset, berikut merupakan cuplikan kode untuk melakukan hal tersebut,

```
df = df[df["Salary"].notnull()]

df = df.dropna()
```

ii. *Specifying the dataset*

Data yang kami gunakan ingin lebih kami spesifik dari entry-entry yang didapatkan, hal tersebut bisa didapatkan dengan memotong dataset berdasarkan variabel *Employment*. Pada variabel ini hanya akan diambil entry-entry yang merupakan *Full-time Employment* sehingga cuplikan kode di bawah dapat membantu memotong dataset menjadi lebih spesifik,

```
df=df[df["Employment"]=="Employed full-time"]
df=df.drop("Employment", axis=1)
```

Disini variabel *Employment* di-*drop* karena tidak ingin digunakan pada model prediksi gaji.

ii. *Handling Variabel Country*

Data country pada projek ini diubah dengan kriteria yaitu entry dari country yang jumlahnya kurang dari 500 akan di *drop*. entry dari country yang jumlahnya kurang dari 500 di drop karena jumlah entry kurang banyak dan dapat mengakibatkan *high variance* dan *high error* pada test set.

```
def CutCountry(categories, minimum):
    NewCountry = {}
    for i in range(len(categories)):
        if categories.values[i] >= minimum:
            NewCountry[categories.index[i]] = categories.index[i]
        else:
            NewCountry[categories.index[i]] = 'Other'
    return NewCountry
```

```
CuttedCountry = CutCountry(df.Country.value_counts(),500)
df['Country'] = df['Country'].map(CuttedCountry)
```

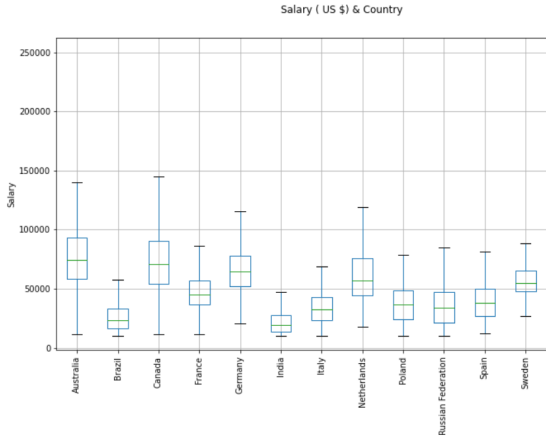
Setelah menggunakan cuplikan kode di atas didapatkan variabel *Country* yang lebih optimal,

Other	12448
United States of America	9175
India	3385
Germany	2753
United Kingdom of Great Britain and Northern Ireland	2604
Canada	1553
France	1396
Brazil	1340
Spain	890
Netherlands	835
Australia	825
Poland	775
Italy	756
Russian Federation	741
Sweden	735

iii. *Handling Outlier*

Data outlier terdapat pada kolom *Salary*, dan *YearsCodePro*. Untuk data *Salary*, dilakukan boxplot salary dengan

Country, lalu outlier di drop secara manual. Sedangkan untuk data *YearsCodePro* langsung dilakukan boxplot dan outlier di drop secara manual. Handling outlier ini digunakan untuk mengambil hanya nilai-nilai median saja dari dataset yang akan digunakan dengan tujuan akan mengoptimalkan model regresi. Dapat dilihat dari *Boxplot* yang didapatkan setelah melakukan handling pada data-data *outlier*,



iv. *Handling YearsCodePro*

Untuk data pada *YearsCodePro* terdapat value string "Less than 1 year" dan "More than 50 years". Kedua data tersebut diubah menjadi 0.5 dan 51. Berikut merupakan cuplikan kodenya

```
def StF(x):
    if x == "More than 50 years":
        return 51
    if x == "Less than 1 year":
        return 0.5
    return float(x)

df["YearsCodePro"] = df["YearsCodePro"].apply(StF)
```

v. *Handling EdLevel*

Data pada EdLevel dihandle dengan cara melakukan generalisasi menjadi 4 kategori yaitu "Bachelor's degree", "Master's degree", "Professional or Doctoral degree" dan "Less than a Bachelor". Berikut merupakan cuplikan kode yang dapat diimplementasikan untuk menghandle variabel *EdLevel*,

```
def CombineEdLevel(x):
    if "Bachelor" in x:
        return "Bachelor's degree"
    if "Master" in x:
        return "Master's degree"
    if "Professional" in x or "Other doctoral" in x:
        return "Professional or Doctoral Degree"
    return "Less than a Bachelor"

df["EdLevel"] = df["EdLevel"].apply(CombineEdLevel)
```

vi. *Handling Categorical Data*

Data Kategorikal pada dataset ada pada kolom EdLevel dan Country. Data ini dihandle dengan cara menggunakan Label Encoder dari library sklearn. Berikut merupakan cuplikan kode yang dapat membantu melakukan Label Encoding pada variabel-variabel tersebut,

```
from sklearn.preprocessing import LabelEncoder
EdLevel_LE = LabelEncoder()
df["EdLevel"] = EdLevel_LE.fit_transform(df.EdLevel.values)
df["EdLevel"].unique()
```

```
Country_LE = LabelEncoder()
df["Country"] = Country_LE.fit_transform(df.Country.values)
df["Country"].unique()
```

D. Modelling & Prediksi

Setelah melakukan *Data Cleaning,* data dipisah menjadi x dan y dimana x merupakan independent variabel dan y dependent variabel. Data yang digunakan sebagai independent variabel adalah *Country*, *EdLevel* dan *YearsCodePro* sedangkan data yang digunakan sebagai dependent variabel adalah *Salary.* Setelah Independent dan dependent variabel dipisah, data akan dibagi menjadi 2 yaitu training set dan test set dengan menggunakan function train_test_split dari library sklearn.

Dalam projek ini, terdapat 2 model yang digunakan. *Linear Regression* model dan *Decision Tree Regression* model. Kedua model tersebut menggunakan library sklearn. Untuk *Decision Tree*

Regression model dibantu oleh *GridSearchCV* untuk mencari parameter yang paling optimal.

Setelah kedua model dibuat, akan dilakukan training dengan cara fitting pada training set. Setelah training, kedua model digunakan untuk memprediksi data pada test set.

III. Hasil dan diskusi

Untuk mengecek error pada kedua model tersebut, projek ini menggunakan metode RMSE (*Root Mean Squared Error*) untuk menghitung error.

Setelah melakukan prediksi pada test set, model linear regression memiliki error sebesar 39425.09 sedangkan decision tree regression model memiliki error dengan perhitungan Mean Squared Error sebesar 28833.56. Dari kedua model tersebut, model yang lebih baik adalah *Decision Tree regression*.

Maka model *Decision Tree Regression* tersebut yang dijadikan model prediksi untuk aplikasi berbasis website yang di publish melalui Heroku.

IV. Konklusi

Model *Decision Tree Regression* menggunakan variabel *Country*, *EdLevel* dan *YearsCodePro* untuk menghitung prediksi gaji seorang *Software Developer* memiliki akurasi model yang terbilang cukup baik dengan error sebesar 28833.56 pada prediksi gaji.

References

1. Indonesia, D. (n.d.). Developer Academy: Memulai Pemrograman Dengan Python. Dicoding. Retrieved June 7, 2022, from https://www.dicoding.com/academies/86
2. Andhika, W. (2021, December 10). Belajar machine-learning, basic of scikit-learn - Wahyu Andhika. Medium. Retrieved June 7, 2022, from https://medium.com/@wahyuandhika/belajar-machine-learning-basic-of-scikit-learn-a1685db819a8
3. A. (2022, March 14). Tutorial Bangun Portofolio Data Menawan dengan Python Streamlit. DQLab. Retrieved June 7, 2022, from https://dqlab.id/tutorial-bangun-portofolio-data-menawan-dengan-python-streamlit
4. A. (2021, August 30). Mengenal Matplotlib untuk Visualisasi Data dengan Python. DQLab. Retrieved June 7, 2022, from https://www.dqlab.id/mengenal-matplotlib-untuk-visualisasi-data-dengan-python
5. W. (2021b, July 8). Belajar Python Mengenal Pandas dan Series untuk Meningkatkan Kompetensi Data. DQLab. Retrieved June 7, 2022, from https://www.dqlab.id/belajar-python-mengenal-pandas-dan-series-untuk-meningkatkan-kompetensi-data
6. Y. (2020, December 7). Belajar Numpy Array Python, Fungsi Yang Populer Dalam Proses Manipulasi Data. DQLab. Retrieved June 7, 2022, from https://www.dqlab.id/belajar-numpy-array-python-bersama-dqlab
7. B. (2022b, February 18). Ambil Keputusan Efektif dengan Decision Tree. Algoritma. Retrieved June 7, 2022, from https://algorit.ma/blog/decision-tree-adalah-2022/
8. B. (2022c, February 28). Apa Itu Linear Regression dalam Machine Learning? Caraguna. Retrieved June 7, 2022, from https://caraguna.com/apa-itu-linear-regression-dalam-machine-learning/
9. *Stack Overflow*. (n.d.-a). Stack Overflow. Retrieved June 7, 2022, from https://insights.stackoverflow.com/survey