

Resume Analysis

Presented By:
KAVIYA K K



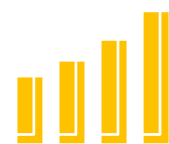
OVERVIEW



INTRODUCTION



PROBLEM STATEMENT



METHODOLOGY



**PREPROCESSING
USING NLP**



**VALIDATING MODEL
USING ML**



FLASK CONNECTIVITY



KEY FINDINGS



CONCLUSION



ROLE IN BUSINESS





INTRODUCTION

Resume Analysis is the primary step in the hiring process. It evaluates the candidates resumes and determines whether they are qualified for a role based on their education, skill sets, technical stuff, experience, and other information captured in their resume. Multiple ways were introduced to cover all these cons to performing this resume analysis process. Various technologies, including Natural Language processing and Machine Learning, were involved in searching for the solutions.





PROBLEM STATEMENT

Today the major problem being faced across the industry is how to acquire the right talent, using minimal resources over the internet and in minimal time. It intends to provide a solution to separating the right candidates from the pack and making sense of knowing that candidates can do the job before you hire them.



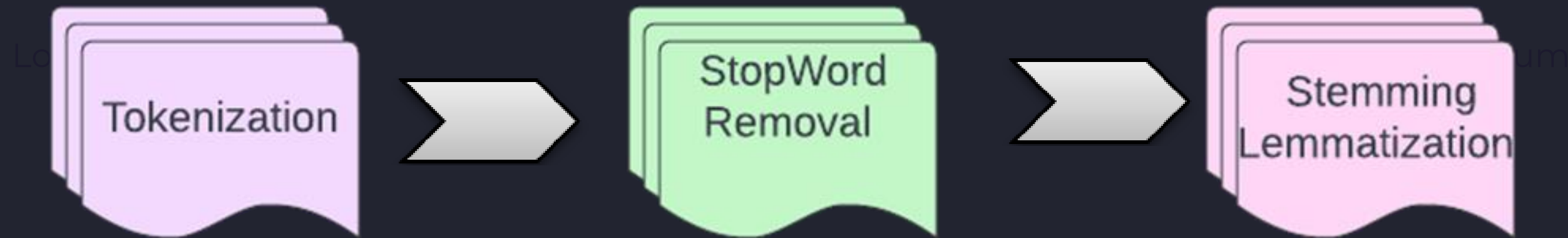
METHODOLOGY

**Preprocessing Using
NLP**

**Validating Model Using
Machine Learning**

Flask Connectivity

PREPROCESSING USING NLP



Tokenization is breaking the raw text into small chunks. Tokenization breaks the raw text into words, sentences called tokens. These tokens help in understanding the context or developing the model for the NLP. The tokenization helps in interpreting the meaning of the text by analyzing the sequence of the words

Stopwords are the words in any language which does not add much meaning to a sentence. They can safely be ignored without sacrificing the meaning of the sentence

Stemming and Lemmatization both generate the root form of the inflected words. The difference is that stem might not be an actual word whereas, lemma is an actual language word



**VALIDATING MODEL USING
MACHINE LEARNING**



FLASK CONNECTIVITY

Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Pooeco. Flask is based on the Werkzeug WSGI toolkit and the Jinja2 template engine. Both are Pooeco projects.

I have connected my backend by using machine learning algorithm model K-Nearest Neighbor Classifier with frontend Web pages by using FLASK web python framework. So, we can directly upload the link of the candidate resume and get to know whether the resume is selected or rejected.

KEY FINDING

FITTING DATA INTO PIPELINE

```
In [45]: modelPipeline(x_train, x_test, y_train, y_test)
```

```
Out[45]: ([Pipeline(steps=[('scaler', StandardScaler()),  
                             ('classifier', LogisticRegression())]),  
          Pipeline(steps=[('scaler', StandardScaler()), ('classifier', BernoulliNB())]),  
          Pipeline(steps=[('scaler', StandardScaler()),  
                             ('classifier', KNeighborsClassifier())]),  
          Pipeline(steps=[('scaler', StandardScaler()), ('classifier', SVC())]),  
          Pipeline(steps=[('scaler', StandardScaler()),  
                             ('classifier', DecisionTreeClassifier())]),  
          Pipeline(steps=[('scaler', StandardScaler()),  
                             ('classifier', ExtraTreesClassifier())]),  
          Pipeline(steps=[('scaler', StandardScaler()),  
                             ('classifier', RandomForestClassifier())]),  
          Model F1_Score Precision Recall Accuracy ROC_AUC  
0 Logistic Regression 0.857143 0.75 1.000000 0.875 0.900000  
1 Naive Bayes 0.800000 1.00 0.666667 0.875 0.833333  
2 K-Nearest Neighbors 1.000000 1.00 1.000000 1.000 1.000000  
3 SVM 0.857143 0.75 1.000000 0.875 0.900000  
4 Decision Tree 1.000000 1.00 1.000000 1.000 1.000000  
5 Extra Trees 1.000000 1.00 1.000000 1.000 1.000000  
6 Random Forest 1.000000 1.00 1.000000 1.000 1.000000)
```

By defining function method calling number of machine learning algorithms to fitting the model by using machine learning pipeline method.



KEY FINDING

USING K-Nearest Neighbor Classifier

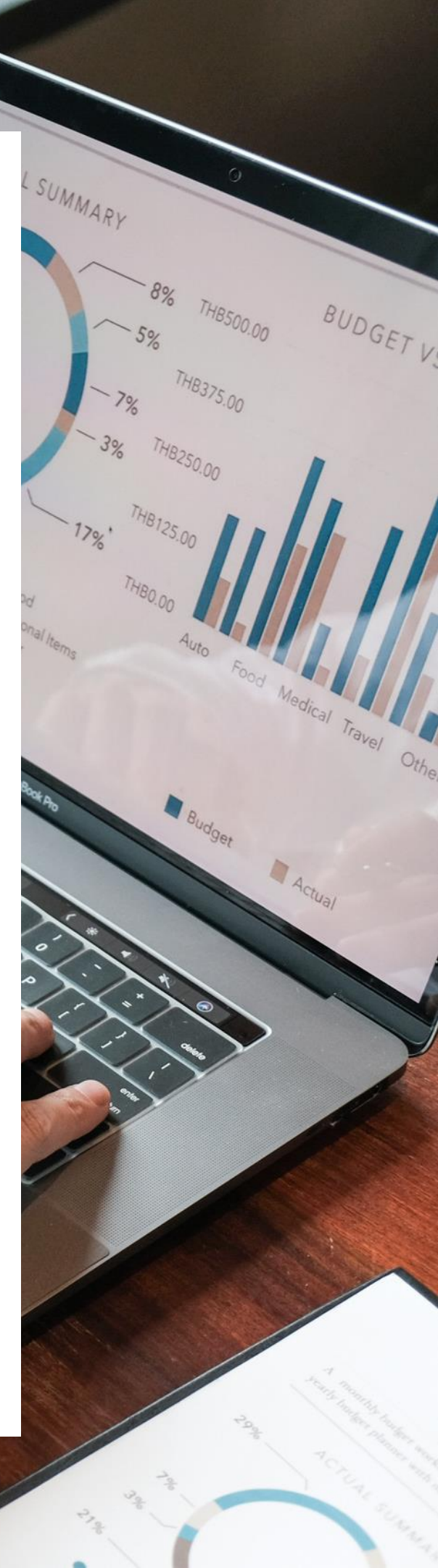
```
5]: model=KNeighborsClassifier().fit(x_train,y_train)
```

```
7]: y_pred=model.predict(x_test)
```

```
ac= skm.accuracy_score(y_test,y_pred)  
print('Accuracy score : ',ac)
```

```
Accuracy score : 1.0
```

By training various machine learning algorithms accuracy are predicted and models are trained well. While comparing to other algorithms K Nearest Neighbors Classifier algorithm model gave best accuracy, finally K nearest neighbor classifier model is best suited for training the model



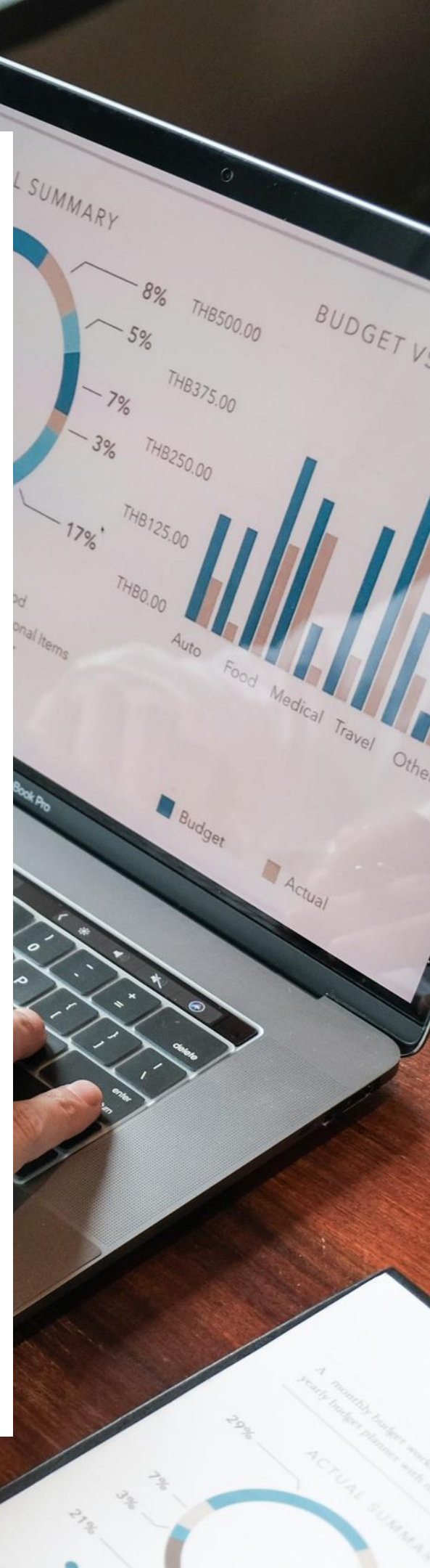
KEY FINDING

```
In [29]: df.head()
```

```
Out[29]:
```

	resume_id	resume_data	candidate_name	validation_percentage	status
0	0	Area\n\nof\n\nSpecialization\n\n\n>\n\nDBM...	Aarthi	35	selected
1	1	\n\nAddress:512\n\nAnandhavar Nagar, Trichy ...	Abishekekumar A	13	not_selected
2	2	\n\nCAPSTONE\n\nPROJECT\n\n\nProject Title...	ANITHA S	31	selected
3	3	\n\nProject Title: \nBlack Friday Dataset\n...	Charumathi C	11	not_selected
4	4	\n\n\n\nCOURSE COMPLETION\n\n\n\n\n...	Deepika	32	selected

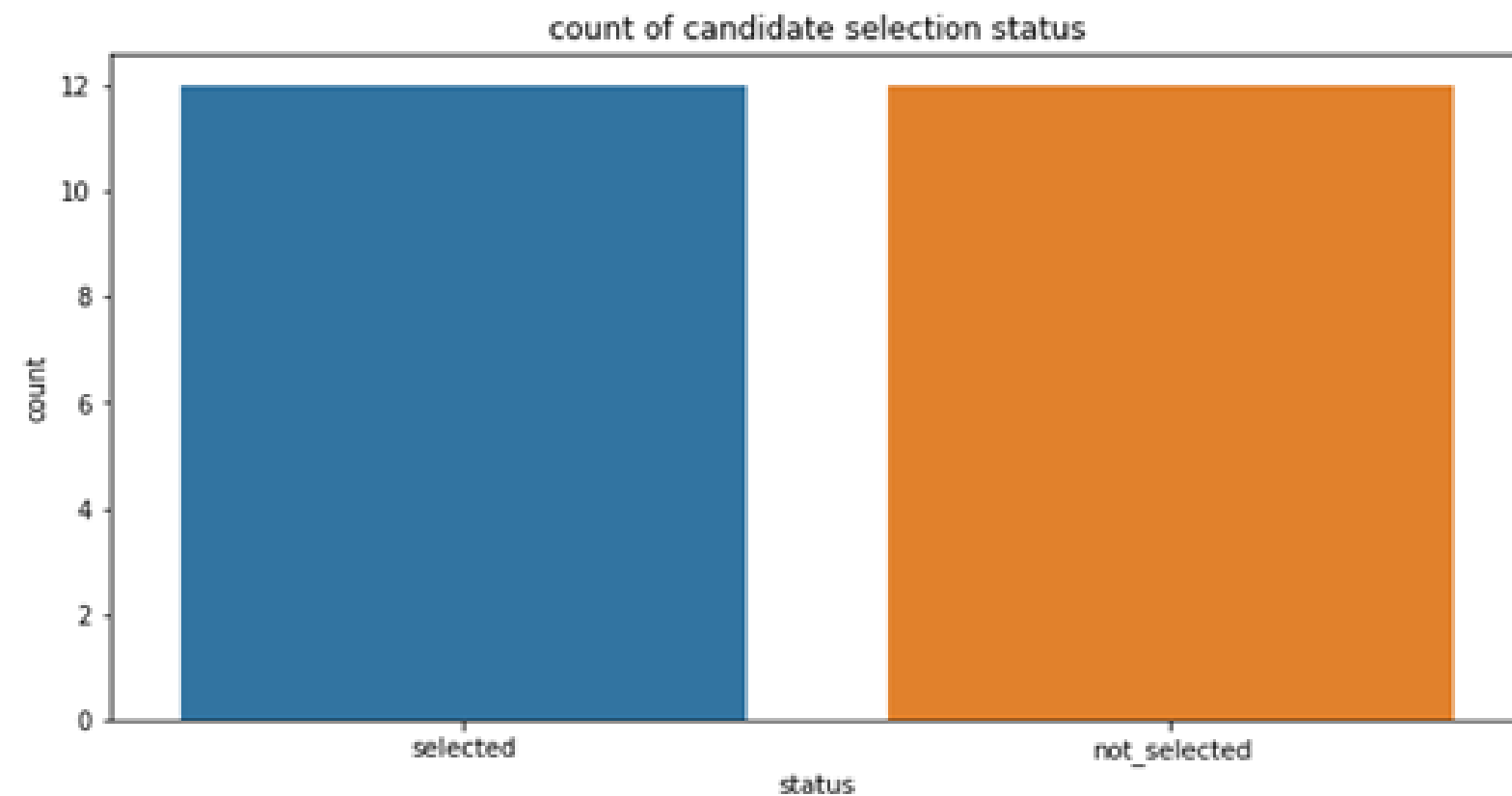
I have validating the resume dataset and skills dataset, by validating I have found the prediction that how each resume has matched with skills and predict the values and store it in data frame and by validating I have made one condition that is if the resume and skills are matched and make prediction of value above '25' means that will be consider as value 'selected', or else the value will be consider as 'not selected'.



KEY FINDING

```
In[8]: plt.figure(figsize=(10, 5))
sns.countplot(df['status'])
plt.title('count of candidate selection status')
```

```
In[8]: Text(0.5, 1.0, 'count of candidate selection status')
```

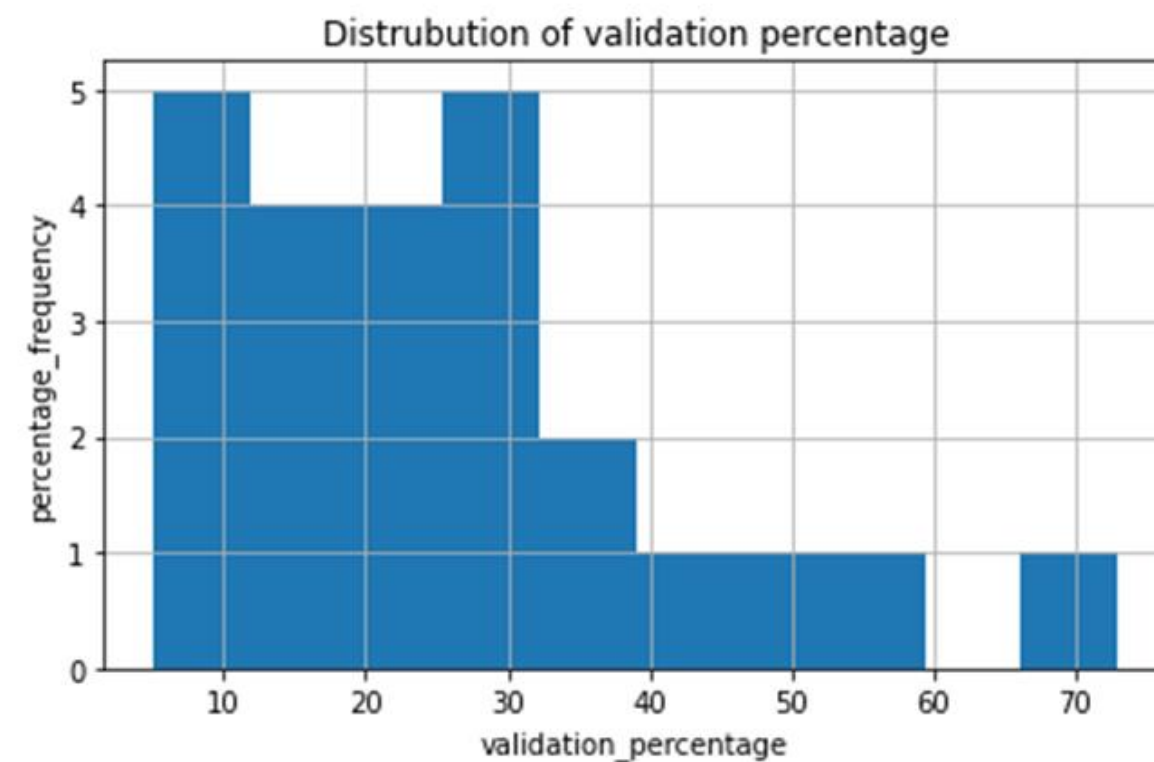


This figure shows the count of candidate selection status based on the company's required skills and candidate resumes.

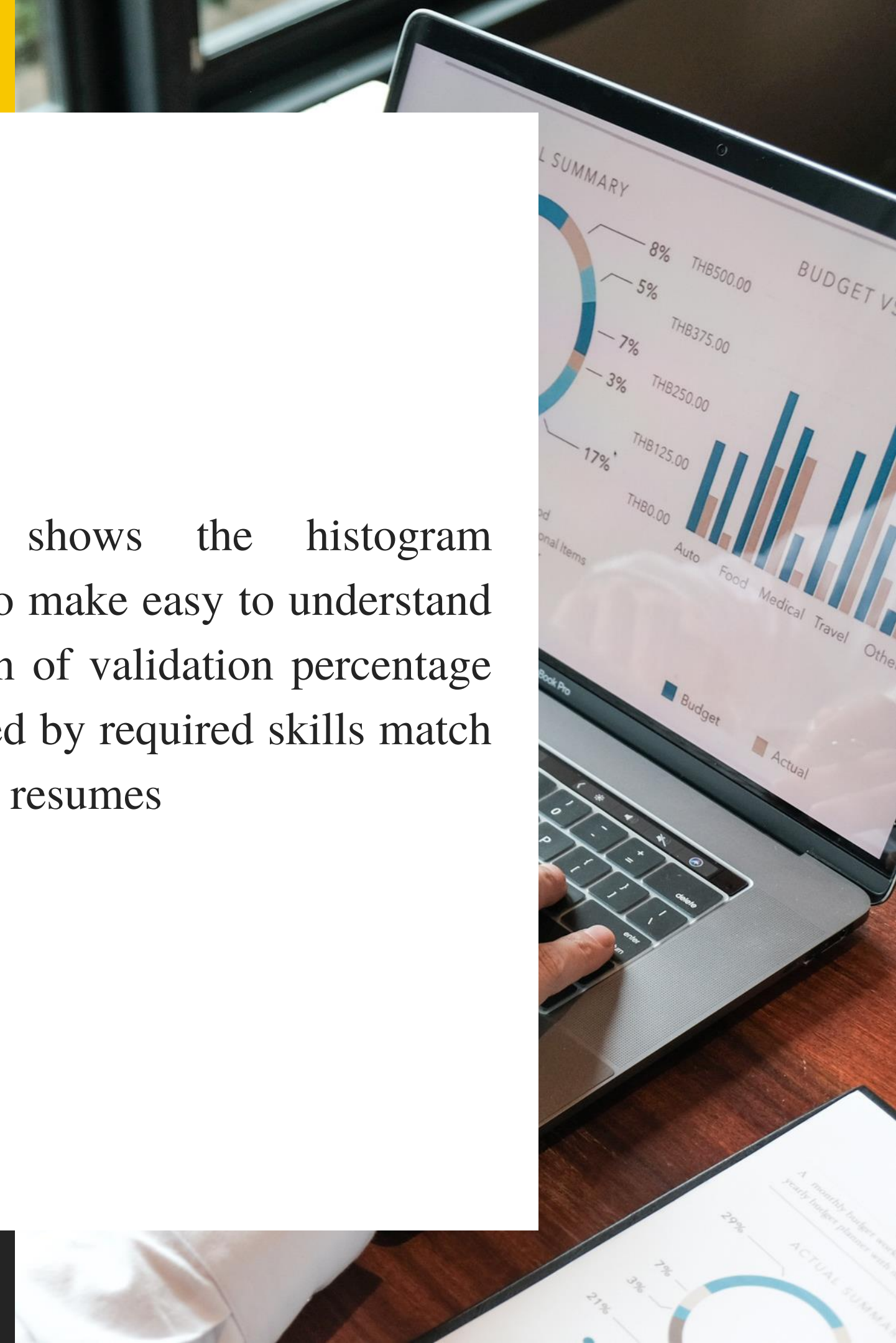
KEY FINDING

[39]:

```
df["validation_percentage"].hist()  
plt.xlabel("validation_percentage")  
plt.ylabel("percentage_frequency")  
plt.title("Distribution of validation percentage")  
plt.tight_layout()  
plt.show()
```



This figure shows the histogram visualization to make easy to understand the distribution of validation percentage score calculated by required skills match with candidate resumes



ROLE IN BUSSINESS

It would help to find the right resume from the large dumps of resumes; would be agnostic to the format in which resume has been created and would give with the list of resumes which are the best match to the job description provided by the recruiter.





CONCLUSION

I have evaluated the concepts of classification is grasped by resume analyses, and classification models have been built by using numerous techniques. This approach will assist business and save time throughout the recruitment process. From the result and discussion, here I conclude that various machine learning algorithm has given different accuracy score percentage whereas K-Nearest Neighbor Classifiers model is best suited model have given the high accuracy score compare to other machine learning algorithms.



Thank You