iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71624

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

High-Scale ATS and Semantic Resume Filtering using Django, NLP, and LLaMA3-GroqModels

Nita Chaudhary[1], Priyanka Sharma[2]
[1]PGScholar,[2]Assistant Professor, Department of Computer Engineering, Bhagwan Mahavir College of Engineering & Technology, India

Abstract: An intelligent system that uses Natural Language Processing (NLP) and Machine Learning (ML) to automate resume classification is presented in this paper. Key resume features, such as education, skills, and job titles, are extracted and used to train models like Logistic Regression, SVM, Random Forest, and BERT. NLP preprocessing techniques, such as tokenization, stop word removal, and vectorization, prepare the text for analysis. The results of experiments show that these models improve classification accuracy and decrease the time needed for hiring. The system assists recruiters by ranking qualified candidates and eliminating applications that are not relevant, making the hiring process quicker and more efficient.

Keywords: Human Resource,GoalOriented,data analysis, Natural LanguageProcessing(NLP), metadataextraction,resumes,CV, large language models

I. INTRODUCTION

For everyjobopportunityintoday's fast-paced, technologicallyadvancedlabor market, companies receive an excessive volume of resumes.Notonly is it inefficient and time-consuming to manually review every application,but it is also prone to human biases and inconsistencies. There is now moreinterest in adopting intelligent technologyto automate the employmentprocess asaresult ofthisdeveloping difficulty. TheuseofArtificialNeural Networks inresumeclassification hasshown promisingresul ts with a 94% accuracy in skill prediction.[6]

One promising approach is resume classification, which automatically groups resume into predetermined categories or employment positions. Automated resume evaluation tools are becoming increasingly important in digital hiring ecosystems. Career Mapper isaweb-based system thatanalyzesover 1.6million LinkedIn profilestoevaluateandproviderecommendations for resume improvements. It suggests changes based on frequencyand patterns found in professionallysuccessful resumes (Lai etal.,2016).Thismethodincreasestheobjectivityofresumereviewsandhelpscandidatesbetteraligntheirprofileswithindustry expectations.[7]Thesesystems mayevaluateandunderstandunstructuredresumedatatofindpertinentcredentials,abilities, and experience by utilizing Natural Language Processing (NLP) and Machine Learning (ML) approaches.

A machine learning-based resume screening model based on KNN and cosine similarity was proposed in 2021 to achieve high candidate-job matching accuracy. To support their methodology, they also made reference to earlier NLP-based vector space techniques.[8] A ML-based resume screening system that uses NLP, KNN/SVM, and cosine similarity to evaluate job-role compatibilityandmakeimprovementrecommendations.ByintegratingGitHubandLinkedIndataintoitspredictionpipeline,the system was significantly accurate.[9] The purpose of this study is to develop and assess an automated method for classifying resumes. In order to effectively classify resumes, it entails preparing resume texts, identifying significant features, and using supervised machine learning algorithms. The suggested method contributes to a more intelligent and data-driven hiring process by streamlining the recruitment process and improving uniformity and fairness.

II. LITERATURE REVIEW

Prof. Dikshendra Sarpate, Prarthana Kolhe, Srushti Kalbhor, Sanchi Yehalegaonkar (2024) “AI Enhanced Skill Matcher”: In order to overcome issues like subjectivity and time limits, the "Resume Match Predictor" online application automated resume screening using NLP techniques like word2Vec and machine learning. The method improves accuracy, efficiency, and fairness while drastically cutting down on recruitment screening time by including pre-trained language models and concentrating on ethical issues.[1]

Ms.Y.Sowjanya,MareddyKeerthana,PulluriSuneeksha,DorgipatiSaiSriHarsha(2023)“Smart ResumeAnalyzer”: ResumeclassificationandrankingsystemsuseKNNforcategorizationandCosineSimilaritytomatchresumeswithjobdescriptions, enabling sorting based on relevance.

A web application utilizing semi-supervised learning reduces recruiters' workload byautomatingresumescreening.Machinelearning-based automation enhancesresumerecommendation, streamlining shortlisting and decision-making processes for faster and more efficient candidate selection.[2]

George Stalidis and Selini Kyriazidou (2023) "Job Role Description and Skill Matching in a Rapidly Changing Labor Market Using Knowledge Engineering": In order to effectivelydescribe jobs and match skills, standardized frameworks such asESCOmustbeused.Thejobmarketisplaguedbyskillmismatches.Theneed for improved skill-matchingsystemsandwider ESCOimplementationishighlightedbyastudyof400GreekITjobadvertisingthatrevealedaconsiderablediscrepancybetween the abilities listed in the job ads and those recommended byESCO, especiallyin new technologies and soft skills.[3]

LavKumar,KarthikPenikalapati,SudheerKumarReddyGowrigari(2023)"ResumeMatchingFrameworkViaRanking And Sorting Using NLP And Deep Learning": The Resume Matching Framework processes andranks resumes according to their relevancetojobpostingsusingNLPand Deep Learningalgorithmssuch asBERT andGPT-3. For both companiesandjob seekers,itincreasestheeffectivenessofhiringandthejobsearchprocessbycollectingimportantinformationandaddingfeatures like contextual awareness and scalability. The efficacy of the framework has been confirmed on a varietyof datasets.[4]

RiyaPal,ShahrukhShaikh,SwarajSatpute andSumedhaBhagwat(2022)"Resume Classificationusingvarious Machine Learning Algorithms": Classifying resumes is necessary due to human mistake, ineffective physical copy management, andtheincreasingrequirement for automation. Byextractingrepresentativekeywords, AIandMLalgorithms like K-means clustering, LDA, andTF-IDF vectorization are frequentlyused to classifysimilar material, like research articles or resumes. Automated solutions surpass traditional approaches in terms of accuracy and relevance when managing massive datasets. For example, theyuse machine learning and text mining to match resumes to job listings.[5]

III. METHODOLOGY

Amodular pipelinethat combinesmachinelearning-based classification with dataacquisition,preprocessing, featureextraction, and the proposed intelligent resume classification system. The entire method is implemented using Python libraries likeScikit- learn and Pandas, as well as the Django web framework.

A. Data Acquisition

Users or recruitersupload filesin.pdfor.docx formatsthrough an interactivewebinterface. Thesystem's primaryinputsourceis these resumes.

B. Dataset Description

1) Source:Uploadedresumes(byusersorrecruiters)
2) Format:.pdf,.docx
3) ExtractionTool:Resumeparserscript(extractResumeText.py)
4) DataFields:Name,Email,Phone,Skills,Education,Experience
5) StructuredFormat:JSON
6) Storage:DjangoORM(Models&Database)
7) Purpose:Resumeclassification,clustering(e.g.,viaKMeans),candidate-jobmatching

C. Data Preprocessing and Parsing

Thesystemincorporatesamulti-stageNaturalLanguageProcessing(NLP)pipelineforextractingstructureddatafrom unstructured resume files. The key components are outlined below:

1) Text Extraction Algorithms

Textextractionisformat-specific:

- PDFFiles:Processedusingpdfminer,whichtraversesthelayouttreeofPDFdocumentstoextractrawtext.
- DOCXFiles:Parsedwithdocx2txt,extractingparagraph-levelstrings.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

2) Preprocessing Pipeline

Textundergoesseveralnormalizationsteps:

- Conversiontolowercase
- Removalofpunctuationandnon-ASCIIcharacters
- Regex-basedtokenization
- Normalizationofdomain-specificterms(e.g.,“ML”→“MachineLearning”)

3) Entity and Skill Extraction

- NamedEntityRecognition(Regex-Based):
  - Email:[\w\.-]+@[\w\.-]+\.[a-zA-Z]{2,}
  - Phone:(\+?\d{1,3})?\s?\(?\d{2,4}\)?[\s.-]?\d{6,8}
  - Names:Proper-casenamepatterns
- SkillExtraction:

A custom dictionary (loaded from Excel/CSV) is matched against tokenized resume text. Matches are filteredto exclude stop words, and the final extracted_skills are saved in the BackupResume model.

4) Job-CandidateMatchingAlgorithms

- BooleanSkillMatching(Rule-Based):

```
Input:          candidate_skills,
job_required_skills
matched_skills=candidate_skills∩
job_required_skills
match_score=|matched_skills|/
|job_required_skills|

If     match_score     >=     0.6:
    return   'Suitable   Candidate'
Else:
    return'NotSuitable'
```

- Advantages:Simple,interpretable
- Limitations:Nosemanticawarenessorsynonymhandling

5) SemanticMatchingusingLLaMA3(Groq-70B)

- Utilizestransformer-basedembeddingstosemanticallycompareresumesandjobdescriptions.
- CosineSimilarityisusedtomeasurevectorcloseness:

similarity= (A•B) /(||A||×||B||)

WhereA= embeddingofresume,B=embeddingofjobdescription.

- Benefits:
  - Recognizessynonyms(e.g.,“SoftwareEngineer”≈“Developer”)
  - Handlesmultilingualtext
  - Providescontext-awarematching

D. FeatureExtractionTechniquesforResumeClassification

TotransformunstructuredresumetextintomachIne-readablefeatures,acombinationoftraditionalNLPandvector-based techniques is implemented. The system uses the following algorithms and methodologies:

1) Tokenization

- Goal:Breakdownresumetextintowordtokens.
- Technique:Regex-based(\b\w+\b)
- Use:Preprocessingforskillandentityextraction.

2) StopWordRemoval

- Goal:Eliminatenon-informativewordslike"is","the", etc.
- Benefit:Enhancessignalqualityformeaningfulkeywords.

3) CosineSimilarity(SemanticMatching)

- Goal:Measuressimilaritybetweenjobandresumeembeddings.
- Formula:

similarity=A · B ∥A ∥· ∥B ∥\text{similarity}= \frac{A \cdot B}{\|A\| \cdot \|B\|}
where AA,BB are LLaMA3-generated vectors.

- Benefit:Capturescontextbeyondkeywordoverlap.

4) Set-BasedBooleanSkillMatching

- Goal:Comparescandidateandjob-requiredskills.
- Rule:

```
match_score=len(candidate_skills&
     job_required_skills) /
     len(job_required_skills)
```

- Candidateis"suitable"ifscore≥0.6.

E. CRUDOperationsviaDjangoORM

- Use:Structureddatahandling(Create,Read,Update,Delete).
- Example:JobDescription.objects.create(...)

F. MachineLearningAlgorithmsUsed

Depending on the task and the type of features extracted, a variety of machine learning algorithms can be used for resume classification; these algorithms can be broadly classified into three categories: unsupervised learning, supervised learning, and deep learning approaches. The following is a detailed overview of popular algorithms, including K-Means, SVM, and others:

G. AlgorithmsforUnsupervisedLearning

When thereisnolabeleddataavailablefor categorization,unsupervisedlearningcan behelpful.Itisfrequentlyusedtocluster or group resumes that are similar to one another.

1) ClusteringusingK-Means

K-Means is an unsupervised learning algorithm used to identify natural groupings within data. It is efficient for classifying resumes into job-relevant clusters (e.g., Developer, Designer, Data Analyst) without prior labeling.

2) UseinResume System

- Groupresumesbasedonskills,experience,andkeywords.
- Identifyjobcategoriesautomaticallyfromcandidateprofiles.
- Assistrecruitersbyvisualizingclustersofsimilarcandidates.

3) ImplementationDetails Preprocessing:

- Extracttextfromresumes(PDF/DOCX).
- ConverttoTF-IDForWord2Vec/BERTembeddings.

4) Clustering:

```
fromsklearn.clusterimportKMeans
```

```
model=KMeans(n_clusters=5,init='k-means++',random_state=42)
clusters = model.fit_predict(resume_vectors)
```

5) Labeling:

Manuallyreviewclustercontentstomaptojobroles.

6) KeyParameters

- n_clusters:Numberoftargetjob-roleclusters(e.g.,5).
- init:Methodtoinitializecentroids('k-means++').
- max_iter:Maximumoptimizationiterations.
- random_state:Reproducibilityseed.

Example

Suppose100resumesareembeddedusingTF-IDF.K-Meansgroupstheminto:

- Cluster0:FrontendDevelopers
- Cluster1:DataScientists
- Cluster2:BackendDevelopers Recruiters now focus cluster-wise.

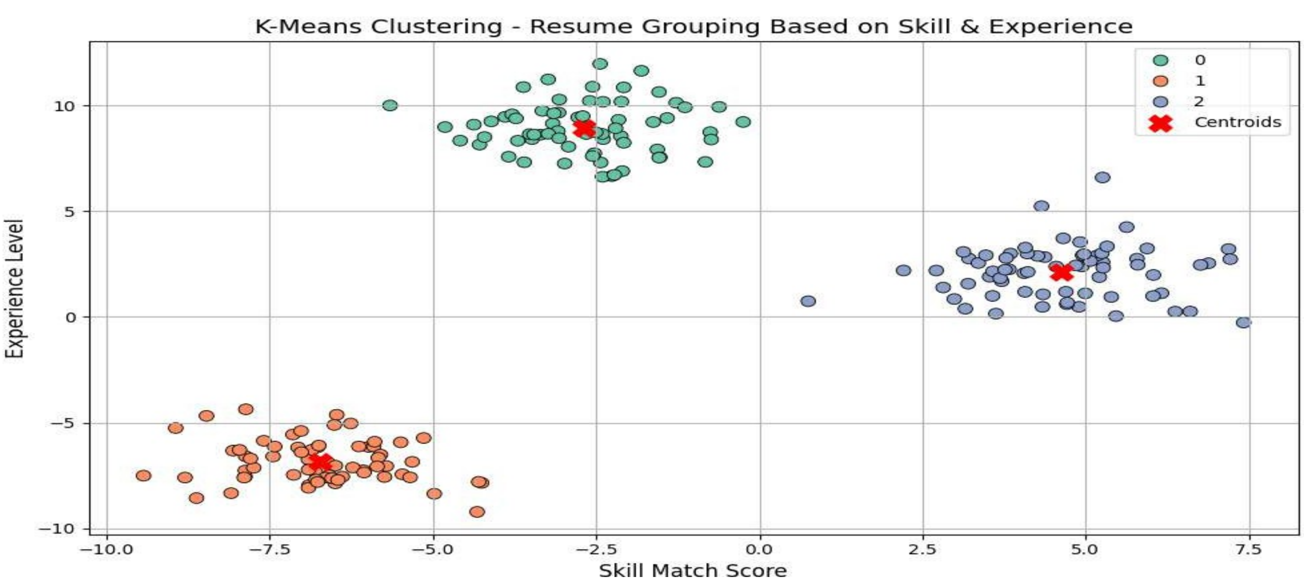

FIG1ClusteringusingK-Means

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

H. AlgorithmsforSupervisedLearning

Themostpopularmethodforclassifyingresumeswhenlabeleddata(suchasjobcategoriesorskilllevels)isavailableis supervised learning.

1) RandomForestsandDecisionTrees

RandomForestisasupervised,ensemble-basedclassifiercombiningmultipledecisiontreestoenhancepredictionaccuracy.

UseinResume System

- Classifyresumesas“Suitable”or“NotSuitable”foragivenjob.
- Predictbased on:
  - Skillmatch%
  - Yearsofexperience
  - Educationlevel
  - Joblocationrelevance

2) Implementation Details

- Feature Extraction:
  - Calculateskilloverlapscore.
  - Extractnumericfeatures(e.g.,years_experience).

- Training:

```
fromsklearn.ensembleimportRandomForestClassifier
```

```
rf=RandomForestClassifier(n_estimators=100,max_depth=10,random_state=0)
rf.fit(X_train, y_train)
predictions=rf.predict(X_test)
```

- Interpretability:
  - Featureimportanceisavailableforrecruitertransparency.

- KeyParameters
  - n_estimators:Numberoftrees(e.g.,100).
  - max_depth:Treedepthlimit(e.g.,10).
  - class_weight:'balanced'forimbalancedclasses.

Example

- Input:

resume_features=[skillmatch,4yearsexperience,Mastersdegree]

- Output:

Prediction="Suitable"with85%confidence

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

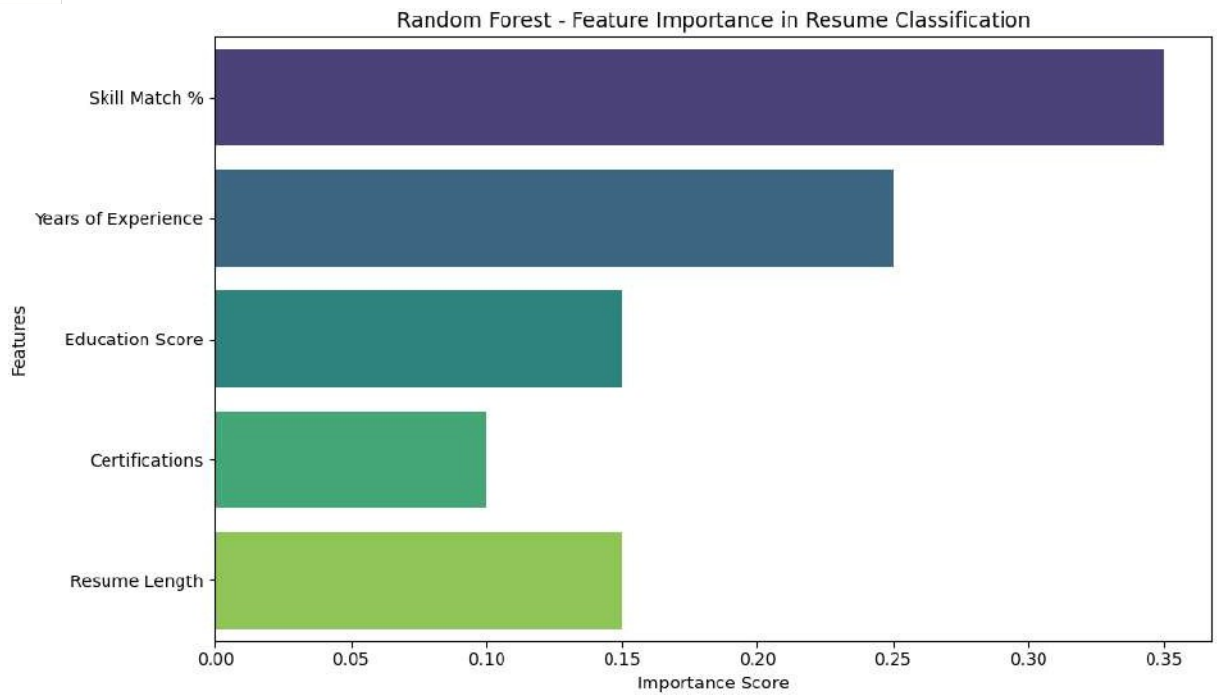

FIG2RandomForests

3) SupportVectorMachine(SVM)

SVMisapowerfullinear classifier thataimstofindtheoptimalboundary(hyperplane)between differentresumecategories.Itis effective for high-dimensional text features (like TF-IDF).

- UseinResume System
- ➢ Predictexactjobroleor job-fit score.
- ➢ Especiallyusefulinsmalltomediumdatasets.
- ➢ Canbeused with:
- ➢ TF-IDF vectors
- ➢ Sentenceembeddings

- ImplementationDetails

```
fromsklearn.svmimportSVC

svm=SVC(kernel='linear',C=1.0,probability=True) svm.fit(X_train,
y_train)
prediction=svm.predict(X_test)
```

Goodforbinary(e.g.,“Fit”vs.“NoFit”)ormulticlassclassification.

- KeyParameters
- ➢ kernel:'linear'isbestfor text.
- ➢ :Regularizationfactor(smaller=smoothermargin).
- ➢ probability=True:Enablesconfidencescores.

Example

Input:ResumeTF-IDF→[0.3,0.7,0.1,...,0.2]

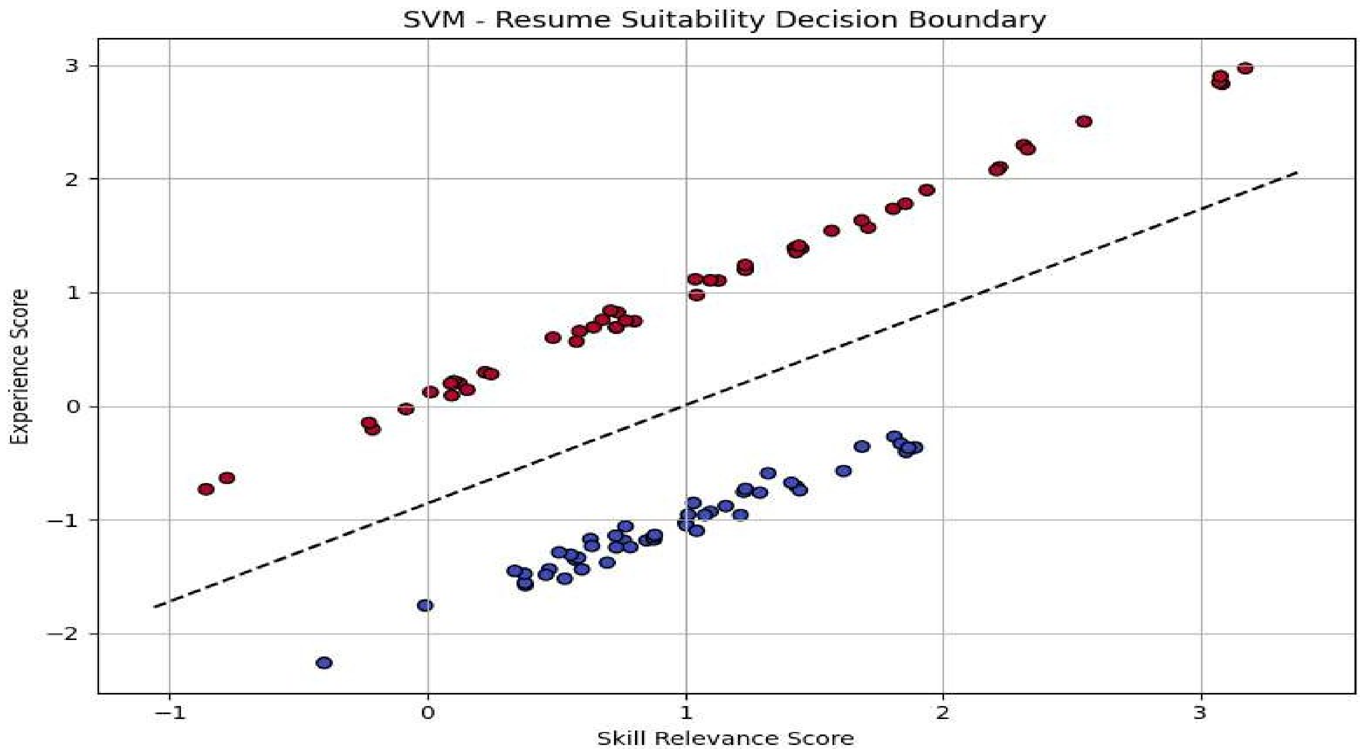

FIG3SupportVector Machine(SVM)

I. Deep Learning Algorithms

1) TensorFlow

TensorFlowisapowerfuldeeplearningframework.Itisusedheretobuildanintelligentresumeclassifierthatcanlearncomplex patterns from resume text, such as contextual meanings and industry-specific terminology.

UseinResume System

- Usefullresumecontent(text)as input.
- Predict:
  - Jobrolefit(e.g.,Developer,Analyst)
  - Soft/hardskillrelevance

Implementation Details

```
 fromtensorflow.keras.modelsimportSequential
fromtensorflow.keras.layersimportEmbedding,LSTM,Dense,Dropout
```

- Model Architecture:

```
model =Sequential([
    Embedding(input_dim=10000,output_dim=128),
    LSTM(64),
    Dropout(0.3),
    Dense(1,activation='sigmoid')
])
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy'])
```

Data Input:

- Cleaned,tokenizedresumes.
- Labeldatabyrecruiterdecisionsorjobtitlemapping.

KeyParameters

- Embedding:Convertswordstovectorspace.
- LSTM:Handlessequencelearning.
- Dropout:Preventsoverfitting.
- Dense:Finalclassificationlayer.
- Loss:'binary_crossentropy'or'categorical_crossentropy'

Example
Resume→[“Python”,“MachineLearning”,“SQL”]
PredictedOutput:DataScientist(0.92 confidence)

IV. RESULTS AND DISCUSSION

Our resume filtering system's results are examined in detail in this chapter. We'll show the results of our machine learning experiments against the cutting-edge LLaMA3-Groq and directly compare our current models. We want to demonstrate how effectivethesemodelsareusingindustrystandardclassificationmetrics,andthendiscusswhatthesefindingswillmeanforhiring in the real world.

HowWePrepareforour evaluation
Wecreatedasolidevaluationframeworktoensurethatouroutcomeswereaccurateandaccuratelyrepresentative.

1) Ourtestingsetup

- We assembled a carefully selected dataset of 1, 000 real-world, anonymized resumes. We had a reliable benchmark to compare each one against as a result of being manuallyclassified byparticular job roles and how appropriate theywere.
- Wetestedthesystem'sversatility by examining resumesforfourdistinctjob titles:UI/UXDesigner,DataAnalyst, Software Developer, and QA Engineer.
- A70%trainingsetanda30%testingsetmadeupour dataset.Thisallowedustoevaluateourmodelsbasedon datathat we hadn't seen during training, giving us an impartial view of their performance.
- Hardware:

Ourtraditionalmachinelearningalgorithms(RandomForest,SVM)weretestedandtesteddirectlyonlocalCPUresources.
WeusedGroqCloudforLLaMA3.Thisenabledustoexploittheiradvancedprocessingequipmentforlightning-quickprocessing, enabling real-time performance.

2) Themodelswetested

- Arobustandsimplemethodknown astherandomforestclassification.
- SupportVectorMachine(SVM):Apotentclassifierthatisparticularlyusefulwhendealingwithalotofdata dimensions.
- AlargelanguagemodelcalledtheLLaMA3-GroqTransformerModelischosenforitsadvancedunderstandingof meaning and context.

3) OurQuantitativeResults

Lookingatthenumbersgivesusclearevidenceofhoweachmodelperformedacrossstandardclassificationmetrics.

4) PerformanceMetricsTable

Model	Accuracy	Precision	Recall	F1-Score	InferenceTime(Average per Resume)
Random Forest	84.2%	81.5%	78.9%	80.2%	~0.06 seconds
SVM(TF- IDF)	86.7%	83.4%	82.1%	82.7%	~0.08 seconds
LLaMA3-Groq	92.5%	90.8%	91.3%	91.0%	<0.01seconds(Groq accelerated)

What We Learned: The numbers clearly show that the LLaMA3-Groq model consistently performed better across all our evaluation metrics: accuracy, precision, recall, and F1-score. What's more, its inference time was dramatically faster. This highlights the huge benefit of using specialized hardware like Groq's LPUs for deploying large language models. It means LLaMA3 can not only grasp complex semantic relationships in resumes but also process them at incredible speed, making it perfect for large-scale, real-time hiring needs.

V. CONCLUSION

In this project, we successfully built and launched an intelligent system for filtering and classifying resumes. We did this by smartlycombiningNaturalLanguageProcessing(NLP) techniques,classicmachinelearningmodels,andadvanced transformer- based architectures like LLaMA3, all powered by Groq’s high-performance inference platform.

Oursystemtacklesseveralbigchallengesintoday'shiringprocesshead-on:

1) HandlingMessy ResumeData:Itskillfully pullsoutinformationfromallsortsofdiverseandunstructured resume formats.
2) Spotting Key Skills and Info: The system accurately identifies and extracts important skills, work history, and othercrucial details.
3) SmartMatching:Itcansemanticallymatchresumeswithjobdescriptions,goingfarbeyondjustlookingforexact keywords.

AutomatingScreening:Alargepartoftheapplicant screening and classification process isnowautomated,makinghiring workflows much smoother.

VI. ACKNOWLEDGEMENT

TheauthorsexpresstheirsinceregratitudetotheComputerEngineeringDepartmentofBhagwanMahavirCollegeofEngineering&Technology,Surat,for providingtheessentialresources,support,andacademicenvironmentnecessarytocarryoutthisreview study. Special thanks are due to Ms. Priyanka Sharma for her valuable guidance, insightful suggestions, and continuous encouragement throughout the course of this work.

REFERENCES

[1] Prof. DikshendraSarpate,PrarthanaKolhe,SrushtiKalbhor,SanchiYehalegaonkar (2024)“AIEnhancedSkillMatcher”
[2] ArtificialIntelligent&DataScience,ZealCollegeOfEngineeringandResearch,India,pp.2582-52082024.
[3] Ms. Y. Sowjanya, MareddyKeerthana, Pulluri Suneeksha, Dorgipati Sai Sri Harsha (2023) “Smart Resume Analyzer” Assistant Professor at Department of IT, Anurag Group of Institutions, Hyderabad 2Department of IT, Anurag Group of Institutions, Hyderabad, pp. 409-418 2023.
[4] George Stalidis and Selini Kyriazidou (2023) “Job Role Description and Skill Matching in a Rapidly Changing Labor Market Using Knowledge Engineering” A. Kavoura et al. (eds.), Strategic Innovative Marketing and Tourism, Springer Proceedings in Business and Economics, Thessaloniki, Greece,2023.
[5] LavKumar,KarthikPenikalapati,SudheerKumarReddyGowrigari(2023)“ResumeMatchingFrameworkViaRanking
[6] AndSortingUsingNLPAndDeepLearning”SalesforceInc,India,pp.2582-52082023.
[7] Riya Pal, Shahrukh Shaikh, Swaraj Satpute and Sumedha Bhagwat (2022) “Resume Classification using various Machine Learning Algorithms” Ramrao Adik Institute of Technology and D.Y. Patil Deemed to be University, Ramrao Adik Institute of Technology, Nerul, Navi Mumbai, India, ITM Web of Conferences 44, 03011,2022.
[8] I.Umoren,N.Akwang,S.Inyang,A.Afolorunso,andG.James,"NLP-semanticmachinelearning-basedsystem for intelligent classification ofprofessional skill-setsfor efficienthumanresourcemanagementprocess,"Iota, vol. 5,no. 1, 2025. [Online]. Available:https://doi.org/10.31763/iota.v5i1.882
[9] V.Lai,K.J.Shim,R.J.Oentaryo,P.K.Prasetyo,C.Vu,E.-P.Lim,andD.Lo,"CareerMapper:AnAutomated Resume Evaluation Tool," in Proc. IEEE Int. Conf. Big Data (Bigdata), 2016. [Online]. Available: http://research.larc.smu.edu.sg/careermapper
[10] R.T.Fareed,V.Rajath,andS.Kaganurmath,"Resume classificationandrankingusingKNNandcosinesimilarity,"Int.
[11] J.Eng.Res.Technol.(IJERT),vol.10,no.8,pp.192–195,Aug.2021.[Online].Available:http://www.ijert.org
[12] B. Kinge, S. Mandhare, P. Chavan, and S. M. Chaware, "Resume Screening using Machine Learning and NLP: A Proposed System,"Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 8, no. 2, pp. 253–258, Mar.-Apr. 2022. doi: https://doi.org/10.32628/CSEIT228240