Journal of
Artificial Intelligence and Soft Computing Techniques

Research Article Open Access

Development of Machine Learning Based Staff-Task Matching Models for Consulting Sector

Onur Aygün[1], Ceren Ulus[2] and Mehmet Fatih Akay[3*]

[1]Innovance Information Technologies, Department of Software Development, Istanbul, Turkey
[2]EFA Innovation Consulting and Technology, Department of R&D, Adana, Turkey
[3]Cukurova University, Faculty of Engineering, Department of Computer Engineering, Adana, Turkey

[*]**Corresponding Author:** Mehmet Fatih Akay, Cukurova University, Faculty of Engineering, Department of Computer Engineering, Adana, Turkey, Tel: +90-530-2752381, E-Mail: mfakay@cu.edu.tr

Received Date: August 16, 2025 **Accepted Date:** August 26, 2025 **Published Date:** August 30, 2025

Citation: Onur Aygün, Ceren Ulus, Mehmet Fatih Akay (2025) Development of Machine Learning Based Staff-Task Matching Models for Consulting Sector. J Artif Intel Sost Comp Tech 2:1-12

Abstract

Nowadays, the consulting industry plays a critical role in ensuring customer satisfaction, as the quality of the services it offers directly impacts customer loyalty and business success. To ensure rapid and effective customer satisfaction, it is essential to select and assign the right staff appropriate to the service. Correct staff-task matching not only increases the efficiency of the service process but also boosts staff motivation, strengthening the corporate brand image. In this context, this study developed staff matching models using various machine learning algorithms including Support Vector Machines (SVM), Light Gradient Boosting Machine (LightGBM), Extreme Learning Machine (ELM), and Logistic Regression (LR) to ensure the compatibility of staff with their tasks. Furthermore, ensemble learning approaches, including Voting, Bagging, Boosting, and Stacking, have been applied to the developed models to enhance their performance and generalization capabilities. The performance of the developed models has been evaluated with Precision, Recall, Accuracy, and F1-Score. Extensive experiments revealed that the model combining the LightGBM algorithm with the Bagging method has shown the superior performance. These results demonstrate the effectiveness of machine learning-based ensemble models in ensuring staff-service fit in consulting services.

Keywords: Staff-Task Matching; Machine Learning; Ensemble Learning; Support Vector Machines; Logistic Regression

Introduction

In today’s global word, individuals or organizations may need external expert support to address the challenges they face in their fields of activity, changing market dynamics, or strategic growth goals. In this context, consulting services are gaining importance as professional guidance processes provided by individuals or organizations specialized in a specific field, drawing on their knowledge, experience, and analytical perspective. Through consulting services, companies can make more accurate assessments of their organizational structures, processes, and goals; they can address problems from a more objective perspective and take more accurate steps in developing solutions.

The effective and sustainable delivery of consulting services is not limited to the presence of expert knowledge and experience; it also requires structuring this knowledge within the right strategic approach. In this context, one of the key determinants of success in consulting processes is the staff-task match. The staff-task match plays a critical role in transforming potential clients into long-term, satisfied clients [1]. This matching process takes into consideration multidimensional criteria such as the expertise, sectoral experience, technical and social skills, communication skills, and previous successes of the consulting staff. At the same time, the client's needs, expectations, sectoral position, and the specifics of the problem they are seeking a solution for are analyzed to ensure the most suitable consultant is assigned. This strategic alignment maximizes customer satisfaction and increases the effectiveness and efficiency of the consulting service [2].

Incorrect staff-task matching leads to the delivery of services that do not meet the client's needs and expectations, leading to decreased customer satisfaction and, consequently, to damage to the company's corporate reputation [3]. Assigning staff unsuitable for the service leads to a loss of efficiency and a decrease in quality standards in project management processes. This leads to extended processes, additional resource and time requirements, and consequently, increased costs, negatively impacting business budgets. Furthermore, staffs working in roles not aligned with their areas of expertise experience a loss of motivation over time, leading to lower individual performance and disruptions to business processes. Demotivated staffs weaken their commitment to the organization, increasing their tendency to leave their jobs. Increasing staff turnover leads to a loss of corporate knowledge, the use of additional resources in recruitment and onboarding processes, and a loss of temporary workforce productivity, negatively impacting a company's operational efficiency and brand value. On the other hand, accurate staff-task matching plays a critical role, as assigning staff to appropriate service areas based on their knowledge, technical competencies, and communication skills not only improves service quality but also strengthens staff motivation and commitment to the organization. This, in turn, results in more efficient business processes [4].

This study aims to develop staff-task matching models for the consulting sector. To this end, staff-task matching models have been developed using machine learning-based SVM, LightGBM, ELM, and LR. In addition, the aim has been also to make the developed models more robust and reliable. In this context, ensemble learning approaches, including Voting, Bagging, Boosting, and Stacking, have been also incorporated into the model development process.

Studies in the literature address the task matching problem at a general level and do not offer approaches for matching specific employees to specific tasks. Furthermore, most existing studies in the literature focus on the suitability of candidate personnel for the desired task. This study, however, focuses on matching a task with an employee. This study specifically focuses on the consulting industry. In the consulting industry, where service quality directly impacts customer satisfaction and loyalty, the problem of matching service personnel to tasks is specifically addressed. Methods such as SVM, Long Short-Term Memory (LSTM), Multi Layer Perceptron (MLP), and Random Forest (RF) are frequently used in the literature. In addition to a general evaluation of SVM, LightGBM, ELM, and LR algorithms, this study also applies and evaluates ensemble learning approaches such as Voting, Bagging, Boosting, and Stacking. A key limitation of this study is the relatively limited scope of the dataset used. It is believed that using a broader and more diversified dataset could improve the generalizability of the models.

This study is organized as follows: Section 2 includes relevant literature. Dataset generation is presented in Section 3. Methodology is presented in Section 4. Staff-task matching models are presented in Section 5. Results and discussion are given in Section 6. Section 7 concludes the paper.

Literature Review

[5] presented a Bidirectional Encoder Representations from Transformers (BERT) based approach that uses deep contextual embeddings to improve job-applicant matching. SVM, LSTM and MLP models have been trained, and their performance has been evaluated using BERT embeddings obtained from real-world job definitions. The results revealed that BERT-powered models performed superior compared to traditional keyword matching techniques. The SVM-based model, which is the most successful model, provided an Accuracy value of 94%.

[6] developed a suggestion system which aims to provide job seekers with more appropriate and diverse job suggestions and to make accurate job matches to individuals. An up-to-date and comprehensive job title dataset has been created by combining data from the European Skills, Qualifications, Qualifications and Occupations and European Employment Services. By extracting meaning from unstructured texts such as Curriculum Vitae (CV), more accurate job suggestions have been presented. The performance of the proposed system has been evaluated both through real job postings and through CV and feedback from Human Resources (HR) experts. The results obtained revealed that the proposed system provides more effective results compared to both traditional and current methods.

[7] developed a data-driven machine learning model with a forecast time of 5 business days and validated in a practical case study of a company. Within the scope of the case study, an optimized workforce planning has been carried out based on the predicted delivery positions. The results revealed that the developed model outperformed both the manual prediction approach used in practice and the automatic machine learning models in terms of accuracy, especially in short-term predictions.

[8] calculated the semantic similarity between Occupational Information Network (ONET) assets and the texts extracted from the CV of job candidates using the ONET database and deep learning technologies. In the first scenario, it has been aimed to identify the ONET occupations that showed the highest compatibility with the candidate's CV; In the second scenario, the most appropriate job selection has been made by the candidate's CV and job descriptions. Evaluations revealed that the proposed approach had better performance than the basic methods in both scenarios. In addition, a usage function has been developed where course suggestions can be submitted to improve certain skills of candidates and become more qualified for certain jobs.

[9] proposed a method for person-position matching in a probabilistic hesitant fuzzy environment involving multiple attribute preferences and time factors. First, plus and minus ratings have been calculated based on the evaluation matrices provided by position managers and job seekers; then, using these data, attribute weights and satisfaction levels have been determined. Multiple matching models have been created by using time satisfaction matrices and optimal matching solutions have been obtained. The effectiveness of the proposed method has been confirmed by the sample analysis performed.

[10] proposed a matching method that considers multiple service expectations in the field of elderly care. First, the actual values of the expectation indexes and the expectation requirements of the elderly and service staff have been determined. Then, the satisfaction levels of both parties have been calculated based on the type of flexible service expectation and it has been decided which side would be preferred in the matching process in line with the inelastic service expectation indexes. Based on this approach, two-sided matching models have been created by considering both underserved and underserved staff scenarios and optimal matching results have been obtained.

[11] proposed an approach using Analytic Hierarchy Process combined with the Simulated Annealing algorithm to enhance the processes of candidate selection and task assignment. A case study has been applied to a Thai manufacturing enterprise to observe the effectiveness of this combination. Results showed that successful candidate re-

cruitment and staff-task matching have been achieved with accurate and rapid quantitative evaluation. The lack of job changes or resignations after one year of observation indicated the practical applicability of the proposed method.

[12] presented a system which leverages machine learning and Natural Language Processing to improve job recommendations. A dataset which is purged from prior user interactions has been used to ensure impartiality of the system. Collaborative and content-based filtering methods have been used to develop candidate recommendation algorithms. The performance of the system has been evaluated using Precision, Recall, and F1-Score metrics.

[13] proposed the Deep Neuro-Fuzzy-based Bilateral Location Privacy-Preserving (DNF-BLPP) scheme to eliminate the staff-task matching rate problem. Non-negative Constraint Matrix Factorization algorithm has been used to complete the missing data based on time-space correlation. The results revealed that DNF-BLPP was effective in the staff-task matching ratio.

[14] proposed the Workforce Composition Balance (WCB) framework for effective task assignment and management of tasks that require skill. Within the scope of this framework, the Volunteer Retention and Value Enhancement algorithm has been combined with skill-based task assignment methods. The results showed the effectiveness of the WCB framework.

[15] proposed a two-stage method for task allocation to optimize the tasks and labor resource matching. In the first stage, relationship analysis, fuzzy clustering and service aggregation operations have been performed. In the second stage, a task allocation model based on talent and collaboration has been proposed. Besides, the Improved Adaptive Genetic Algorithm has been presented in the second stage.

[16] aimed to develop a model that provides appropriate staff-task assigning at the appropriate time, enabling organizations to make more efficient decisions. For this purpose, a prototype has been created with machine learning algorithms to improve staff evaluation using real-time data. According to the performance evaluation of the model, the RF method provided a 90% Accuracy rate.

[17] proposed a job matching forecast model to decrease the duration of recruitment processes and to select capable candidates, using EfficientNet model. HR data have been analyzed to obtain feature representations for each staff. The obtained features have been used as input in the system and performance of each individual in a certain position has been estimated. These estimations have been used for staff-job matching. According to the results, %86.8 Accuracy and 0.413 of Loss value have been obtained with the proposed model.

Dataset Generation

The dataset has been provided from Innovance HR. It contains 840 rows of data and staff CV. The dataset contains manually labeled real class information consisting of values of 0 and 1 representing candidates who have been hired and those who have been not. Records containing missing or incorrect data have been excluded. This created a balanced (positive-negative) classification dataset. The model's input features also included features that measure the textual similarity of candidate CVs and position descriptions. This similarity metric has been used to assess the fit between the candidate and the position. In this context, for each match, the position, role description, and CV text have been converted into semantic vectors using the Sentence-BERT (SBERT). SBERT similarity scores have been calculated for all samples. Thus, by effectively using unstructured texts in addition to structured data, much more accurate and explainable matching results have been obtained.

Various features have been created to increase model success. These are the semantic similarity score, CV word count, number of words in the position and role text in the CV, and number of position name words in the CV. The attributes and their descriptions are given in Table 1.

Table 1: Attributes and their descriptions

Attributes	Descriptions
Data_id_num	A number that the same as the name of the CV file and uniquely identifies each row of data.
Blind_resume	The name of the CV file that has been anonymized
Position	The target position/title required by the relevant consulting service or project
Project_and_role_detail_description	A description of the expected expertise, competencies, past experience, and project-specific requirements for the position.
Client_name	The anonymized name of the client
Client_Industry	The client's industry
Label	A binary label that indicates the accuracy of the match.
Technologies	Technologies in which the candidate has expertise and actively uses in their projects.
Education	The university and department the candidate graduated from.
Experience_work_experience	The candidate's previous companies, positions, work dates, and key responsibilities.
Projects	The projects in which the candidate has participated.
Similarity	The semantic similarity score between the Position + Project and Role Detail Description and the CV text (float, between 0 and 1).
CV_word_count	Total word count in the CV text.
Common_kw_count	Total number of words in the position + role text used in the CV.
Position_kw_in_cv	Total number of times the words in the position name appear in the CV.

Methodology

Support Vector Machine

SVM is a technique based on statistical learning theory. While initially designed for classification, SVMs' primary applications now include regression and the classification of small, high-dimensional, nonlinear datasets. SVMs are built on the principles of statistical learning theory, minimizing the VC dimension and structural risk. Learning occurs without error identification using a limited sample size, and the model's sensitivity is analyzed. The minimum deviation of the hyperplane from the sample points is used to determine the optimal universal capability. SVMs include linear and nonlinear regressions. The kernel function, which evaluates the similarity between data points, and the cost loss function, or regularization parameter, are critical parameters [18].

Light Gradient Boosting Machine

LightGBM uses tree-based learning methods that are considered computationally powerful. It stands out as a fast-processing algorithm. Importantly, LightGBM grows vertically, meaning it grows in the form of leaves; other algorithms grow horizontally, creating trees [19].

Logistic Regression

LR is a frequently used statistical analysis technique today for predicting binary outcomes, such as yes or no, based on past observations in a dataset. It is based on a logistic function that transforms probability into a value between 0 and 1. The model includes an optimization procedure that maximizes the probability of the observed data given the predicted probabilities [20].

Extreme Learning Machine

ELM, a two-layer neural network with training in the second layer and fixed and random in the first layer, has emerged as a critical algorithm today. Recently, ELM has become a preferred choice for feature selection, clustering, regression, and classification. Training of ELM is achieved through hardware implementation and parallel computing techniques. ELM is currently being heavily utilized in many fields, including computer vision and biomedical engineering. Based on theories of generalization performance in neural networks, ELM posits that hidden neurons are important but rarely require tuning. In these theories, the weights connecting inputs to hidden nodes are randomly assigned and never updated due to the randomly generated structure of the hidden nodes [21].

Voting

A Voting classifier is a machine learning model that trains an ensemble of various models. In this method, the findings from each classifier are fed into the Voting classifier, and the output class is predicted based on the highest majority of votes. Voting ensemble techniques are often used in ensemble machine learning models to combine predictions from multiple models [22].

Stacking

Proposed by Wolpert in 1992, stacked generalization, also known as Stacking, is a heterogeneous learning technique that combines several base learners to train a model, unlike the homogeneous Bagging and Boosting methods, which directly combine the outputs of several learners to obtain the final prediction. Typically, Stacking consists of several base learners (level 0) and a meta-learner (level 1). The outputs of the base learners are given as input to the meta-learner. The precision of the base learners and their diversity, a measure of the dependence or complementarity between learners, significantly determine the performance of a Stacking algorithm [23].

Boosting

Boosting algorithms use a weighted averaging approach to transform weak learners into strong learners. During Boosting, the original dataset is divided into multiple subsets. Each subset is used to train the classifier, which in turn generates the model set. Examples misclassified by the previous model are weighted more heavily in creating new subsets. A subsequent concatenation procedure improves the performance of the overall model by integrating these weak models via a cost function. Each weak model is trained independently before the prediction process; however, no specific model selection is made at the end; instead, the outputs of the models are integrated. Boosting is a method that flexibly constructs multiple weak learners sequentially. Intuitively, each new model focuses on the examples previously observed to be the most difficult to classify, resulting in a stronger learner with low bias. In this approach, each weak learner is trained, predictions are made, misclassified examples are identified, and then the next weak learner is trained with an updated training set containing these incorrect examples [24].

Bagging

Bagging is an important technique used to improve the stability of machine learning models. Foreseeing the various advantages of stability, Breiman proposed an ensemble meta-algorithm to stabilize any basic learning algorithm. Short for bootstrap aggregation, Bagging retrains the basic algorithm by subjecting it to various perturbations on the training data and averaging the resulting predictions. This method uses resampling techniques to reduce variance, remove discontinuities, and increase the stability of the basic algorithm A. The meta-algorithm randomly generates bags from the training dataset D, runs the basic algorithm A on each bag, and produces the final prediction by averaging the outputs of the resulting models [25].

Staff-Task Matching Models

The staff-task matching models have been developed using SVM, LightGBM, ELM, and LR. Additionally, the impact of ensemble learning approaches such as Voting, Bagging, Boosting, and Stacking on prediction performance has been also evaluated. Model hyperparameter ranges are provided in Table 2.

Table 2: Model hyperparameter ranges

Model	Hyperparameter Range
SVM	“C_Values”: [0.1 - 100]
LightGBM	“Verbosity”: [-1],
	“Seed”: [42],
	“Num_Boost_Round”: [100],
	“Num_Leaves”: [15, 31, 63],
	“Min_Child_Samples”: [5, 10, 20],
	“Max_Depth”: [-1, 5, 10],
	“Learning_Rate”: [0.01, 0.05, 0.1],
	“N_Estimators”: [50, 100, 200],
	“Min_Split_Gain”: [0, 0.01, 0.1]
LR	“Random_State”: [42],
	“Max_Iter”:[100-1000],
	“C”: [0.01, 0- 100]
ELM	“N_Hidden”: [50]
	“Random_State”: [42]
Voting	LR (“C”: [10],
	“Max_Iter”: [1000], “Random_State”:[42])
	SVM(“C”:1.0, “Random_State”: [42])
	LightGBM (“Learning_Rate”: [0.01], “Max_Depth”: [-1], “Min_Child_Samples”: [20], “Min_Split_Gain”: [0], “N_Estimators”: [100], “Num_Leaves”: [15], “Random_State”: [42])
Bagging	LR (“N_Estimators”: [50])
	SVM and LightGBM (“N_Estimators”: [20],
	“Random_State”: [42], “N_Jobs”: [-1])
Boosting	LR (“C”: [10], “Max_Iter”:[1000], “Random_State”: [42]), “N_Estimators” : [20], “Learning_Rate”: [1.0])
	LightGBM (“Learning_Rate” : [0.01], “Max_Depth” : [-1], “Min_Child_Samples”: [20], “Min_Split_Gain”: [0], “N_Estimators”: 100, “Num_Leaves”: [15], “Random_State”: [42])
Stacking	LR (“C”: [10], “Max_Iter”: [1000])
	SVM (“C”: [1.0])
	LightGBM (“Learning_Rate”: [0.01], “Max_Depth”: [-1], “Min_Child_Samples”: [20], “Min_Split_Gain”: [0], “N_Estimators”: [100], “Num_Leaves”: [15])

Development of the SVM Based Models

Model 1

In the first experiment, the SBERT (Sentence-BERT) model has been used to calculate semantic similarity scores between each "position + project description" and the CV text. The dataset, which contains the obtained cosine similarity scores and their true/false match information (label: 1/0), has been used in the SVM model for non--linear, high-discriminative classification. The RBF (Radial Basis Function) kernel has been chosen as the model's kernel function, gamma = "scale." Twenty percent of the dataset has been reserved as the test set, and the training process has been conducted using this structure. This method aimed to distinguish the correspondence between position-project descriptions and CVs with high accuracy.

Model 2

All embeddings generated by the SBERT model (both CV-based and position/project-based) have been saved on disk in .pkl format during the initial run. In subsequent training and testing steps, these embeddings have been loaded directly from disk, allowing similarity scores to be calculated without the need to rerun the SBERT model. This approach significantly accelerated the similarity calculation process and improved resource efficiency. Thus, despite the increased data size, model training has been completed in minimal time because no additional SBERT inference operations have been performed.

Model 3

To improve model performance, various combinations of kernel functions and the Cost and Gamma hyperparameters have been evaluated. In this context, the kernel functions have been rbf, linear, poly, and sigmoid; and the Gamma values have been scale and auto.

Model 4

In the Model-3 phase, systematic performance tests have been conducted on different kernel functions and hyperparameter combinations of the SVM model, and the parameters that provided the highest performance have been identified. In the Model 4 phase, the model has been retrained using the best parameter combination identified in Model 3.

Model 5

The kernel, C, and gamma hyperparameters of the SVM model have been systematically scanned using the GridSearchCV method, and the best parameter combination has been determined based on the cross-validation F1 score.

Development of the LightGBM Based Models

Model 1

The LightGBM model has been configured with the objective parameter "binary" for the binary classification problem (correct match/incorrect match). The metric parameter "binary_logloss" has been selected as the loss function, and the verbosity value has been set to -1 to suppress unnecessary warnings and outputs. The boosting_type parameter "gbdt" (Gradient Boosting Decision Tree) has been selected as the Boosting method, and the seed value has been fixed at 42 to ensure repeatability of the experiments.

Model 2

In addition to the semantic similarity score calculated with SBERT, three new features have been created to improve the model's performance. The cv_word_count feature provides an additional indicator for more comprehensive CVs by representing the total number of words in the CV. The common_kw_count feature indicates the number of keywords common to the position description and the CV. The position_kw_in_cv feature indicates the number of occurrences of each word in the position name in the CV. These additional features aim to contribute to more accurate modeling of structural and contextual similarities between the position and the CV.

Model 3

Basic hyperparameters of the LightGBM model have been optimized using the GridSearchCV method.

Development of the LR Based Models

Model 1

To prevent overfitting, L2 regularization has been applied with penalty = 'l2', and solver = 'lbfgs', an efficient and fast algorithm, has been used in the analysis process.

Model 2

As with the LightGBM model, the LR model has been retrained with three new features (cv_word_count, common_kw_count, and position_kw_in_cv) generated in addition to SBERT similarity. The maximum iteration count (max_iter) for the model has been increased from 100 to 1000 to anticipate that multiple features and balanced class weights (class_weight) might require more optimization iterations. Furthermore, the class weight parameter (class_weight) has been updated from None to balanced, thus equalizing the impact of positive and negative examples, thus preventing the model from exhibiting bias.

Model 3

In order to determine the optimal Beta (C) value, grid search method has been applied on the LR model and tested with 3-fold cross-validation.

Development of the ELM Based Models

The ELM model has been trained with parameters n_hidden = 50 (number of hidden neurons), activation = tanh (activation function), and random_state = 42 (for repeatability of results). While not as complex as deep learning methods, ELM has been considered a faster and more flexible alternative to classical machine learning approaches. Using additional features generated with SBERT similarity, the goal was to achieve high accuracy in a short time; it has been anticipated that the advantages of ELM would become more evident, especially when working with more features and data.

Development of the Voting Based Models

LR, SVM, and LightGBM models have been trained independently with the best hyperparameters determined previously. The outputs of these three models have been combined using VotingClassifier, applying both hard Voting (majority of class labels) and soft Voting (average of positive probabilities) strategies. By combining the strengths of the different algorithms, the uncertainty and error tolerance of the model have been reduced.

Development of the Bagging Based Models

To improve the overall accuracy and reliability of the model, the Bagging (Bootstrap Aggregating) method, an ensemble learning approach, has been applied. In this method, LR, SVM, and LightGBM-based models have been trained repeatedly on different sets of random samples of the training data. This approach aims to increase the generalizability of the system and achieve more stable predictions by reducing the sensitivity of a single model to random data fluctuations or outlier examples.

Development of the Boosting Based Models

Boosting is an ensemble learning method that aims to achieve a stronger model with higher overall performance by sequentially training weak learners. LightGBM and AdaBoost (based on LR) algorithms have been chosen for this approach. LightGBM, itself a GB algorithm and based on the principle of direct boosting, is used here for comparison purposes only. AdaBoost, on the other hand, trains multiple LR models sequentially, with each model attempting to correct examples misclassified by the previous model. This allows LR, which draws linear decision boundaries alone, to capture more complex data patterns collectively. The SVM model has been not used at this stage because it is not directly compatible with Boosting algorithms. Furthermore, the AdaBoost and GB implementations in the scikit-learn library do not natively SVM. This is primarily because SVM is inherently considered a strong learner and cannot function effectively as a weak and fast learner, as Boosting algorithms require. The AdaBoost algorithm, in its original design, has been optimized only for weak learners and does not perform as well as strong models like SVM. AdaBoost is a separate ensemble learning algorithm that works with weak learners. In this study, AdaBoost has been implemented with a LR model; examples that have been incorrectly classified by the first LR model have been identified, and subsequent LR models have been trained to focus on these challenging examples. As a result of this process, a non-linear and more flexible decision boundary has been obtained by combining different LR models.

Development of the Stacking Based Models

In this study, it has been applied the Stacking ensemble learning approaches to leverage the strengths of different machine learning algorithms. In the Stacking approach, multiple base models (LR, SVM, and LightGBM) are trained independently on the same data, and the predictions of each model on the test set are fed into a second learner, called a meta-model; in this case, LR has been chosen as the meta-model. The meta-model produces the final classification result by optimally combining the predictions of the base models. This brings together the overlapping and complementary aspects of different model types on the data, increasing generalizability and providing a more balanced decision-making mechanism through the weighted contribution of the resulting predictions.

Results and Discussion

The results obtained with the developed models are given in Table 3. The discussion points pertaining to the results obtained from SVM-based models are presented below.

Table 3: The results obtained with the developed models

Method	Model Type	Precision	Recall	Accuracy	F1-Score
SVM	Model 1	0.6333	0.6333	0.6271	0.6333
	Model 2	0.7037	0.6333	0.678	0.6667
	Model 3	0.7241	0.7	0.7118	0.7118
	Model 4	0.7586	0.7333	0.7458	0.7458
	Model 5	0.5946	0.7333	0.6102	0.6567
LightGBM	Model 1	0.6786	0.6333	0.661	0.6552
	Model 2	0.7931	0.7767	0.7797	0.7797
	Model 3	0.7812	0.8333	0.7966	0.8065
LR	Model 1	0.7	0.7	0.6949	0.7
	Model 2	0.7778	0.7	0.7458	0.7368
	Model 3	0.7667	0.7667	0.7627	0.7667
ELM	Model 1	0.7857	0.7333	0.7627	0.7586
Voting	Hard Voting	0.8276	0.8	0.8136	0.8136
	Soft Voting	0.8462	0.7333	0.796	0.7857
Bagging	LightGBM	0.8065	0.8333	0.8136	0.8197
	SVM	0.8	0.8	0.7966	0.8
	LR	0.8214	0.7667	0.7966	0.7931
Boosting	Boosting - LightGBM	0.7143	0.8333	0.7458	0.7692
	AdaBoost - LR	0.75	0.7	0.7288	0.7241
Stacking	LR + SVM + LightGBM	0.75	0.8	0.7627	0.7742

- The incorporation of hyperparameter optimization and embedding features led to a marked improvement in model performance.
- Model 4 demonstrated the highest efficacy by identifying the optimal combination of kernel, cost, and epsilon parameters.
- Conversely, the outcomes associated with Model 5 suggest that the application of GridSearchCV did not yield the anticipated enhancement in performance.
- The discussions regarding the results obtained with LightGBM models are presented below.
- The incorporation of additional features contributed positively to the overall performance.
- The application of GridSearchCV for hyperparameter tuning resulted in the most balanced and superior outcomes across evaluation metrics.
- The most successful performance has been obteined with Model 2.

A general comparison of the developed forecasting models is presented below.

- Additional features and GridSearchCV optimization in LR have provided significant performance improvements.
- ELM demonstrated strong overall performance, yielding comparable results.
- Hard Voting achieved higher performance compared to the Soft Voting in the Voting models.
- The model developed using LightGBM and Bagging approaches provided superior performance.
- It has been observed that the application of LightGBM within the Boosting framework substantially enhanced prediction performance.
- Although the Stacking approach combining LR, SVM, and LightGBM exhibited relatively lower performance compared to Bagging and Boosting methods, it nevertheless demonstrated noteworthy success.

When the results are examined in general, LightGBM demonstrated superior performance over SVM and LR with the incorporation of additional features and hyperparameter optimization. Within ensemble learning approaches, Bagging and Voting methods significantly enhanced predictive performance relative to base models. Although Boosting methods underperformed compared to Bagging, Voting, and Stacking in overall metrics, they exhibited strong results in terms of Recall. The application of GridSearchCV for hyperparameter tuning generally contributed positively to model performance across algorithms. Overall, the most successful predictive performance has been obtained using LightGBM within the Bagging framework.

Conclusion

In the high-volume consulting industry, accurate analysis of customer demands and needs is a critical factor that directly impacts service quality and customer satisfaction. In this context, accurate staff-task matching not only increases operational efficiency and boosts staff motivation, strengthening the organization's brand image. In line with these principles, this study developed staff-task matching models using SVM, LightGBM, LR, ELM, and ensemble learning approaches such as Voting, Boosting, Stacking, and Bagging. The performance of the developed models has been evaluated using Precision, Recall, Accuracy and F1-Score. The most successful model has been obtained with LightGBM+Bagging. Studies in the literature address the task matching problem at a general level and do not offer approaches for matching tasks to specific employees. This study develops unique models focusing on matching personnel and consulting services specifically for the consulting sector.Most existing studies in the literature focus on the suitability of candidate staff for the desired task. This study focuses on matching a task with staff. The problem of matching service staff to tasks in the consulting industry, where service quality directly impacts customer satisfaction and loyalty, is specifically addressed. In addition to the general evaluation of SVM, LightGBM, ELM, and LR algorithms, ensemble learning approaches such as Voting, Bagging, Boosting, and Stacking have been also applied and evaluated.

References

1. Bányai T, Kaczmar I (2024) Staffing and human resource assignment in U-shaped production cells. Advanced Logistic Systems-Theory and Practice, 18: 5-14.

2. Cucus A, Aji LB, Ali AFBM, Aminuddin A, Farida LD (2022). Selection of prospective workers using profile matching algorithm on crowdsourcing platform. In 2022 5th International Conference on Information and Communications Technology. 122-6.

3. Yusoff M, Ikram MNSBM, Janom N (2018) Task assignment optimization for crowdsourcing using genetic algorithm. Advanced Science Letters. 24: 8205-8.

4. Cetin K, Tuzkaya G, Vayvay O (2020) A mathematical model for personnel task assignment problem and an application for banking sector. An International Journal of Optimization and Control: Theories & Applications. 10: 147-58.

5. Jirjees AK, Ahmed AM, Abdulla AA, Lu J, Noori EM, et al. (2025). Machine Learning for Recruitment: Analyzing Job-Matching Algorithms. Machine Learning. 27: 1.

6. Rosenberger J, Wolfrum L, Weinzierl S, Kraus M, Zschech P (2025) CareerBERT: Matching resumes to ESCO jobs in a shared embedding space for generic job recommendations. Expert Systems with Applications. 275: 127043.

7. Eichenseer P, Hans L, Winkler H (2025) A data-driven machine learning model for forecasting delivery positions in logistics for workforce planning. Supply Chain Analytics. 9: 100099.

8. Alonso R, Dessí D, Meloni A, Recupero DR (2025) A novel approach for job matching and skill recommendation using transformers and the o* net database. Big Data Research. 100509.

9. Yue Q, Liu L, Tao Y, Huang H (2025). Person-Position Matching Decision Considering Multi-attribute Preferences and Time Factors in Probabilistic Hesitant Fuzzy Environment. International Journal of Fuzzy Systems. 1-23.

10. Yu C, Gao T (2025) A matching method for elderly care service staff with multiple types of service expectations. PloS one. 20: e0309419.

11. Nguyen TTH, Pham VT, Sangsongfa A (2024) AHP Method Combined with Simulated Annealing Algorithm for Solving Optimal Selection and staff Assignment. In International Conference on Research in Management & Technovation. 183-93.

12. Pias SA, Hossain M, Rahman H, Hossain MM (2024). Enhancing Job Matching Through Natural Language Processing: A BERT-based Approach. In 2024 International Conference on Innovations in Science, Engineering and Technology. 1-6.

13. Sun Z, Liu A, Xiong NN, Zhang S, Wang T (2024) DNF-BLPP: An effective deep neuro-fuzzy based bilateral location privacy-preserving scheme for service in spatiotemporal crowdsourcing.

14. Samanta R, Ghosh SK (2024) Sustainable volunteer engagement: Ensuring potential retention and skill diversity for balanced workforce composition in crowdsourcing paradigm. arXiv preprint arXiv:2408.11498.

15. Tao W, Cui Z, Yue J, Chenhao W (2024) Electromechanical Product Service Optimization in Cloud-based Design: A Two-Stage Method for Task Allocation. In 2024 IEEE 18th International Conference on Control & Automation. 141-7.

16. Zakaria AY, Abdelbadea E, Raslan A, Ali T, Gheith M, et al. (2024) An Improved Enterprise Resource Planning System Using Machine Learning Techniques. Journal of Software Engineering and Applications. 17: 203-13.

17. Zhang H, Dousin O (2024) EfficientNet-Based Prediction Model for Person-Job Matching Values. Informatica. 48: 155-66.

18. Yuan H, Yang G, Li C, Wang Y, Liu J, et al. (2017) Retrieving soybean leaf area index from unmanned aerial vehicle hyperspectral remote sensing: Analysis of RF, ANN, and SVM regression models. Remote Sensing. 9: 309.

19. G Ke, Q Meng, T Finley, T Wang, W Chen, et al. (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree, NIPS.

20. Maharjan R (2021) Employee Churn Prediction using Logistic Regression and Support Vector Machine.

21. Zhang J, Ding W (2017) Prediction of air pollutants concentration based on an extreme learning machine: the case of Hong Kong. International journal of environmental research and public health. 14: 114.

22. Batool A, Byun YC (2024) Toward improving breast cancer classification using an adaptive Voting ensemble learning algorithm. 12: 12869-82.

23. Zhang Y, Liu J, Shen W (2022) A review of ensemble learning algorithms used in remote sensing applications. Applied Sciences. 12: 8654.

24. Mahajan P, Uddin S, Hajati F, Moni, MA (2023). Ensemble learning for disease prediction: A review. In Healthcare. 11: 1808.

25. Soloff JA, Barber RF, Willett R (2024) Bagging provides assumption-free stability. Journal of Machine Learning Research. 25: 1-35.